

Nuance®  
**OmniPage®  
Capture SDK**  
**20.2**

**User's Guide**





## **Nuance Communications, Inc.**

### **SOFTWARE LICENSE AGREEMENT**

**Please read carefully before using this software.**

THIS AGREEMENT STATES THE TERMS AND CONDITIONS UPON WHICH NUANCE COMMUNICATIONS, INC. ("NUANCE") OFFERS TO LICENSE THE SOFTWARE INCLUDED IN THIS DEVELOPER KIT (THE "KIT") TO YOU. BY INSTALLING OR OTHERWISE USING THE SOFTWARE, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PLEASE PROMPTLY RETURN THE KIT TO WHERE YOU OBTAINED IT FOR A REFUND. THIS AGREEMENT SHOULD BE READ IN CONJUNCTION WITH THE SECTION IN THE ACCOMPANYING DOCUMENTATION (THE "DOCUMENTATION") ENTITLED "LICENSING THE PRODUCT".

#### **1. License.**

The Kit includes a set of software components (such as source files, binary libraries and data files) designed for integrating Nuance image processing and recognition technology (such image processing and recognition technology, the "Nuance Technology") into an application that you develop (the "Developed Application").

**This license allows you to:**

- a. use one copy of the Kit and the Documentation internally to integrate one copy of the Nuance Technology into the Developed Application.
- b. use, internally, one copy of the Nuance Technology as and only if integrated in the Developed Application.

In order to (i) sublicense, sell, rent, lease, transfer or distribute the Developed Application containing the Nuance Technology, or (ii) use more than one copy of the Nuance Technology as part of the Developed Application you need to purchase additional licenses from Nuance.

- c. make one copy of the Kit and/or the Nuance Technology in machine-readable form solely for backup purposes, provided that your backup is not installed on any computer.

No patent rights are granted under this Agreement.

#### **2. Restrictions.**

**In addition, you may not:**

- a. SUBLICENSE, MARKET, SELL, RENT, LEASE, TRANSFER OR DISTRIBUTE THE KIT OR THE DOCUMENTATION UNDER ANY CIRCUMSTANCES;
- b. SUBLICENSE, SELL, RENT, LEASE, TRANSFER OR DISTRIBUTE THE NUANCE TECHNOLOGY ON A STAND-ALONE BASIS UNDER ANY CIRCUMSTANCES;
- c. electronically transfer the Kit and/or the Documentation from one computer to another over a network;
- d. decompile, reverse engineer, disassemble or otherwise reveal the code of the Kit and/or the Nuance Technology to a human readable form. Unauthorized copying of the Kit, the Nuance Technology, or the Documentation, including portions thereof, is expressly forbidden. You understand that Nuance may update or revise the Kit, the Nuance Technology or the Documentation and in so doing incurs no obligation to furnish such updates to you. Developed Applications containing the Nuance Technology must prominently display the following Nuance copyright notice sometime during execution ("Nuance™ OCR software © Nuance Communications, Inc. All rights reserved");



e. use the Kit to produce applications that may be shared through the world wide web in an Application Service Provider (ASP) model without the express written authorization of Nuance; or

f. incorporate any Restricted Software in any Developed Application containing the Nuance Technology. As used herein, "Restricted Software" means any software that is licensed under terms that could or might require, by virtue of the incorporation or inclusion of such software in the Developed Application that any portion of the Kit or the Nuance Technology be made available in source code or human readable form.

### **3. Termination.**

This Agreement is effective until terminated. Except for Sections 2, 3, 4, 6, 7, and 8, this Agreement shall terminate automatically upon your breach of your obligations under this Agreement. Upon such termination, you agree to destroy the Kit, including the Nuance Technology and Documentation, and all copies. This remedy shall be in addition to any other remedies available to Nuance.

### **4. Export Control.**

You cannot use the Kit, the Nuance Technology, and/or Documentation in any way or act in any way in connection with the Kit, the Nuance Technology, and/or Documentation that violates any laws or regulations of the United States or foreign governments relating to exports and/or export controls.

### **5. Government End Users.**

This Section applies to all acquisitions of the Kit and/or the Nuance Technology by or for the Federal government or by any prime contractor or subcontractor (at any tier) under any contract, grant, cooperative agreement, or other activity with the Federal government. By accepting delivery of the Kit and/or the Nuance Technology, the government hereby agrees that this software qualifies as "commercial" computer software within the meaning of the acquisition regulation(s) applicable to the procurement. The terms and conditions of this Agreement shall pertain to the government's use and disclosure of the Kit and/or the Nuance Technology. If this Agreement fails to meet the government's needs or is inconsistent in any respect with Federal law, the government agrees to return the Kit and the Nuance Technology, unused, to Nuance. The following additional statement applies only to acquisitions governed by DFARS Subpart 227.4 (October 1988): "Restricted Rights-Use, duplication, and disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (Oct. 1988)." In the event any of the above referenced agency regulations is amended or replaced, the equivalent successor regulation shall apply instead."

### **6. WARRANTY DISCLAIMER.**

THE KIT, THE NUANCE TECHNOLOGY, AND THE DOCUMENTATION ARE PROVIDED "AS IS". NUANCE DOES NOT WARRANT THAT THE OPERATION OF THE KIT OR THE NUANCE TECHNOLOGY MEETS YOUR REQUIREMENTS OR OPERATE FREE FROM ERROR. NUANCE DISCLAIMS ALL OTHER WARRANTIES AND CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY NUANCE, ITS DEALERS, DISTRIBUTORS, AGENTS OR EMPLOYEES SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY, AND YOU MAY NOT RELY ON ANY SUCH INFORMATION OR ADVICE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

### **7. LIMITATION OF LIABILITY.**

NUANCE SHALL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOST BUSINESS PROFITS, LOSS OF INFORMATION, BUSINESS INTERRUPTION, OR THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE KIT OR THE NUANCE TECHNOLOGY EVEN IF NUANCE OR ITS REPRESENTATIVES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION



MAY NOT APPLY TO YOU. NUANCE'S LIABILITY FOR DAMAGES HEREUNDER SHALL IN NO EVENT EXCEED THE LICENSE FEE FOR THE KIT.

#### **8. Copyright/Ownership.**

The Kit, Nuance Technology and Documentation are the proprietary products of Nuance, all of which are protected by copyright and other intellectual property laws. You acquire only the right to use the Kit, Nuance Technology and Documentation and do not acquire any rights, express or implied, in the Kit, Nuance Technology or Documentation other than those specified in this Agreement. Nuance or its licensors shall at all times retain all right, title, interest, including intellectual property rights, in the Kit, Nuance Technology and Documentation.

#### **9. General.**

This Agreement shall be governed and interpreted in accordance with the laws of the Commonwealth of Massachusetts, USA, without giving effect to its conflict of laws principles. This Agreement shall not be governed by the United Nations Convention for the International Sales of Goods, the application of which is expressly excluded. This Agreement contains the entire agreement between the parties with respect to the subject matter hereof, and supersedes all prior agreements or understandings (oral or written) regarding such subject matter unless the parties have entered into and signed an OEM License Agreement covering the same Kit, Nuance Technology and Documentation as is covered by this Agreement, in which event the OEM License Agreement controls if it conflicts with this Agreement.

**Copyright © 2018 Nuance Communications, Inc. All rights reserved.**

**License Agreement Revised October 3, 2016**

#### **Third Party Licenses/Notices**

The following acknowledgments are presented in the alphabetical order of their titles.

##### **Asian language support**

Asian OCR capabilities are jointly developed by the Beijing Wintone Information Technology Corporation Ltd and Nuance Communications, Inc. All rights reserved.

##### **Crypto API**

Partner: Microsoft CAPICOM 2

##### **Dictionaries from CorrectSpell**

Partner: INSO/Vantage Research, Language dictionaries used by the recognition modules MOR, PLUS, DOT and MAT.

Copyright notice: International CorrectSpell™ spelling correction system © 1993 by ScanSoft

##### **Dictionaries from Proximity**

Partner: Proximity, Language dictionaries used by the recognition modules MTX and PLUS.

Copyright notice: The Proximity Dictionaries © 2000, all rights reserved. Proximity Technology, Inc.

##### **eBook conversion**

Partner: Microsoft Reader



#### **Export Options dialog controls**

Partner: Allan Nielsen, Supergrid.

Copyright notice: Supergrid control, copyright © 1999.

#### **File compression library**

Partner: Jean-Loup Gailly and Mark Adler.

Copyright notice: Zlib copyright © 1995-1998 Jean-loup Gailly and Mark Adler.

#### **Form Template Editor UI**

DrawTools, Cyotek ImageBox, DotNet Bar

#### **Handprint recognition module**

Partner: reRecognition, Alphanumerical handprint recognition module (RER).

Copyright notice: Copyright © 2002-2008 re Recognition GmbH. <<http://www.rerecognition.com>>

#### **Image file format**

Partner: Microsoft HD Photo 1.0

#### **Image file reading (Canon MFPs)**

Partner: JBIG-KIT Markus Kuhn

#### **Installer pipeline service**

Partner: InstallShield

#### **JPEG image file library**

Partner: Independent JPEG Groups Thomas G. Lane (freeware).

Copyright notice: The Independent JPEG Group's software, JPEG Software, copyright © 1991-1998, Thomas G. Lane. All rights reserved.

#### **JPEG 2000**

Partner: Kakadu software.

Copyright notice: This product was developed using Kakadu software <<http://www.kakadusoftware.com>>

#### **Licensing and other cryptography**

Partner: OpenSSL Project.

Copyright notice: This product includes software developed by the OpenSSL project <<http://www.openssl.org/>> with software written by Eric Young and Tim Hudson.

#### **Linux XML parsers Xerces and Xqilla**

Partner for Xerxes: The Apache XML Project

Copyright notice: Both items are licensed under the Apache License, version 2. See <<http://www.apache.org/licenses/LICENSE-2.0/>>

#### **Multi-format 1D/2D barcode image processing library:**

Partner: Zebra Crossing

**OCR for CTJK**

Partner: Tsinghua Asian OCR

**Optimized PDF for web-view**

Partner: Linearized PDF creator Sandor Palasik

**PDF digital signatures**

Partner: RSA Data Security.

Copyright notice: Part of this software is derived from the RSA Data Security Inc. MD5 Message-Digest Algorithm.

**PDF encryption**

Partner: Dr Brian Gladman.

Copyright notice: AES encryption/decryption © 2001, Dr Brian Gladman, Worcester, UK.

**PDF encryption**

Partner: SHA256 LibTomCrypt

**PDF file reading**

Partner: The FreeType Project

Copyright notice: Portions of this software are copyright © 2006 The FreeType Project <[www.freetype.org](http://www.freetype.org)>. All rights reserved.

**PDF file reading writing**

Partner: JBIG2 Xerox and Cornell Univ

**PDF rendering**

Partner: Palasik P4

**PNG image file library**

Partner: Colosseum Builders, Inc.

Copyright notice: This software is based, in part, on the work of Colosseum Builders, Inc.

**Read and write PNG files**

Partner: Libpng

**XML parsing**

Partner: MSXML Microsoft

# 1

## 1 - Introduction

Welcome to the Nuance® OmniPage® Capture Software Development Kit, released by Nuance Communications, Inc. This kit provides you with image management and recognition technologies for building 32-bit and 64-bit Microsoft® Windows® applications. We trust you find it a productive tool in your development work. To those of you examining an evaluation copy, we hope you gain positive experience that leads to a purchase.

### 1.1- Abbreviations Used in the Document

Some specific abbreviations occur in this guide, and are used instead of their full definitions except for their first mentioning, as follows:

- NCLT - Nuance Central Licensing Tool
- NCLS - Nuance Central Licensing Service (formerly also known as LMS - License Manager Service)
- NLM - Nuance License Management solution
- HWFP - Hardware Fingerprint (a unique hardware specific ID)
- LFR - Logical Form Recognition™
- IVC - Image Viewer Control (a visual control)
- TEC - Text Editor Control (a visual control)

### 1.2- Kit Components and Overview

For Windows, the following variants of the Capture SDK are available:

#### 1.2.1 - SDK Configuration

- Western OCR Kit, v20.1

#### 1.2.2 - Optional Add-Ons

- Asian Plus Add-on, v20.1
- Recognition Add-on, v20.1

#### 1.2.3 - Runtime License Packs (Desktop Runtime)

- Western OCR, Desktop, Fixed Volume, v20.1
- Asian Plus Add-on, Desktop, Fixed Volume, v20.1
- PDF & XPS Add-on, Desktop, Fixed Volume, v20.1
- Barcode 1D & 2D Add-on, Desktop, Fixed Volume, v20.1
- ICR Add-on, Desktop, Fixed Volume, v20.1

# 2

- OMR Add-on, Desktop, Fixed Volume, v20.1
- Formatted Output Add-on, Desktop, Fixed Volume, v20.1
- Document Classifier Add-on, Desktop, Fixed Volume, v20.1
- Visual Toolbox Add-on, Desktop, Fixed Volume, v20.1
- FTE and DCTT Desktop, Fixed Volume, v20.1

## 1.2.4 - All Server Licenses

- Page Per Month 50K 1-Year
- Page Per Month 250K 1-Year
- Page Per Month 1M 1-Year
- Page Per Month 5M 1-Year
- Page Per Month 50K 2-Year
- Page Per Month 250K 2-Year
- Page Per Month 1M 2-Year
- Page Per Month 5M 2-Year
- Page Per Month 50K 3-Year
- Page Per Month 250K 3-Year
- Page Per Month 1M 3-Year
- Page Per Month 5M 3-Year
- Page Pack 50K 1-Year
- Page Pack 250K 1-Year
- Page Pack 1M 1-Year
- Page Pack 5M 1-Year

## 1.2.5 - Desktop OEM Licenses

- Western OCR, Desktop, OEM, v20.1
- Asian Plus Add-on, Desktop, OEM, v20.1
- PDF & XPS Add-on, Desktop, OEM, v20.1
- Barcode 1D & 2D Add-on, Desktop, OEM, v20.1
- ICR Add-on, Desktop, OEM, v20.1
- OMR Add-on, Desktop, OEM, v20.1
- Formatted Output Add-on, Desktop, OEM, v20.1

# 3

- Document Classifier Add-on, Desktop, OEM, v20.1
- Visual Toolbox Add-on, Desktop, OEM, v20.1

## 1.2.6 - Server OEM Licenses

- Western OCR, Server, OEM, v20.1
- Asian Plus Add-on, Server, OEM, v20.1
- PDF & XPS Add-on, Server, OEM, v20.1
- Barcode 1D & 2D Add-on, Server, OEM, v20.1
- ICR Add-on, Server, OEM, v20.1
- OMR Add-on, Server, OEM, v20.1
- Formatted Output Add-on, Server, OEM, v20.1
- Document Classifier Add-on, Server, OEM, v20.1
- Visual Toolbox Add-on, Server, OEM, v20.1

## 1.2.7 - Evaluation Licenses

- Evaluation 45-days, Fixed Volume, v20.1
- Evaluation 45-days, OEM, v20.1
- Evaluation Kit valid for 45 days from issuance, v20.1 (including eval, desktop, runtime)
- Evaluation Page Pack 5K valid for 45 days from issuance, Server, v20.1

The availability of add-ons depends on the type of the licenses you have.

Evaluation versions of this Kit are obtained by download which includes click-through acceptance of an evaluation agreement with Nuance. Typically, evaluation versions provide access to all of the above technologies.

This User's Guide gives you an overview of the program, its features and the main Toolkit components. It is available in PDF format. The help systems for RecAPI, IPRO and the MediumWeight Visuals present comprehensive, detailed information on areas of interest. For generic guidance, consult the General Information Help. The License Manager used up to version 19 is replaced by a new Nuance Central Licensing Tool, so read the following sections carefully.

Evaluation Kits are supplied with a time-limited development license. Test distribution licenses are also time-limited.

There are three distribution licensing schemes available:

- **Fixed Volume Licenses** (require activation)
- **OEM licenses** (variable volume licenses supplied under a contract with Nuance)
- **Central licenses** (Page Pack and Page Per Month licenses)

# 4

The distribution licensing method to be used does not need to be decided until the application is ready for release. For details, see the **OmniPage Capture SDK 20.1 - General Information Help**.

All licensing for Linux and Macintosh kits (development and distribution) is handled exclusively through OEM licenses.

System requirements for the Toolkit can be found in the General Information online help system, or on the [Nuance website](#) under **OmniPage Capture SDK**.

## 1.3- Overview of Building and Distributing Applications

To guide you through the process from purchasing the Nuance OmniPage Capture SDK to developing your own application, see the following topics:

- (Pre-)Installation ("[Before Installing](#)" below, "[Installation Steps](#)" on the next page)
- Getting started with the Nuance OmniPage Capture SDK ("[Deciding on the Interface](#)" on page 7, "[Running the Sample Viewer Application](#)" on page 7)
- Distribution ("[Preparing a Distribution File Set](#)" on page 8, "[Testing Applications Using Temporary Test Licenses](#)" on page 9)
- Assistance ("[Getting Help](#)" on the next page, "[Introduction](#)" on page 1)

This guide gives you an overview of the areas listed. For more in-depth information, consult the General Information Help.

## 1.4- Quick Start with the SDK

To get started without delay and get to the point where you can start building your application, refer to the following chapters in this Guide:

- Chapter 1.5: "[Before Installing](#)" below
- Chapter 1.6: "[Installation Steps](#)" on the next page
- Chapter 1.9: "[Basic Test to Check Your Installation](#)" on page 6

## 1.5- Before Installing

1. You do not need to remove previous major versions of the Nuance or ScanSoft OmniPage Capture SDK products. Should you wish to, do this through the Windows Control Panel's **Programs and Features** (Windows 7) item. For earlier (point) releases of the current version, check the Release Notes in the Readme file ([ReadMeEng.htm](#)).
2. Ensure that the installation package is available: Download the appropriate 32-bit and/or 64-bit package from Nuance Network (<https://network.nuance.com>) according to your need – whether you want to create a 32-bit or a native 64-bit application.

# 5

3. Browse the downloaded and unzipped content for the latest information: You can find a Release Notes (`ReadMeEng.htm` file) and a User's Guide document (`CSDK_UsersGuide.pdf`) in the **Documentation** subfolder of the unzipped content quoting a section on the new features and changes that summarizes the additional and improved features compared to earlier versions. You can also find these and other documents in the Documentation sub-folder after installation, under the destination folder of installation.
4. Check that your system meets the minimum requirements for memory and disk space and be ready to select install location.
5. Ensure that internet access is available for product license activation which is required to use the SDK: Internet access can either be available on your developer machine (the computer where you are about to install the SDK) or on another one that you use for reaching the activation webpage. In case your developer machine has internet access, it is recommended that you activate your license directly online. If you choose to activate using a second machine, make sure you can transport files between this computer and your developer workstation, and use Manual activation. For details see the section on activation.

## 1.6- Installation Steps

Make sure that all other Windows programs are closed; we recommend that the anti-virus software be disabled.

1. Ensure you have administrator privileges on the installation computer.
2. Browse in the downloaded and unzipped content to the file `setup.exe` and run it.
3. Accept the license agreement when prompted, and press **Next**.
4. Provide your User Name and that of your Organization.
5. Accept or define the destination path and a program folder for the program binaries and icons.
6. When installation is complete, click **Finish**.
  - The Nuance OmniPage Capture SDK setup program installs all files to the developer system; components (Form Template Editor, Document Classifier Assistant) can be selected during the setup procedure.
  - If you do not have the Microsoft .NET 4.0 Framework on your computer, it is installed silently during OmniPage Capture SDK installation, because it is a prerequisite for several SDK components.
  - You do not have to provide your license key during installation.
  - In some cases, the installer recommends a reboot. This should be accepted.

**Hint:** Ensure that the licensing service (Nuance Central Licensing Service) is added to the list of firewall exceptions on the machine you plan to use for activation.

## 1.7- Getting Help

This User's Guide is available in PDF format in a Nuance OmniPage Capture SDK sub-folder called **Documentation**, on the Nuance OmniPage Capture SDK Developer Community web-site. It presents the

# 6

following seven chapters:

- **Introduction:** ([page 1](#)) Installation, Distribution, etc.
- **System Requirements:** ([page 12](#)) Windows only
- **New Features and Changes:** ([page 14](#)) Overview of innovations and improvements
- **RecAPI:** ([page 30](#)) Overview of the native C++ interface with over 400 entry points
- **IPro:** ([page 43](#)) Overview of ActiveX controls and the object model, and the use of workflows
- **Visual Components:** ([page 66](#)) Overview of ready-made and customizable UI elements
- **Appendix:** ([page 80](#)) Sample barcode images

The Toolkit is shipped with the following HTML help files:

- **General Information:** System requirements, Licensing, Testing and Distributing Applications, etc.
- **RecAPI:** Full reference of all functions, enumerations and parameters for C++ implementation; it also contains the full RecPDF API reference. This system contains most of the overview and reference topics not repeated for IPro or Visuals.
- **IPro:** Full reference of the object model, events, methods and properties.
- **Visuals:** Full description of all visual components. It contains MediumWeight Visuals Namespace and VisualsLib Namespace along with MediumWeight Visuals Controls and Visuals Controls.

The help includes Tutorials, Sample Codes and Reference topics. Access this through the Windows Start menu or Help buttons in the programs. Readme files present latest information and solutions to recently reported problems.

A Porting Guide (available on demand) documents the architectural changes that must be handled when upgrading from Nuance OmniPage Capture SDK 12 to Nuance OmniPage Capture SDK 15.

All these items are stored in electronic form in the sub-folder Documentation. The full collection is accessible to those evaluating the Kit, and also to purchasers, regardless of which Kit or Add-ons are purchased.

## 1.8- Basic Test to Check Your Installation

To check whether your Nuance OmniPage Capture SDK installation has been successful, you can perform some basic tests by means of the RecAPI Sample Viewer Application. (On launching this application see the section "[Running the Sample Viewer Application](#)" on the next page further below in this document).

Launch the Sample Viewer and then run Sample 1. The Output Window displays the results of initialization and Module Information. The initialization test confirms that installation and license activation were successful and also that the recognition engine is ready for use on your machine. The module information test lets you check that all licensed modules are really available.

If initialization fails, required Activation steps (in particular, running the NCLT) may not have been performed properly.

This sample has additional information for you to check, for instance:

# 7

- the version number of the Nuance OmniPage Capture SDK you have just installed

You can safely ignore error messages displayed under Module Information in this window. The program only generates these messages for demonstration purposes.

## 1.9- Deciding on the Interface

For evaluation purposes we recommend working with the object oriented interface for the RecAPI and possibly the supplied test application ITest. Use the IPRO interface only if you need to access the functionality of the MediumWeight Visuals. For your development work both interfaces yield the same recognition accuracy. RecAPI is divided into two parts, so your evaluation and development choices are as follows:

- C#.NET and other .NET RecAPI interface: object oriented interface for the RecAPI can be used from these programming languages, this is the most widely used interface nowadays.
- The RecAPI C/C++ interface (Kernel API) is best suited to high throughput single page processing tasks with plain text, XML, or IOT PDF output, and for applications where processing time needs to be minimized. In order to minimize processing time, you can also use multithreading provided via `kRecProcessPagesEx`.
- The RecAPIPlus interface supports complex outputs, and it can also be used from the Object Oriented .NET class interface; it additionally provides support for document-level and high-volume (possibly unattended) processing. It supports fully formatted output, one-step workflow-style processing and other special features such as redaction and form data extraction. It leverages multithreading to deliver performance benefits on multi-core and hyper-threaded computers.
- The IPRO interface can also be used from .NET (Samples/ISApp/NewCSHARP) and is necessary for developers who wish to take advantage of the MediumWeight Visuals that helps create and use OmniPage-compatible workflows. IPRO supports document-level processing, along with the ability to easily access all objects and properties involved. ActiveX (another layer on IPRO) controls are better avoided for high-volume processing; see the RecAPI Sample on 24/7 running.

## 1.10- Running the Sample Viewer Application

A sample viewer application is provided for each interface. These are designed to demonstrate the functionalities of the toolkit through over 60 working samples per platform. We recommend you view the sample indexes to determine which are closest to your configuration and development plans, and run those. For a general overview, we suggest you run samples 1-14, to test basic functionality.

All samples are available and can be viewed before any license activation, but only those samples will run, for which the used technology is licensed on the actual machine.

The Sample Viewers can be launched from the Windows Start Menu, under **Nuance > OmniPage Capture SDK 20.1 > Samples**. Choose the viewer appropriate for the interface you plan to use for development.

- CSharp Class Samples (C# 10.0)
- IPRO Sample Viewer Application (C# 10.0)
- IPRO Sample Viewer Application (C++ 10.0)

# 8

- IPRO Sample Viewer Application (VB.NET 10.0)
- New IPRO Sample Viewer Application (C# 10.0)
- New Visual Controls Sample Application (C# 10.0)
- RECap Sample Viewer Application (C)
- Visual Controls Sample Application (C# 10.0)
- Visual Controls Sample Application (C++ 10.0)
- Visual Controls Sample Application (VB.NET 10.0)

The Sample Viewer window displays the following areas:

- **Description window:** summarizes the purpose of the current sample.
- **Console window:** displays status messages as the samples run to denote progress and to report any problems encountered.
- **Function window:** lists all functions/objects involved in running the sample. Select a function/object and then press **Help** for detailed information on that item.

---

**Note:**

To use a scanning device with the Capture SDK, it must be already installed using its own installation software and functioning correctly. Consult the scanning topics in the documentation.

---

## 1.11- Distributing Applications

### 1.11.1 - Preparing a Distribution File Set

This section describes how to distribute the necessary Engine files with your application developed using this Capture SDK.

The Distribution Wizard is a separate executable you can use to collect distribution file sets. Since it collects the files from the Nuance OmniPage Capture SDK installation folder, it always gets the latest installed content. In case you have any patch or point release installed, the file set collection reflects this update.

The Nuance OmniPage Capture SDK setup program installs all files to the developer system; components (Form Template Editor, Document Classifier Assistant) can be selected during the setup procedure on the **Customize installation** screen.

To distribute an application, follow the steps below:

1. Prepare an Engine configuration for distribution (create a distribution file set).
2. Include the file set in a setup program or installer you have written.
3. Copy these Engine files (together with your integrating application files) to your customer machine.

# 9

4. Read the hints in the Distribution Wizard log, and make sure the necessary system files are installed on the target computer.
5. Setup the user's system for scanning, if necessary.
6. Ensure that recognition modules (and the PDF output converter and other Add-ons) used by your integrating application get licensed.

These steps are described in detail in the General Information help.

## 1.11.2 - Testing Applications Using Temporary Test Licenses

After you have finished compiling the distribution file set, the next step is testing your application. This guide presents the basic outline: for details see the corresponding topic in the General Information Help.

The test computer can be any computer where the Capture SDK has not been installed, and no developer license has been activated.

1. Be aware which recognition modules are used by your application.
2. Check whether your distribution licensing supports all the RECOGNITIONMODULEs required by your application.
3. Ensure that your application checks the availability of required modules.
4. Select the distribution file set of the Engine.
5. Activate one or more temporary Fixed Volume Licenses on the test computer.
6. Check which Engine modules are licensed on the test computer.
7. Check which Engine modules are licensed, available and initialize correctly.
8. Fully test the application and then remove the log file (`DistributionWizard.log`) and the `DISTR_TST.exe` executable from the test-folder.

All aspects of the application should be tested and passed through a QA procedure, to ensure correct interaction with the Nuance OmniPage Capture SDK in the test environment.

This section describes how Nuance Communications, Inc. licenses the products included in the Capture SDK to you. This section IS INTENDED AS A GUIDE ONLY. Your license agreement (the Software License Agreement) with Nuance governs the terms and conditions of your use of Nuance's products. In the event of any inconsistency between the terms of this section and the Software License Agreement, the terms of the License Agreement shall prevail.

The Capture SDK provides a programming interface to a variety of image processing and recognition technologies. The product includes scanning, image pre-processing, recognition and output formatting software modules. The Capture SDK is designed to help create software applications that contain Nuance recognition software that can be resold to third parties. Before reselling the software application, however, distribution licenses for the Capture SDK modules being used and distributed with the application must be obtained from Nuance. The correct licensing of the required modules is vital; otherwise either the Engine initialization fails or one or more modules become unavailable.

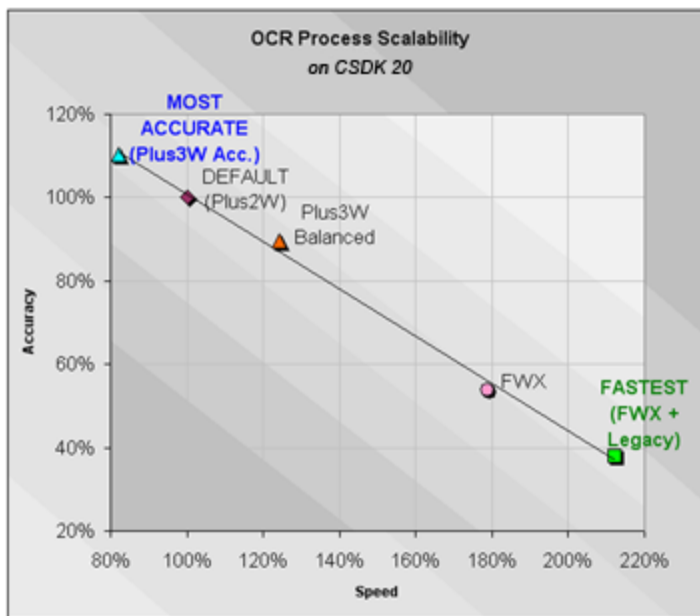
# 10

For more details on licensing, see the General Information Help.

## 1.12- Performance Comparison of Engine Combinations

This chart shows the speed/accuracy trade-off using selected Engine combinations and trade-off settings. Data was measured on a 3 GHz Core Duo machine with 2 GB RAM, with the following test set:

- about 5000 pages in single-page TIFF files, with 1.13 million words.
- mixed resolutions (200-300 dpi), both black-and-white and color
- mixed languages (English, French, German, Italian, Hungarian, Polish, Portuguese)
- mixed layout (magazines, business letters, faxes, business cards...)
- mixed scanners



For both speed and accuracy the 100% value is for the default configuration (Plus2W). This means the fastest configuration is slightly over twice as fast as the default. The percentage differences translate into word accuracy values as follows:

- Most accurate (Plus 3W): 97.2%
- Default (Plus 2W): 96.9%
- Fastest (FRX + Legacy): 92.5%

# 11

## 1.13- Image Size Limits

There are no changes in size limitations compared to earlier versions of Capture SDK.

## 1.14- Uninstalling the Product

Before uninstalling the Nuance OmniPage Capture SDK, it is recommended that you remove your fixed-volume development license.

Go to the Windows Control Panel and select **Programs and Features** (Windows 7) and then choose **Nuance OmniPage Capture SDK 20.1**. This removes the Kit. Be sure to keep your license key and license codes in a safe place; you may need them to re-install the program or to install an upgrade. Some files may not be removed during un-installation, for example, result files generated by running the sample viewers.

# 12

## 2 - System Requirements

System requirements for a computer used to **develop** a Capture SDK-enabled application

- Operating Systems:
  - Microsoft Windows Server 2008\* R2
  - Microsoft Windows Server\* 2012/R2
  - Microsoft Windows Server\* 2016
  - Microsoft Windows 7 x86/x64
  - Microsoft Windows 8/8.1 x86/x64 (Windows RT not supported in this release)
  - Microsoft Windows 10 x86/x64
- Microsoft .NET Framework 4
- Pentium4 1.6 GHz or higher processor (Intel Core or higher CPU is recommended)
- 4 GB minimum RAM (6 GB recommended, more for working with grayscale or color images and more for multithreaded applications)
- 4 GB free hard disk space

\* Nuance OmniPage Capture SDK runs on Server type operating systems when Server license feature is present in the Nuance OmniPage Capture SDK license.

System requirements for a computer used to **run** a Capture SDK-enabled application

- Operating Systems:
  - Microsoft Windows Server 2008\* R2
  - Microsoft Windows Server\* 2012/R2
  - Microsoft Windows Server\* 2016
  - Windows Vista SP1 32/64,
  - Microsoft Windows 7 x86/x64
  - Microsoft Windows 8/8.1 x86/x64 (Windows RT not supported in this release)
  - Microsoft Windows 10 x86/x64
- Microsoft .NET Framework 4
- Pentium4 1.6 GHz or higher processor (Intel Core or higher CPU is recommended)
- 2 GB minimum RAM (4 GB recommended, more for working with grayscale or color images, and more for multithreaded applications)
- 300 MB free hard disk space (less if not all recognition modules are distributed)

# 13

\* Nuance OmniPage Capture SDK runs on Server type operating systems when Server license feature is present in the Nuance OmniPage Capture SDK license.

## 2.1- User Development Tools

The following development tools are considered fully supported by the Capture SDK.

- Microsoft Visual Studio 2008-SP1
- Microsoft Visual Studio 2010
- Microsoft Visual Studio 2012
- Microsoft Visual Studio 2013
- Microsoft Visual Studio 2015 Update 2 (used to produce the Capture SDK)

Other (not tested) development environments with COM and/or .NET Client support should also work with Capture SDK.

# 14

## 3 - New Features and Changes

This document presents an overview of the innovations, improvements and changes in version 20.1 of the OmniPage Capture SDK.

The texts include brief references to the most relevant RecAPI functions or IPRO objects or MediumWeight Visuals controls. The 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.

### 3.1- Version 20.1 Changes

- **Hebrew OCR Language added to this Release**  
Both DTEXT and Formatted outputs are supported.
- **Arabic Formatted Output support added to this Release**  
From this release Thai and Arabic formatted outputs is supported.
- **New DCTool to support automated Document Classification Training**  
**dctool.exe** is a command line tool (in bin\bin64 folder) that allows CSDK application developers to create better document classification solutions with their own training data. It automates DCAssistant capabilities using command line options. For details, see the description displayed when you type `dctool -help` parameter in command line and the Document Classifier Assistant help. It can be redistributed in the solution application.
- **User written custom workflow support in Workflow Runner**  
This release introduces the custom workflows. They can be written by the CSDK users in .NET or C++ using full IPro or even RecAPI to implement special programs or business logic, for instance Form Template Processing or Document Classification. CSDK has a sample custom workflow under the **C:\Program Files (x86) \Nuance\OPCaptureSDK20\Samples\WorkflowRunner\CustomJobItem** path. Another complete sample solution with a sample Job Type and Application is located in the **C:\Program Files (x86) \Nuance\OPCaptureSDK20\Samples\WorkflowRunner\Zoning.zip** file. Unzip it and follow the description in the **ReadMe.txt** file. Study those samples and the Tech Note part of the documentation for the details.
- **New Sample Projects for Intelligent Workflow Designer and the new inline descriptions in the produced WorkflowXMLs**  
Find the **.ocrjob** files in the **bin/bin64** folder. Load them into the Workflow Designer. Run inside the Designer under the **Job->Run** menu. Open the output WorkflowXML files in Notepad and study the xml comments. There are descriptive comments for better understanding the WorkflowXML.
- **ICR2 Recognition Engine (handwriting, Thai) updated. It does not require separate VS2013 CRT to be installed**  
It requires VS2015 CRT same as CSDK.

# 15

- **Control setting in handling JPG images without DPI data**

`Kernel.Imf.JPG.Resolution` (int, default is `USE_DEFAULTDPI`) `USE_DEFAULTDPI` if no DPI data available in the JPG file, CSDK uses `Kernel.Imf.DefaultDPI` for the actual resolution. Default `USE_EXIF` if exif data available in the JPG file, use its resolution number. If no such data exists, CSDK takes `Kernel.Imf.DefaultDPI` setting

- **kRecLoadImgM has new a setting to the number of colors in the image**

`Kernel.Imf.MinimizeBitsPerPixel.FromMemory` (boolean, default is false for compatibility reason) `kRecLoadImgMC` investigates the number of colors in the image (depending on a setting) and adjusts `BitsPerPixel` to a minimum (similar to `kRecLoadImg`)

- **New UI tool for handling Fixed Volume Licenses in the redistribution**

This program is a MFC application (`OPLicMgrUI.exe`) that makes the end user's life easier. A simple UI drives users through the license activation and the process of moving Fixed Volume Licenses from one computer to another.

- **Improvements in Automatic Language Detection, 3D deskew**

- **Asian language support added to Form Template Processing and FTE**

- **General bug fixing and stability improvements**

- **New HTML5 output converter**

## 3.2- Version 20 Changes

### 3.2.1 - Document Classifier Assistant

New set of functions and structures on RecAPI to support Document Classification functionality. Document Classifier Assistant allows the user of an application to create Classification Knowledge Data. DCA can also be used for test classifications (matching) to verify the correctness of the training data. The new API can be used to load a Classification Knowledge Data and to do actual classifications based on it; it also allows post-training in the field. Consult the documentation for the details.

## 3.3- Intelligent Workflow Runner

### 3.3.1 - Introducing OCRService/CSDKWorker component of the Nuance OmniPage Capture SDK V20

OCRService is a high availability API for long running Nuance OmniPage Capture SDK applications with scale-up capability and declarative programming method.

OCRService 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. OCRService has a very simple API. The actual Nuance OmniPage Capture SDK workflow is given in an XML string in a declarative way. OCRService has many built-in commonly used Nuance OmniPage Capture SDK workflows, like converting images to Docx or searchable PDF. These are basically IProPlus workflows. An OCRService string describes workflows and their parameters. It also names the input-output files and sets Nuance OmniPage

# 16

Capture SDK settings. After a OCRService has been processed, a response XML string returns to the application with statistics and errors, if any.

In this Nuance OmniPage Capture SDK release the OCRService is not fully documented yet. The `AssistantApp.exe` should be used to create classic workflows (binary `xwf` file) and a tool `XWFToXML.exe` to convert `.xwf` files to OCRServices. The content of an OCRService file can be given a first parameter of `Run()` function.

The OCRService process maintains multiple Nuance OmniPage Capture SDK worker processes. It stops and restarts workers on a regular basis (default: every 10 OCRService execution) and at every occurrence of a Nuance OmniPage Capture SDK error, crash, hang or infinite loop. Before a restart, OCRService cleans 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 OCRService `Run()` function execution is an asynchronous process. There is a done event to notify the application about the end of an OCRService. OCRService maintains a small queue of OCRServices feeding the Nuance OmniPage Capture SDK worker processes. The number of workers can be parameterized. Each Nuance OmniPage Capture 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 OCRService, the application can reduce (throttle) the CPU usage of a single worker if needed.

## 3.3.2 - OCRService usage

- The two out-proc COM processes must be registered (at admin prompt or in the application installer):  
`OCRService.exe / Regserver`  
`OCRServer.exe / Regserver`  
 (Note: by default the Nuance OmniPage Capture SDK Installer does not register these)
- In Visual Studio, add Nuance OmniPage Nuance OmniPage Capture SDK OCRService 20.1 Type Library to references of your application. (In COM tab of the **Reference Manager** dialog)  
 From a C++ program, use `#import "OCRService.exe" named_guids` to access OCRService COM.
- Set `OCRServiceLib Embedded Interop Types` property to `False`.
- Implement the application. Use provided sample program as a guideline.

## 3.3.3 - Advanced usage of OCRService

OCRService can be hosted in a Windows NT Service (maintaining OCRService process life cycle). Get OCRServices from durable data store components like database queues, ServiceBus, Azure ServiceBus. In such a system there can be multiple nodes (scale-out) each running with an OCRService instance with multiple workers.

## 3.3.4 - OCRService API

```
void SetLicense(string LicenseFile, string Key)
void Run(string XML, string ID)
```

# 17

```

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, Sys-
tem.Array ResponseXMLFiles)
.

int TimeOut { get; set; } // maximum OCRService 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; } // OCRServices between two restarts
bool Log { get; set; } // turn the log off
.

```

## 3.4- Other major changes

- This release the Nuance OmniPage Capture 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 Tool.  
In addition from v20 Nuance OmniPage Capture SDK starts to count processed pages. Nuance OmniPage Capture 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 Nuance OmniPage Capture SDK processes running on other computers in the LAN. NCLT creates monthly usage reports to help Nuance OmniPage Capture 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 to install NLS or Side by Side mode.
- Enhanced performance: this release includes general stability and functionality bug fixes
- Asian zoning improvements
- Jpeg 2000 Engine updated
- Kakadu version updated to 7.8
- Nuance Vocalizer 2.0/2.1 (RealSpeak Solo removed) 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 Nuance OmniPage Capture SDK release.
- Logical Form Recognition Engine can now recognize rounded check boxes
- Docx, Xlsx, Pptx Outputs do not require .NET framework Installed
- RER Handwritten OCR engine (ICR2) updated and has been built using VS2013. If your distribution uses the RER engine, your installer has to install the vc\_redist of VS2013.

# 18

- 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 Nuance OmniPage Capture SDK Installer installs vc\_redist for VS2010.
- RecIProPlus API retired: the RecIProPlus component has been removed from this release. If you still intend to use it, the current Nuance OmniPage Capture SDK release contains the obsolete component source files. Locate these (`RecIproPlus.cpp` and `RecIproPlus.h`) at the following path: **<Nuance OmniPage Capture SDK 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.

## 3.5- Version 19.20 Changes

### 3.5.1 - Service Release

Contains all the fixes applied in v19.11.

### 3.5.2 - New API for .NET Applications

It is the replacement of the old P/Invoke interface of the RecAPI. This new API is object oriented, like other .NET style APIs or the .NET Framework itself.

In addition, the RecAPI P/Invoke has been reorganized to support this new feature. The new P/Invoke is implemented in two new assemblies: `Nuance.OmniPage.Nuance OmniPage Capture SDK.CAPI.dll` and `Nuance.OmniPage.Nuance OmniPage Capture SDK.ArgTypes.dll`. In a new P/Invoke style project, you can use these two assemblies, but we encourage you to use the new .NET API instead.

Old P/Invoke (`CAPI_PInvoke.dll`) is supported, but it might be removed from the next major Nuance OmniPage Capture SDK Release.

If you use `RecPDF P/Invoke` and `PDF.RecAPIError` exception, you have to add `Nuance.OmniPage.Nuance OmniPage Capture SDK.ArgTypes.dll` to the References:

### 3.5.3 - New Arabian Language Support

New tag in LANGUAGE enum: `LANG_ARA`.

### 3.5.4 - New Asian Recognition Engine with trade-off setting

Trade-off setting works with the Asian engine.

### 3.5.5 - Unified Compression Settings

See in `KernelAPI.h`. These are not available on `IproPlus` in this release. They have effect on DTXT and Imf outputs. Sample68

```
RECERR RECAPIKRN kRecSetCompressionLevel(int sid, int CompressionLevel);
```

# 19

```

RECERR RECAPIKRN kRecGetCompressionLevel(int sid, int *pCompressionlevel);

typedef enum
{
    COMPRESSION_FAST,
    COMPRESSION_ADVANCED
} COMPRESSION_TRADEOFF;

RECERR RECAPIKRN kRecSetCompressionTradeoff(int sid, COMPRESSION_TRADEOFF
CompressionTradeoff);

RECERR RECAPIKRN kRecGetCompressionTradeoff(int sid, COMPRESSION_TRADEOFF*
pCompressionTradeoff);

```

## 3.5.6 - New API for loading image file page without decompressing the image inside

See in KernelAPI.h. It is not available on lproPlus in this release. Sample67

```

RECERR RECAPIKRN kRecLoadImgDataStream(int sid, HIMGFILE hIFile, HPAGE
*phPage, int iPage);

RECERR RECAPIKRN kRecLoadImgDataStreamF(int sid, LPCTSTR pFilename, HPAGE
*phPage, int nPage);

RECERR RECAPIKRN kRecDecompressImgDataStream(int sid, HPAGE hPage);

RECERR RECAPIKRN kRecFreeImgDataStream(int sid, HPAGE hPage);

RECERR RECAPIKRN kRecLoadImgDataStreamMC(int sid, BYTE *lpBuf, size_t bufLen,
LPCOMPRESSED_IMG_INFO lpCImg, HPAGE *phPage);

RECERR RECAPIKRN kRecLoadImgMC(int sid, BYTE *lpBuf, size_t bufLen,
LPCOMPRESSED_IMG_INFO lpCImg, HPAGE *phPage);

```

Supported compression: IMG\_COMPRESSION enum.

## 3.5.7 - New API for creating PDF searchable output

See in KernelAPI.h. It is not available on lproPlus in this release. Sample66

```

RECERR RECAPIKRN kRecMakePagesSearchable(int sid, LPCSTR pFilename, int
fromPage, const HPAGE* ahPage, int nPage, IMAGEINDEX iiImg);

RECERR RECAPIKRN kRecConvert2DXTXEx(int sid, const HPAGE *ahPage, int nPage,
IMAGEINDEX iiImg, LPCTSTR pFilename);

```

## 3.5.8 - New functions

See in KernelAPI.h. It is not available on lproPlus in this release.

# 20

```
#define IMF_PDF_AS_IMAGE 1ss="style2"> #define IMF_PDF_AS_IMAGE 1
#define IMF_PDF_NO_TAGS 2
#define IMF_PDF_ANN_FORM (4 << 0)
#define IMF_PDF_ANN_TEXT (4 << 1)
#define IMF_PDF_ANN_HIGHLIGHT (4 << 2)
#define IMF_PDF_ANN_UNDERLINE (4 << 3)
#define IMF_PDF_ANN_CROSSOUT (4 << 4)
#define IMF_PDF_ANN_STAMP (4 << 5)
#define IMF_PDF_ANN_LINE (4 << 6)
#define IMF_PDF_ANN_SQUARE (4 << 7)
#define IMF_PDF_ANN_CIRCLE (4 << 8)
#define IMF_PDF_ANN_POLYGON (4 << 9)
#define IMF_PDF_ANN_POLYLINE (4 << 10)
#define IMF_PDF_ANN_PENCIL (4 << 11)
//! Setting & getting load flags for pdf file loading
/*! The kRecSetImfLoadFlags function set an IMF Load Flags combination
\param[in] sid
Settings Collection ID.
\param[in] fFlag
The IMF Load Flag combination
\retval RECERR
*/
RECERR RECAPIKRN kRecSetImfLoadFlags(int sid, DWORD fFlag);
RECERR RECAPIKRN kRecGetImfLoadFlags(int sid, DWORD *pfFlag);
```

## 3.5.9 - ITest is able to run with any Nuance OmniPage Capture SDK Kit

### 3.5.10 - New OEM licensing mode – 'Internal mode'

- Does not require .NET Framework
- Does not require the NuanceLS.exe running in any way (in service or in side-by-side process)
- Does not use TCP/IP
- Does not require Internet connection in runtime/deployment at all

# 21

- Standard OEM license keys issued by Manufacturing from the beginning of 2014 can be used
- Does not require new API – simply a slight modification in application source code

## 3.5.11 - General Improvements in PDF MRC Output

## 3.6- Technical Notes

The following issues are added to the documentation.

### 3.6.1 - Changes in P/Invoke style newly developed applications

Enums and Structures moved to a separate assembly `Nuance.OmniPage.Nuance OmniPage Capture SDK.ArgTypes.dll` and P/Invoke itself lives in `Nuance.OmniPage.Nuance OmniPage Capture SDK.CAPI.dll`. Newly developed P/Invoke applications need to add these assemblies to the References. Old `CAPI_PInvoke.dll` is deprecated, it will not be supported from the next major Nuance OmniPage Capture SDK Release.

Please study the new API for .NET Application.

## 3.7- Version 19.11 Changes

### 3.7.1 - Service Release

Contains all the fixes applied in v19.00

### 3.7.2 - Uninstall v19.00 (32-bit and 64-bit) before installing v19.11

Form Template Editor v1.10 requires Nuance OmniPage Capture SDK v19.11. Nuance OmniPage Capture SDK 19.00 and 19.11 side by side installation is not supported in this release. In redistribution, any Nuance OmniPage Capture SDK versions can coexist on the same computer (in different installation folders).

### 3.7.3 - Fine-tuned Form Template Editor and Form Template Processing

- No error returned if the template contains reference anchors but does not contain positional anchors (bugfix).
- More accurate determination of zone positions when the template contains only two positional anchors.
- Processing camera images: more accurate determination of zone positions when the template does not contain positional anchors (only reference anchors).
- Better handling of ambiguous reference anchors (select the best one when there are more anchor candidates).
- Better calculation of confidence when the template contains only two positional anchors.
- Better calculation of confidence when the template contains reference anchors but does not contain positional anchors.

# 22

## 3.7.4 - PDF MRC output improvements

Better layout separation for smaller MRC PDF size.

## 3.7.5 - PDF/A1a /2a improvements

Contains all the fixes applied in v19.00

## 3.7.6 - Formatted Layout Header/Footer detection improvement

Better header / footer detection

## 3.7.7 - BAR engine development

New setting `kernel.OcrMgr.BarCheckBarTypes.Enable.C39_C39EXT` is introduced for automatic barcode recognition in case of BAR\_C39 and BAR\_C39\_EXT barcode type. If this setting is true and BAR\_C39 and BAR\_C39\_EXT barcode types are enabled at the same time, the recognition engine automatically decides which barcode type is more likely to be encoded in the barcode.

## 3.7.8 - ReclproPlus VS2005 support binaries removed

**VS2005 ISApp** and **VSApp** samples also removed

## 3.7.9 - Manual Activation

Web Page: <http://activation.scansoft.com/csdkactivation>

## 3.7.10 - New command line switch to get the hardware finger print used by Nuance Licensing

`nuancelt.exe hwfp ->` creates a `csdk_hwfp.txt` file under the user's my documents folder.

## 3.7.11 - New CSharp Sample Application for IProPlus/Nuance.OmniPageSDK.IproPlus and some Visuals/Nuance.OmniPageSDK.ToolBoxes Assemblies

`[CSDKInstallPath]\Samples\HowCouldWeUseIt` is a demo application for the usage of IProPlus and some parts of the Visuals. Load, build and run the sample to see the sources.

Ignore the source control warning dialog when opening `HowCouldWeUseIt.sln` for the first time

## 3.7.12 - FM\_ASIAN Filling Method

The use of this filling method is not necessary. In most cases there is no need to manually specify zone parameters. In the few remaining cases you may use FM\_OMNIFONT even for Asian.

# 23

## 3.8- Version 19.00 New Features

### 3.8.1 - Camera Image handling improvements

- Obtaining camera flag from EXIF info (to determine whether an image came from a digital camera)
- Auto-deskew option can be turned on or off (if on, the program determines whether to apply 2D or 3D deskewing: for normal or camera images)
- Automatic resolution (DPI) estimation
- ½ resolution enhancement (automatic preprocess step: if a camera image contains very large characters, the `II_BW` image resolution is decreased to an OCR engine optimized value)
- Shading correction: new shading correction algorithm is performed by the automatic binarization
- Modified workflow for camera images for better OCR results (resolution enhancement and text line straightening are applied to images coming from a digital camera)

### 3.8.2 - Major update on BAR engine; several new bar types supported

#### 3.8.2.1 - Existing 2D Types Newly Supported By BAR

---

**Note:**

The following types used to be supported by BAR\_AMP only.

---

- [BAR\\_QR](#): QR Code (Quick Response Code); can encode Japanese Kanji and Kana characters, music, images, URLs, emails. De facto standard for Japanese cell phones. These codes are also the most frequently used type to scan with smartphones.
- [BAR\\_DMATRIX](#): Data Matrix; can encode text or numeric data, used in the food industry in autocoding systems to prevent food products being packaged and dated incorrectly. Increasingly used throughout the United States.

#### 3.8.2.2 - Existing 1D Types Newly Supported By BAR

---

**Note:**

The following types used to be supported by BAR\_AMP only.

---

- [BAR\\_A2of5](#): Airline 2 of 5; airline tickets and baggage handling
- [BAR\\_2of5](#): Standard 2 of 5, Industrial 2 of 5; used in the photofinishing and warehouse sorting industries, as well as sequentially numbering airline tickets. Digits are encoded with 5 bars, 2 of which are always wide.
- [BAR\\_C93](#): alphanumeric, variable length symbology; used primarily by Canada Post to encode supplementary delivery information.
- [BAR\\_C39\\_NSS](#): Code 39 without start-stop delimiters (denoted '\*')
- [BAR\\_MAT25](#): Code 2 of 5 Matrix; usage of code 2 of 5 Matrix has strongly decreased; it has high information density, but low tolerance and also not self-checking.
- [BAR\\_PATCH](#): set of 6 barcodes used to separate documents during scanning; does not actually contain data; it is simply a code which provides information to document processing.
- [BAR\\_PLANET](#): Postal Alpha Numeric Encoding Technique (PLANET) barcode; used by the United States Postal Service to identify and track pieces of mail during deliver.
  - identifies mail piece class and shape
  - identifies the Confirm Subscriber ID
  - includes up to 6 digits of additional information that the Confirm subscriber chooses, such as a mailing number, mailing campaign ID or customer ID
  - ends with a check digit

### 3.8.2.3 - New 1D Barcode Types Supported By BAR Only

- [BAR\\_CODE11](#): also known as USD-8; used primarily in labeling telecommunications equipment
- [BAR\\_ITALIAN25](#): also known as Italian Post ITF 2/5, Posteltaliane Registered Mail Barcode, Posteltaliane ITF; used by Italian Post Office (Posteltaliane) in Registered Items product; a service providing legal certification of delivery and proving that items have been sent. Particularly useful for participation in public tenders or competitions, as well as for administrative and legal purposes. The Italian Post 25 is based on Interleaved 2 of 5 (ITF) standard.
- [BAR\\_MSI](#): based on the original Plessey Code. MSI, also known as Modified Plessey, is used primarily to mark retail shelves for inventory control.
- [BAR\\_BOOKLAND](#): It is a real EAN-13 barcode with 978 and 979 prefix that follows a specific format ; it is used for book ISBN numbers. Our implementation enables the ISSN number as well whose prefix is 977.
- [BAR\\_ITF14](#): GS1 implementation of an Interleaved 2 of 5 bar code to encode a Global Trade Item Number; ITF-14 symbols are generally used on packaging levels of a product, such as a case box of 24 cans of soup. Also known as UPC Shipping Container Symbol. It is a fix 14 length ITF barcode with mandatory checksum. The check digit is part of the symbol structure.
- [BAR\\_EAN14](#): used for traded goods; also known as Shipping Container Code SCC-14. It is fix 14 length UCC 128 barcode with prefix 01. Its structure is: 01 [13 digit data] [1 digit checksum]. Therefore the real result string length of this symbology is 16.

# 25

- [BAR\\_SCCC18](#): (Serial Shipping Container Code); used throughout the supply chain as an identifier for item tracing and internal control; It is fix 18 length UCC 128 barcode with prefix 00. Its structure is: 00 [17 digit data] [1 digit checksum]. Therefore the real result string length of this symbology is 20.
- [BAR\\_DATABAR\\_LIMITED](#): GS1 DataBar Limited, RSS-14 Limited; designed for very small item identification and mainly used in the healthcare industry, specifically designed to be read by wands and handheld laser scanners, and cannot be read efficiently by omni-directional scanners
- [BAR\\_DATABAR\\_EXPANDED](#): GS1 DataBar Expanded, RSS-14 Expanded; expanded version of the DataBar barcode symbology which is capable of encoding the 14-digit GTIN, and additional data such as expiration dates, country of origin and serial numbers - up to 74 numeric or 41 alphabetic characters.
- [BAR\\_4STATE\\_ONECODE](#): USPS Intelligent Mail; allows both PLANET and POSTNET barcode information to be combined into a single barcode to track mailings, request address-quality services (including updated address-change information) and return-mail service.
- [BAR\\_4STATE\\_AUSTRALIA](#): used by Australia Post for Postal code and automatic mail sorting

## 3.8.2.4 - Types Supported By BAR\_AMP Only

### Note:

The following types are not supported by BAR.

- BAR\_4STATE\_DK1
- [BAR\\_AZTEC](#)
- [BAR\\_C32](#)

The BAR\_4STATE barcode family includes the following barcode types:

- OneCode
- AustraliaPost
- RoyalMail
- RoyalDutch
- Singapore
- CanadianPost

## 3.8.3 - PDF splitter and Output file splitter

Two new settings are implemented: one for the PDF splitter and the other for the output file splitter. The output file splitter setting specifies the maximum number of pages in each output while the PDF file splitter setting specifies the maximum size of each PDF. If both are given, the splitter splits the PDF/output that comes first.

for PDF Image output:

```
Kernel.IMF.PDF.SplitMaxSize
```

# 26

```
Kernel.IMF.PDF.SplitMaxPages
```

for DTXT/PDF output:

```
Kernel.DTXT.Pdf.SplitMaxSize
```

```
Kernel.DTXT.PDF.SplitMaxPages
```

for all Formatted outputs:

```
Converters.Text.ConverterName.SplitMaxSize
```

```
Converters.Text.ConverterName.SplitMaxPages
```

## 3.8.4 - JBIG2 lossy compression quality improvement

Default compression is set to lossless (user can set it back to lossy); developed better shape detection to avoid unnecessary character changes.

## 3.8.5 - Major update on Form Technologies

### 3.8.5.1 - Form Recognition

Forms exist in many different types and layouts and it is a common task to automatically identify and/or extract data from filled forms. OPCSDK's form handling features allow developers to create applications that perform such batch-like form processing tasks.

Use the Form Template Editor tool to prepare forms for automated processing by these programs. As input, the application can take many different image file types and various PDF flavors.

The Form Template Editor lets you

- create form page templates from your scans and files.

Do this by:

- defining the type and position of form input fields by drawing fillable field zones
- drawing anchor zones around some pre-printed static form content: these aid the processing program to identify specific forms and adjust the positions of the fillable fields to follow possible image distortions (slipping, etc.)
- organize templates into Template Libraries: these can contain a number of form templates related to a specific task
- export the designed Template Library to a specific file format that contains only the definition of the template zones. The exported library is used later on by the form processing program.
- load test images (filled forms) and use the testing features to see how they match the templates in a Template Library
- extract the data from actual filled forms after successful testing

Form Template Editor operates in a two-step process: template definition step and processing step.

### 3.8.5.2 - Template Definition Step

Scanned empty or filled form images can be used for template creation using the new Form Template Editor standalone desktop application. The Logical Form Recognition (LFR) module helps the creation phase by automatically detecting form fields. The created form templates can be tested within the Form Template Editor.

### 3.8.5.3 - Processing Step

During form processing, OPSDK automatically selects the appropriate template for each scanned image. It matches the selected template to the actual image (form registration), recognizes the form fields, and converts the captured data to the selected output formats.

Form templates and form template libraries can be used for automatic Job Separation as well. In this case form templates should contain one or more Selection Anchors only.

## 3.8.6 - New ePub output format

ePub output format to support ebook applications from Apple, Android, Nook, Sony and Kobo.

```
Converters.Text.ePub
```

```
Converters.Text.ePubSimple
```

```
Converters.Text.ePubPoem
```

## 3.8.7 - Thai engine integration

Text, XML and searchable PDF output only.

Image processing integration: despeckling, auto orientation and deskew, auto zoning. Auto orientation, deskew, zoning: These modules are not designed for Thai and there are no plans to adjust them to Thai. Only normal orientation, no skew is handled. Zoning is done by our Legacy algorithm which does not support magazines well. Additional Language support file is `ttfth.rec`

## 3.8.8 - New Assemblies

### 3.8.8.1 - New Assembly For 1<sup>st</sup> Class .NET Support To IproPlus

Add Reference to `Nuance.OmniPageSDK.IproPlus.dll` assembly to access IproPlus in your .NET Form application. See the `NewISAppCSharp100.exe` sample application.

### 3.8.8.2 - New Assembly For 1<sup>st</sup> Class .NET Support To Visuals

Add reference to `Nuance.OmniPageSDK.ToolBoxes.dll` assembly to access Visuals in your .NET Form application. See the `NewVSAppCSharp100` sample application.

# 28

## 3.8.9 - Layout Retention and Document formatting improvements

### 3.8.9.1 - Make PDF File Searchable

A new workflow step is available inside DocuDirect. Input is a set of PDF files of any flavor. This step discovers image-only parts or pages, runs OCR and adds the text to the PDF, leaving notes and annotations intact. This functionality exists in the 'eDiscovery Assistant for searchable PDF', but is now available for a job, allowing for timed, recurring and unattended job running, plus the use of watched folders.

### 3.8.9.2 - Electronic Book Support

The widely-used file type ePub is now supported with three output converter choices. Export your scanned documents or image files to your favorite portable devices. This augments the existing ability to save texts to Amazon's Kindle book reader.

## 3.8.10 - Augmented audio support

The premium speech product Nuance® Vocalizer Expressive™ is supplied with Nuance OmniPage Capture SDK 19. It lets you export audio to MP3 files in English, French and German – listen to your documents on the go! The existing Nuance RealSpeak Solo® remains available with support boosted from 8 to 12 languages (or 9 to 14 voices) with the addition of Japanese, Polish, Russian, Turkish and Australian English. It allows recognized texts to be read aloud, and also provides for saving to MP3 files.

## 3.8.11 - Support for Windows 8 and Office 2013

Nuance OmniPage Capture SDK 19 runs with this latest operating system and its applications.

## 3.8.12 - Obsolete Formatted outputs removed

- `Converters.Text.Excel97`
- `Converters.Text.Word2000` (native Doc format, replaced with `Converters.Text.RTF2000`)
- `Converters.Text.Word97` (native Doc format, replaced with `Converters.Text.RTF2000`)
- `Converters.Text.RTF97`
- `Converters.Text.RTF6`
- `Converters.Text.RTF2000SWord`
- `Converters.Text.eBook`
- `Converters.Text.MicrosoftReader`

## 3.8.13 - WIA2 support

Native Rsd support for WIA2 image input interface

# 29

## 3.8.14 - MP3 Audio output format

- `Converters.Text.Audiomp3` added as a new converter
- `Converters.Text.Audio` modified to create MP3 instead of WAV

## 3.8.15 - DocX, XlsX, PptX improvement for Google Docs and Apple Pages

Better aftermarket OXML application support

## 3.8.16 - Handling 32-bit bitmap input images

## 4 - RecAPI

The OmniPage Capture SDK 20.1 lets developers access a broad range of algorithms and workflows developed and utilized by Nuance Communications, Inc. as well as integrate them into their own applications. Version 20.1 comes with a series of new features, as detailed in the previous chapter ("[New Features and Changes](#)" on page 14). This entails the introduction of a small number of new RecAPI functions, but the underlying architecture is unchanged from version 16 to 20. Therefore, the term Nuance OmniPage Capture SDK (without version number) used in this chapter refers to versions 16 or 20, Nuance OmniPage Capture SDK 20.1 refers to version 20.1 only.

Throughout the documentation we frequently refer to Nuance OmniPage Capture SDK as Engine meaning all the modules of RecAPI.

This chapter is intended to give you an overview of RecAPI, discussing the following:

- Introducing RecAPI
- KernelAPI Modules
- RecAPIPlus
- Alternate Linking of RecAPI

For those upgrading to version 20.1 from a pre-Nuance OmniPage Capture SDK 15 version, a Porting Guide is available on demand in PDF format, found on the Nuance website, providing guidance on the modifications necessary to port applications from version 12 to 16 to 20.1.

### 4.1- Introducing RecAPI

RecAPI is a traditional, standard native 'C' API separated into two layers: KernelAPI and RecAPIPlus. This simplifies the architecture and also makes the toolkit's offering more flexible: higher speed processing to simple text output or more thorough processing with formatted output. RecAPI also provides a base for RecAPI P/Invoke assembly for .NET managed applications.

A higher layer set of services is provided by IPRO and RecIPRO - with an updated object model. IPRO serves as the base for a range of visual components. To access IPRO and MediumWeight Visuals documentation, use their own Help systems or see "[IPRO](#)" on page 43 and "[Visual Components](#)" on page 66 of this Guide.

Here are the main RecAPI components:

#### 4.1.1 - KernelAPI

This is a page-oriented interface providing plain text output (DirectTXT) with high throughput. It has three page parsing choices: Legacy, Fast and Standard.

The latter is the default and gives the most thorough and accurate results.

#### 4.1.2 - KernelAPIS

This is a wrapper for KernelAPI to provide it with alternate linking, so the application and the Nuance OmniPage Capture SDK engine can be separated, to provide more robust functioning.

# 31

## 4.1.3 - RecAPIPlus

This is a document-oriented extension to KernelAPI that provides a full range of formatted outputs. It includes document-level layout consolidation and has the same page parsing choices as KernelAPI. It allows multi-page high-volume processing to be done efficiently, particularly on multi-core systems. It offers composite one-step functions to perform a succession of tasks from just one command.

## 4.1.4 - RecAPIPlus S

This is a wrapper for RecAPIPlus to provide it with alternate linking, so the application and the Nuance OmniPage Capture SDK engine can be separated.

## 4.1.5 - Object Oriented RecAPI model support for .NET users

This is a P/Invoke to bridge between native and managed applications. It allows the functionality of KernelAPI and RecAPIPlus to be presented through a unified interface.

The RecAPIPlus part of RecAPI contains a user interface based on a document-oriented approach. You can implement functionality similar to that of OmniPage Ultimate via just a handful of functions. When exporting multi-page documents you can use a series of output converters to provide layout retention quality equal to that in OmniPage Ultimate. Use the `RecProcessPagesEx` function to process many multi-page image files in one step. In this case, Nuance OmniPage Capture SDK runs processing on several image preprocessing and OCR threads, enabling you to make the most of HyperThreading and MultiCore processors and multi-processor computers. Getting high quality page layout retention and document level formatting adds somewhat to total processing time.

The KernelAPI part of RecAPI is useful when processing speed is a critical factor and/or there is no need for layout retention. This is the basic API with page-level functions. Therefore, this is very similar to APIs of earlier Nuance OmniPage Capture SDK versions. We recommend that you use it for applications that do not require total format retention and rich output formats. KernelAPI generates its simple outputs directly from HPAGE. Here you can choose `TXT`, `CSV`, two different `XML` types, or `PDF` (image on text). Since page formatting is not present here, page-oriented processing on the KernelAPI level gets faster.

In both cases you can choose between three page parsing algorithms: Fast, Legacy or Standard. The latter is default and requires more time but delivers the best possible result, improving both recognition accuracy and format retention. Since the Standard page parsing package occupies some 20 MB, the Distribution Wizard offers it as a separate component.

## 4.2- KernelAPI Modules

The following section gives you an overview of the KernelAPI modules.

### 4.2.1 - General Operations Module

This module contains the basic types and functions necessary to use the general services of the KernelAPI. Initialization is an essential prerequisite for utilizing KernelAPI functionality. User authorization can also be

# 32

performed in this module. You can get useful information about the OmniPage Nuance OmniPage Capture SDK (for instance accessible modules, etc.).

Most RecAPI operations work through Settings Collections. A Settings Collection acts as a kind of profile. It stores a set of operational settings (handled by the Settings Manager Module).

Distinct Settings Collections can store different values for given settings. A Settings Collection is authorized by an identifier that must be passed to all operations using that Settings Collection. Settings Collections can be user-created or deleted.

## 4.2.2 - Error Handling Module

The majority of RecAPI operations have the return value type `RECERR`. This is an enumeration type collecting all possible error codes in OmniPage Nuance OmniPage Capture SDK. The integrating application should always check the return value of RecAPI functions to make sure that the function has successfully completed its operation. Use this module to obtain information about any errors that occurred in the last operation and to see their categories.

## 4.2.3 - Settings Manager Module

This is the manager of settings at the KernelAPI level. The settings are organized into a tree. Every node of the tree has a symbolic name, which is determined by its path in the tree. These names can be divided into parts along their point separators. Every section of the name between two points (dots) corresponds to an individual node of the path. The sections of the name of a given node appear in the correct order to denote all the nodes in the path going from the root of the tree to the given node. The name of the root is an empty string. There may be non-setting nodes in the tree. Nodes (and therefore settings) can be accessed via handles.

For example, here are some paths and nodes relating to image deskewing and despeckling. Double-border boxes denote nodes with a setting; single border boxes denote nodes without settings.

Two symbolic (path) names are shown as examples, for nodes with settings. Each setting can take only one value at a time, but different values can be stored in different Settings Collections, as the example shows for the deskew mode.

The RecAPI online help lists all the settings in the whole tree. See the topics: 'List of all the RecAPI Settings' and 'Converter Settings' under 'Related Pages'.

The settings of StsMan can have a number of types: integer (including int, bool and enum), double, integer array (including int, bool, enum and set array), double array, string and Unicode string.

DevKit users can create their own settings, store them in the tree and maintain them in the same way as existing settings. Every setting has a default value, which cannot be changed during the whole lifetime of that setting. It can be set only when it is created. On the other hand, every setting has a current value, which can be different in various Settings Collections. The current value can be changed anytime in any collection. Settings values from any collection can be saved into files, and loaded from them later.

## 4.2.4 - Image File Handling Module

This module supports loading and saving different image file formats. During loading, the system can automatically detect the file format and compression method of the image files without using file extensions.

# 33

Both this and the Image Handling Module support B/W, grayscale and color images. Nuance OmniPage Capture SDK can read the following multi-page image file formats: `DCX`, `MAX`, `PDF`, `TIFF` and `TIFF-FX`. Loading input can be an image file, or the memory of the integrating application. After the file is loaded, the image is forwarded to the Image Handling Module. Saving input is also provided by the Image Handling Module.

To achieve faster image file handling for multi-page TIF and PDF files, call `kRecOpenImgFile` prior to calling `kRecLoadImg`. This way there is no need to lose time opening the same image file repeatedly for each page. The open, load, close (`kRecOpenImgFile`, `kRecLoadImg`, `kRecCloseImgFile`) functions are connected by the `HIMGFILE` handle in an already familiar way. To load single-page images, or to load single pages from multi-page files, you can use the `kRecLoadImgF` function, thus the need to use the `HIMGFILE` handle here is eliminated.

## 4.2.5 - The Scanning Module

The scanning subsystem supports a wide selection of scanner models through TWAIN or WIA. For conditions of implementing ISIS support, contact your Nuance sales manager. Preparing the scanning subsystem requires the following steps:

- The scanning subsystem must be initialized for actual scanning. Any other scanning function can only be called after the successful initialization of the scanning subsystem.
- The scanner type must be set through the Scanner Setup Wizard (can be accessed directly through the API).
- The Scanner Setup Wizard supports a majority of the popular scanner models. For an up-to-date list of the supported scanners, visit <http://support.scansoft.com/compatibility>. You can update your scanning system through the Wizard by downloading a hint file from the Internet.
- The scanning subsystem requires an additional initialization besides that of the Engine, done with `kRecScanInit`.
- Both the Scanner Setup Wizard and the entire scanning subsystem can be distributed along with your integrating application.
- Having initialized the scanning subsystem, images coming from the scanner device are forwarded to the Image Handling Module.

---

### Note:

During scanning the integrating application should disable all functionalities that can interfere with the scanning. For example, the application should not allow the user to terminate it by any means. The whole scanning module is now thread safe, so it is possible to initiate two scanning processes in parallel, using two Scanners. Calls from two different threads to a single Scanner are queued.

---

## 4.2.6 - Image Handling Module

The Capture SDK Engine is page oriented: the basic unit used in processing is typically the image of the page. Most functions require a page as an input parameter, but some API calls also operate on rectangular areas of

# 34

the image (zones). The Image handling module enables the Engine to handle several pages at the same time. The Engine can only manage and process pages within its own memory space: images must be loaded into the Engine first (by the Image File Handling Module, or the Scanning Module). Possible image sources include files, scanning devices or the memory of the integrating application. The best resolution for B/W images is 300 or 400 dpi. For grayscale or color images the optimal recognition resolution is 150 to 300 dpi.

Nuance OmniPage Capture SDK offers pre-recognition functions to apply image preprocessing procedures to images. These enhance their quality, and yield more accurate auto-zoning (carried out by the Zone Handling Module) and recognition (carried out by the Recognition Module). Preprocessing can include any or all of the following steps:

- Inversion: automatic, programmed or none
- Rotation: automatic, programmed or none
- Deskewing: automatic, programmed or none
- 3D DESKEW: automatic, programmed or none
- Use of SET tools via the ActiveX Image Viewer Control (IVC)

In addition, image despeckling and resolution enhancements are performed internally to render better results, with or without the above transformations.

Additional tasks in this module include

- user modification of the loaded images,
- detection and cutting book pages (when the image contains two book pages),
- auto-detection of table location and structure for the Table Recognition Module,
- line information detection,
- auto-detection of border, box or frame location and styling.

## 4.2.7 - Zone Handling Module

The zone is a rectangular area or the union of specifically located rectangular areas in the page. The upper limit of its dimensions is full page size. It also contains a feature of interest to the user. The union of rectangles must have a so-called pizza box shape: the top of each rectangle in the union must touch the bottom of the rectangle above it. A pizza box-shaped zone is a compound and irregular zone.

The image data covered by each zone is handled and processed (typically recognized) separately, according to zone-specific parameters.

This module can handle two types of zones in separate zone lists of each page: user zones and OCR zones. Zones can be added to the proper zone list of any given HPAGE in three different ways:

- add zones automatically (auto-zoning) - adds OCR zones
- add zones manually (by specifying the zone coordinates and attributes) - adds user zones, or
- add zones from a zone file (created by the user through API functions) - adds user zones.

# 35

You can start auto-zoning directly or it automatically runs at the beginning of the recognition process provided that there are no OCR zones already defined.

You can also influence the method and performance of auto-zoning via different settings. When you use user-zones, you can specify whether to have auto-zoning also run or not. Individual user zones are transformed into one or more OCR zones prior to recognition. Recognition is ALWAYS carried out on OCR zones and it may also change them due to post-processing.

- Based on their content, zones can be classified as follows:
- flowing text (horizontal or vertical with three rotations)
- table (handled by the Table Recognition Module)
- graphics (no recognition performed)
- form (handled by the Form Recognition Module).

## 4.2.8 - Form Recognition Module

The form recognition module supports two form processing workflows:

- Analyzing empty non-filled forms and converting them into one of several active (fillable) digital form document formats.
- Recognizing the contents of filled forms based on pre-constructed form templates and outputting the respondent data contents in some kind of digital format that can be easily processed.

### 4.2.8.1 - Analysis And Activation Of Empty Forms

The current empty form recognition core is mainly based on the Logical Form Recognition™ (LFR) technology which was the backbone of OmniForm, and is now an integrated part of the OmniPage Capture SDK. LFR recognizes textual and non-textual elements in empty forms - like check boxes, circle texts, comb fields, tables/cells, graphics, lines and boxes. These detected form field zones cannot be modified programmatically, but the page can be viewed and modified in the Text Editor Visual control (TEC).

The aim is to turn a form image into a fillable electronic form that can be distributed to respondents or posted on a website. The data in the returned forms can then be processed by any suitable means, outside the scope of the Nuance OmniPage Capture SDK. Filled form data is expected to be returned electronically, so there is no need here for anchor zones.

#### Typical processing workflow:

1. Load a single-page image of a form.
2. Specify the whole page as a single form page or define one or more separate form type user zones manually.
3. Perform page decomposition. LFR runs inside the defined form zone(s), as part of the Zone Handling Module, auto-detecting form elements and creating active form controls.

# 36

4. Inspect the auto-detected form objects in the TEC visual control. Move, resize or delete them as required. Add new objects if needed. TEC also lets you view and modify form object attributes.
5. Save the results to a format that supports active form elements: formatted XML, RTF (Microsoft Word) and PDF (optionally with Acro-Form-like fillable field definitions).
6. Form object attributes can also be modified in the target application.

## 4.2.8.2 - Recognition Of Filled Forms

The main steps of filled form recognition are form template definition with field and anchor zone placement, template registration / application, based on the anchor and form-element information saved to the form template and recognition of form contents, using the usual recognition tool-set. Theoretically a form template is a set of form-field zones and anchor definitions. Form-field zones are normal input zones with some additional layout information attached; they identify the areas where we require fill-in data on a specific form-page (e.g. fill-texts, combs and checkboxes). Anchor definitions are specified as ignore zones during template definition, and receive some additional information when the template is saved, to assist the matching of the template to incoming forms.

**Typical processing workflows (step two has four alternatives):**

### Creating a form template definition file

A single-page empty form image is loaded. If it is scanned, automatic deskewing is recommended. If from file, any supported image file format is acceptable, including any flavor of PDF - these are all treated as imaged input. The input can also be an active PDF form (see how it is handled under 2d). In all cases, the form should contain distinctive shapes (preferably four) to be used as anchors for incoming forms. These shapes should not be printed in a color that is to be dropped-out programmatically.

#### 1. Manual zoning

User inserts form zones if their exact position on the page is known and also specifies up to four ignore zones that contain the shapes to be used as anchors.

#### 2. Auto-detection of form objects

User calls LFR to identify form objects and then calls the Create Template function to create user zones from the form objects. This may also detect the anchor shapes and create ignore zones for them.

#### 3. XML form object definition

User loads the XML output from OmniPage Ultimate, having used it to process the empty form, optionally having changed the LFR output in the OmniPage Text Editor and exported the page to XML output. This output contains information about form objects. The loading process parses the XML file and creates user zones and anchors based on the file.

#### 4. Active PDF form

User opens an active PDF form, containing suitable Acro-Form controls for both form objects and anchors. In this case LFR does not need to run, just call the Create Template function.

In all cases, the user can access or modify the user zones before they are saved. This can be done in the Text Editor visual control as already described or by KernelAPI calls. In cases 2a-2c, the anchor zones may also be modified. In case 2d (active PDF forms) the anchor zones must not be modified. KernelAPI functions allow

form-field identifiers and attributes to be attached to the user zones. These include restrictions to zone content from the checking subsystem - such as regular expressions, dictionaries, character filters like 'numbers-only', or other user-defined checking routines.

The Save Template function saves the zone set, including the attached items and the anchor bitmaps into a template file. This is a standard Nuance OmniPage Capture SDK zone file extended with anchor data.

#### **Processing filled forms:**

Use the Nuance Form Template Editor for this task. Consult the **How to Use?** section of the **Nuance Form Template Editor Help** for details.

### **4.2.9 - Table Recognition Module**

This module detects tables on the page and links the recognition result and the table information to each other. A table is described by the list of its cells in reading order (from left to right and top to bottom) ordered by their top-left coordinates. Cells cannot have blank spaces between them, they cannot intersect each other, and all cells must be rectangular.

#### **4.2.10 - Recognition Module**

The Engine can load several recognition engines. Licensing determines which ones are available. You can select which available engine to run in a given user zone by specifying a particular filling method suitable for the zone content, or by specifying an actual recognition engine. In addition, you can also influence which recognition module is selected by the Engine for the processing of the individual OCR zones that are created automatically. If no instruction or guidance is given, recognition module selection is automatically determined.

The recognition engines available are as follows:

- PLUS3W: 3-way voting Omnifont engine for machine-printed text
- PLUS2W: 2-way voting Omnifont engine for machine-printed text (default)
- MTX: M/TEXT Omnifont engine for machine-printed text in 12 languages
- MOR: MOR multi-language Omnifont engine for machine-printed text in over 120 languages
- FRX: Fireworx multi-language Omnifont engine for machine-printed text in 56 languages
- DOT: engine for 9-pin or 24-pin draft dot-matrix printouts
- MAT: matrix matching engine for codified scripts (OCR-A, OCR-B etc.)
- HNR: engine for hand-printed digits
- RER: reRecognition (third-party) engine for hand-printed alphanumerical characters
- OMR: engine for optical marks (checkmarks)
- BAR: engine for bar codes
- BAR\_AMP: AmpLib (third-party) engine for 1D and 2D bar codes.

#### **PDF recognition**

# 38

PDF files can be separated into two basic classes: image-only PDF, normal PDF. Recognition of image-only PDF is exactly the same as that of any other image format. The normal PDF contains both image and text data. The Nuance OmniPage Capture SDK can extract this textual information to boost OCR accuracy. In general, text in a PDF file is reliable, so accurate results are possible even when the quality of the imaged text is low. See ["New Features and Changes" on page 14](#) or improvements in PDF handling. The fullest description of PDF functionality in the Nuance OmniPage Capture SDK is found in the 'Related Pages' section of the RecAPI online help system.

## 4.2.11 - Recognition Data Handling Module

In addition to the usual output choices, you can also access raw data providing significantly more information than the character codes and attributes themselves. This data stores all information that can be obtained on characters. For instance, it allows access to alternatives such as character tips or word suggestions and to recognition confidence data from the recognition engine(s) that ran in conjunction with Spell Checking Module output.

## 4.2.12 - Language, Character Set, and Code Page Handling Module

This module is required to be present for successful Engine initialization. This module handles language, character set and code page related settings and their combinations.

## 4.2.13 - Spell Checking Module

The spell checking module consists of the following separate parts:

- Spell checking: using language-specific dictionary elements, it uses third-party language-specific spell checkers. There are two kinds of spell checkers: Language dictionaries (language checking), and Vertical dictionaries (professional dictionaries for a given set of languages).
- User-written checking: using user-written callback functions.

---

### Note:

User-written callbacks are deprecated.

---

You can combine any of these during the recognition process: these are steps in determining the acceptability of words. The use of these checks can still be enabled or disabled separately at the zone level.

The Nuance OmniPage Capture SDK is delivered with over 20 different language dictionary files. These are generic language dictionaries; typically they contain between 100,000 and 200,000 entries.

Vertical dictionaries are based on specific professions. They can be treated as extensions to the language dictionaries, but they can also be used when no language dictionary is specified. The Capture SDK is delivered with nine different vertical dictionaries:

- Dutch Legal Professional Dictionary
- Dutch Medical Professional Dictionary
- English Financial Professional Dictionary
- English Legal Professional Dictionary

- English Medical Professional Dictionary
- French Legal Professional Dictionary
- French Medical Professional Dictionary
- German Legal Professional Dictionary
- German Medical Professional Dictionary

A User dictionary is a list of words , so-called UDItems. User Dictionaries can be created or modified by the user manually or through KernelAPI calls. User-written checking callback functions are parts of the integrating application. They receive the string to be checked and the index of the zone where this string comes from. The application must evaluate the string; express its opinion on the recognized string's acceptability, then return this to the Engine. The Recognition Module uses feedback from the Spell Checker module along with other data to make its assessment of recognition confidence.

---

**Note:**

The confidence reporting system works best when the 3-way voting engine runs. If other machine print recognition modules are used, confidence information is still available, but the system's ability to report on confidence properly is reduced. This results in a higher level of false negative and false positive reporting of suspicious recognition results.

---

#### 4.2.14 - Direct TXT Output Converter Module

This module allows you to convert recognized text simply and quickly. That is, you use the output of the recognition module as is (without reading order and paragraph detection). Therefore the DirectTXT Outputs are simpler than the Layout Retention Output conversions (available in RecAPIPlus) and also faster to produce, because they do not include slow detection processes. The following DirectTXT output types are available:

- DirectTXT Text output: a simple text file.
- DirectTXT CSV output: a comma-separated text file, a simple format to represent tables. Microsoft Excel can read this format.
- DirectTXT Formatted Text output: This converter delivers plain text, but attempts to keep the page layout as detected in the original image: this creates a text file that simulates columns and boxes using tabulators.
- DirectTXT PDF output: contains the whole image of the original page and the text behind the image on a separate layer (image on text PDF). These PDF files especially suit the purpose of page archiving, because they contain both the original image and recognized text.
- DirectTXT XML output: typically used for further processing recognized data. You can easily parse (e.g. `MSXML`) or transform (`XSLT`) the output xml file. The format of the xml output is specified by the same schema as the Layout Retention Xml Output (<http://www.nuance.com/omnipage/xml/ssdoc-schema3.xsd>).

# 40

- DirectTXT Binary output: used for creating files directly from the recognition data without any character conversion and formatting. It is the most usable output format for barcodes containing binary data (for example Code128 or PDF417 barcodes containing encrypted data).

When you specify an already existing file name, the TXT type outputs are appended.

The DTEXT module can be especially useful for applications that do not require formatting but speed is an important factor, for example: indexing, archiving, or some form processing applications. When programming with KernelAPI this is your only output choice. It is possible to purchase distribution licenses that exclude formatted output (see ["Preparing a Distribution File Set"](#) on page 8 of Chapter 1).

## 4.3- RecAPIPlus

RecAPIPlus - as a part of RecAPI - contains a programming interface based on a document-oriented approach and provides formatted page output.

### 4.3.1 - General Service Functions

These functions are responsible for the initialization and termination of Nuance OmniPage Capture SDK, loading and saving RecAPIPlus settings, and setting the count of the OCR threads.

### 4.3.2 - Simple Multi-Page Document Handling

With these functions the User's application can implement document-based solutions. The application can create multi-page documents. The application can insert recognized pages that have been processed using KernelAPI functions. It can also delete or move pages or save and load the document to and from the proprietary OPD file type. When the document is ready, the application can export it into the output file, using the appropriate output format.

### 4.3.3 - One-Step Functions

RecAPIPlus has two distinct one-step functions, i.e. the user's source code contains only one call to such a function that performs the whole workflow. One uses the 123-type workflows from OmniPage Ultimate (loading, image preprocessing, recognition, page and document formatting, exporting). It can process several multi-page image files or it can work from a scanner. During the process, it uses all settings as previously adjusted by the application. The other function executes workflow files created by the Workflow Assistant (see [Intelligent Workflow Runner](#)), the Workflow Applet, the DocuDirect in OmniPage Ultimate or by the ITest component of the Nuance OmniPage Capture SDK.

### 4.3.4 - Layout Retention Output

RecAPIPlus provides complex and accurate layout retention outputs in several file formats like RTF, DOC, WordML, XLS, PDF, WP. (One-step functions also use the output converters of this module.)

In several cases our goal is to retain the original layout in the output document, as much as possible. Different converters have different capabilities for layout retention. There are five output levels:

- Plain Text (previously 'No formatting')
- Formatted Text (previously 'Retain Font and Paragraphs')

# 41

- True Page®
- Flowing Page
- Spreadsheet.

Not every converter can realize each level of output formatting. For example, an outputted Word document or a PDF document can use Flowing Page and True Page levels, which are very similar to the original layout. There are also simple text converters, which can retain only the text (in Plain Text mode) or the text and its attributes (in Formatted Text mode). In addition, final output results can be influenced by adjusting settings. A help topic table lists which formatting levels are supported by each output converter.

## 4.4- Alternate Linking of RecAPI

The KernelAPI and RecAPIPlus interfaces use dynamic linking. This means that all required DLL files are loaded at initialization time. It also requires all DLL files to be placed at a known location. The Nuance OmniPage Capture SDK searches them using the Standard Windows Search Order. This need for the program and its DLL-s to be located at a predetermined path may constrain the distribution of the integrating application. The KernelAPIS and RecAPIPlusS modules solve this problem. These are thin interface layers for binding the Engine at run-time. They load KernelAPI or RecAPIPlus from a path specified in the initialization function and get all the entry points of the required DLL-s. The actual Engine load happens when `kRecInitS` or `kRecInitPlusS` is called. Other DLL files are loaded only when a function requiring their services is called.

Since these modules are wrappers on the corresponding KernelAPI and RecAPIPlus modules, they make it possible to separate the integrating application from the OmniPage Nuance OmniPage Capture SDK files without placing the Nuance OmniPage Capture SDK folder into the environment variable PATH. This can make the application less vulnerable to processing issues raised by the Nuance OmniPage Capture SDK, and also avoids the risk of identically named program files being created in both the application and the Nuance OmniPage Capture SDK.

Depending on how the release or debug versions of the integrating application use the C run-time libraries, one of the following small KernelAPIS or RecAPIPlusS static libraries should be linked to the application:

### Release version of KernelAPIS

- `KrnAPIS.LIB` - single-threaded C run-time library - statically linked,
- `KrnAPIS_MS.LIB` - multi-threaded C run-time library - statically linked,
- `KrnAPIS_MDyn.LIB` - multi-threaded C run-time library - dynamically linked.

### Release version of RecAPIPlusS

- `RecAPIPlusS.LIB` - single-threaded C run-time library - statically linked,
- `RecAPIPlusS_MS.LIB` - multi-threaded C run-time library - statically linked,
- `RecAPIPlusS_MDyn.LIB` - multi-threaded C run-time library - dynamically linked.

The debug version of the integrating application requires a static library to be linked as follows:

# 42

## **Debug version of KernelAPIS**

- `KrnAPISd.LIB` - single-threaded C run-time library - statically linked,
- `KrnAPIS_MSd.LIB` - multi-threaded C run-time library - statically linked,
- `KrnAPIS_MDynd.LIB` - multi-threaded C run-time library - dynamically linked.

## **Debug version of RecAPIPlusS**

- `RecAPIPlusSd.LIB` - single-threaded C run-time library - statically linked,
- `RecAPIPlusS_MSd.LIB` - multi-threaded C run-time library - statically linked,
- `RecAPIPlusS_MDynd.LIB` - multi-threaded C run-time library - dynamically linked.

# 43

## 5 - IPRO

This chapter introduces IPRO. It contains an overview of the structure of IPRO, with particular emphasis on the object model. The object model diagram is followed by brief summaries on the objects themselves. For detailed reference, see the IPRO Online Help system.

The chapter ["New Features and Changes" on page 14](#) of this Guide includes some topics on innovations to the IPRO functionality.

This chapter discusses the following:

- About ReclPRO
- ComKit Libraries
- Error Handling in ReclPRO
- Avoiding Potential Deadlock Situations
- Supported Development Environments
- The IPRO Object Model
- Object Model Overview
- Object Summaries

### 5.1- ComKit Libraries

The ComKit 20.1 libraries are made to support programmers of COM-enabled environments. They replace the IPRO libraries of the version 12 SDK. The new version is tightly integrated with the latest Nuance OmniPage Capture SDK Engine and is based on the OmniPage Ultimate architecture. The version 20.1 object model is largely backward compatible with that of version 16. Because the Nuance OmniPage Capture SDK OCR engine underwent a major revision between versions 12 and 15, the current object model is not backward compatible beyond the previous one.

This document summarizes the COM object hierarchy of the SDK, and the interface details for the specific objects.

Since Nuance OmniPage Capture SDK 15, this toolkit does not support Windows 9x platforms including Windows 95, Windows 98, Windows 98 SE and Windows Millennium editions. These OS platforms cannot be used as development platforms or runtime environments.

### 5.2- Avoiding Potential Deadlock Situations

IPRO (see [Deciding on the Interface](#)) is a highly multi-threaded architecture; the benefit is boosting performance on multi-processor, multi-core and multi-threaded systems. However, because it also massively overlays the Windows messaging architecture (that has limitations in multi-threaded environments), it might be a potential source of deadlock situations if not carefully used. You can find a set of hints below about how to avoid deadlock scenarios when using IPRO, and optionally Visual Toolbox.

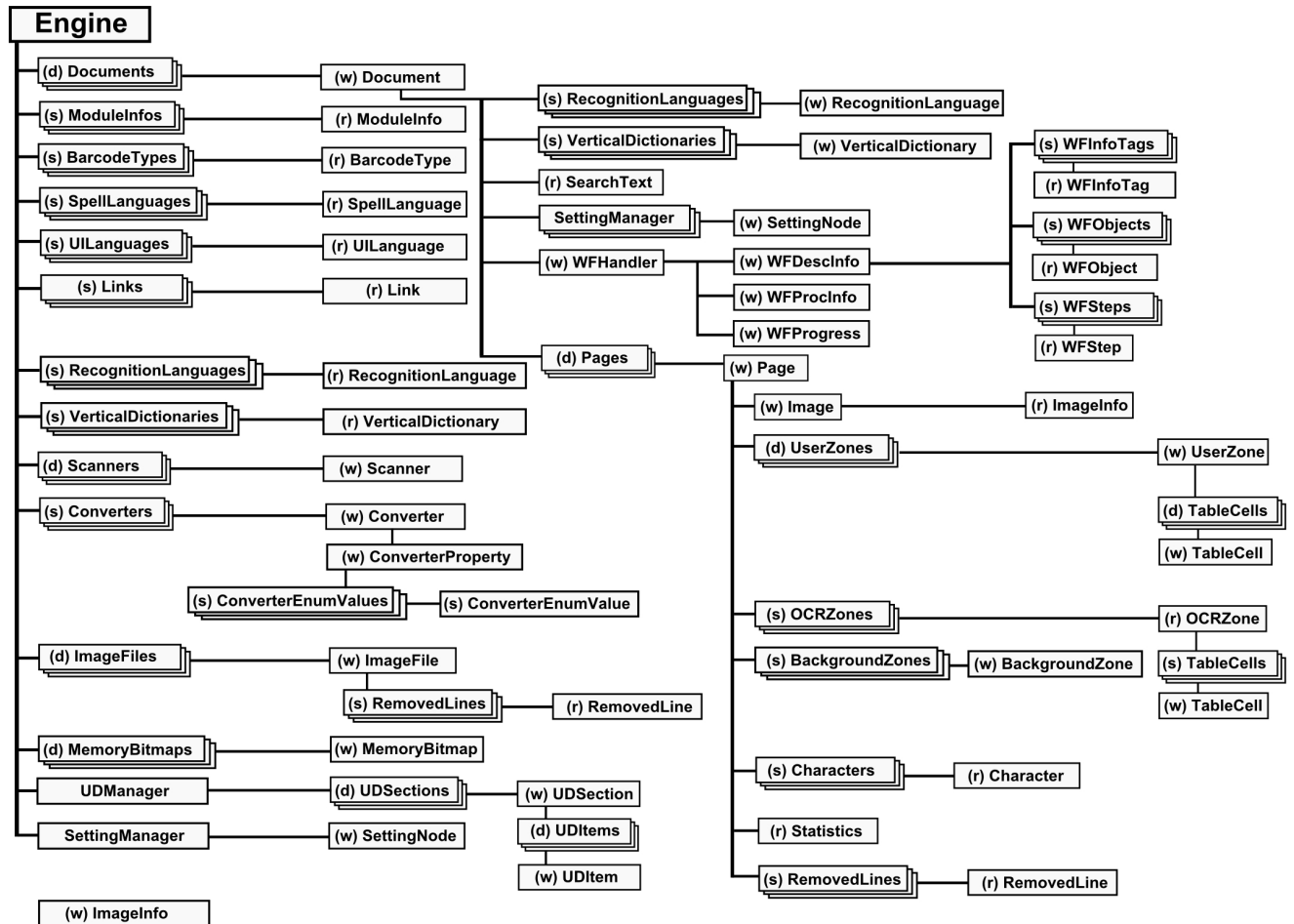
# 44

How to use IPRO and Visual Toolbox in a safe approach:

- Avoid calling back to IPRO in event handlers. Many IPRO events receive IPRO object references. If it is really important, use the received object reference to query basic properties, but never initiate massive processing which causes hard IPRO usage or additional event firing (which might be in a different threading context).
- Avoid calling the SendMessage API targeting windows that were created in different threads. E.g. never send messages from a worker thread to a window that was created in the main UI thread. This is a typical reason for deadlocks in multi-threaded windows applications even without any IPRO, or other COM object usage.
- Avoid accessing objects that were created in a particular thread from a different thread. If possible, encapsulate all IPRO-related object creation and processing into a single thread, or ensure proper interface marshalling if multiple thread usage is necessary and you need to access objects in a different thread than the creator.
- Ensure that during an IPRO call the program does not enter into another parallel call. This might happen, for example, if you have a UI button associated with some processing and click it a second time before the processing initiated by the first click has finished. Because the IPRO Workflow manager has its own message loop to enable the integrator application's UI to be responsive and painted correctly, this type of encapsulation is possible. A good approach is disabling the related UI elements during any processing so that the user cannot click the button at the wrong time even if the interface is responsive at the time.

## 5.3- The IPRO Object Model

The IPRO object structure is a complex, multifunction hierarchical object model which is detailed in this documentation. Below is a comprehensive reference diagram. It can be viewed in larger size in the electronic delivery of this document.



## Legend

Creatable object

Object

Collection

[s] : Static collection (enumerates information but could not be extended or modified)

[d] : Dynamic collection (elements are created in run time, they can be added, changed or deleted)

[r]/[w] : Readable / writable nature of the object. [r] objects can only be read to get information on them; [w] objects can be modified in run time.

# 46

## 5.4- Object Summaries

Below is the brief summary on the characteristics of IPRO objects. For detailed information on each of them, see the IPRO Online Help system.

### 5.4.1 - BarcodeType / Barcode Types

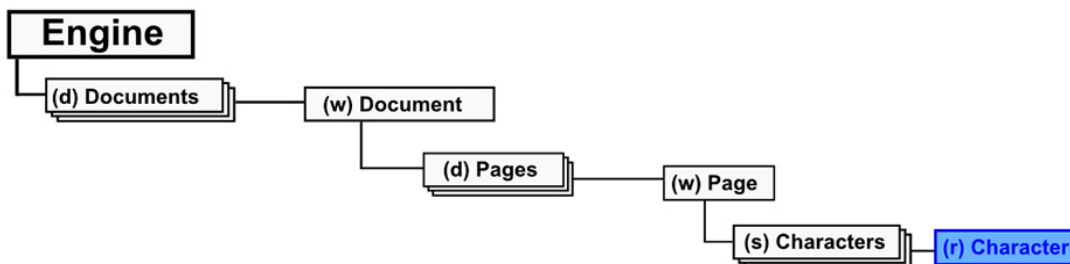


The **BarcodeType** object contains information about the particular barcode type.

The **BarcodeTypes** collection is a sorted set of BarcodeType objects. Each element represents a recognizable barcode type. Using its properties, the BarcodeTypes collection enables you to gather information about the supported bar code types.

This collection serves multiple purposes. Under the Engine objects it provides barcode type information for listing purposes. If the parent of this collection is a Document object, the contained barcodes can be enabled or disabled for recognition. By default the following barcode types are enabled: EAN, ITF, Code 39, Code 128 and Codabar.

### 5.4.2 - Character / Characters



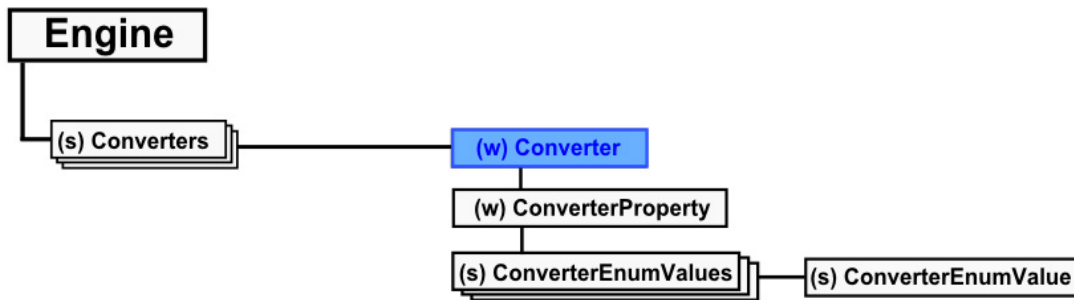
The **Character** object provides information about a single recognized character, such as the character code, coordinates of the character on the image, confidence information etc. The character coordinates are treated as OCR image coordinates.

The **Characters** collection is a sorted set of Character objects, where each element represents a single character in the recognition result. By accessing its properties, the Characters collection enables you to gather information about the recognition result. The collection is sorted by the OCR zone indices (`OCRZoneIndex` property).

## 47

By default, this collection is empty. After the `Recognize` method of the Document or Page object is called, it is filled with the recognition result. The returned Characters can be filtered by zone index and cell number.

### 5.4.3 - Converter / Converters



The **Converter object** provides properties and methods to customize the layout of the output files created by the Document and Page objects' `ConvertResult` method. The converter settings are organized into a tree-structure, and they are available through the ConverterProperty object, which represent a setting node in the tree.

There are three types of converters: text converters, image converters, and form converters. Text converters create a (formatted or unformatted) text file from the recognition results. Image converters create image files from the input images, so they do not require the input pages to be recognized. Form converters are used in form data extraction.

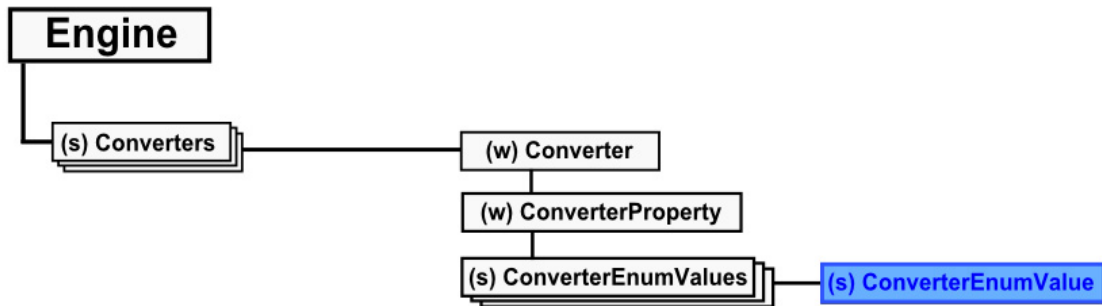
There is a multitude of possibilities to set converter parameters. Use `ConverterProperty` to customize them. To query them, use the `Item` method of the Converters collection. Converter names and parameter setting possibilities are detailed in the RecAPI documentation under List of all Converter settings of RecAPI.

The **Converters collection** is a sorted set of output Converter objects. Using its properties, the Converters collection enables retrieving information about the supported converters.

A converter can be 'cloned' using the `CloneConverter` method. It makes an exact copy of the source converter and is useful to make customized converters for the same output format. The cloned converters can later be deleted using the `RemoveClonedConverter` method if no longer needed. 'Factory' converters cannot be deleted.

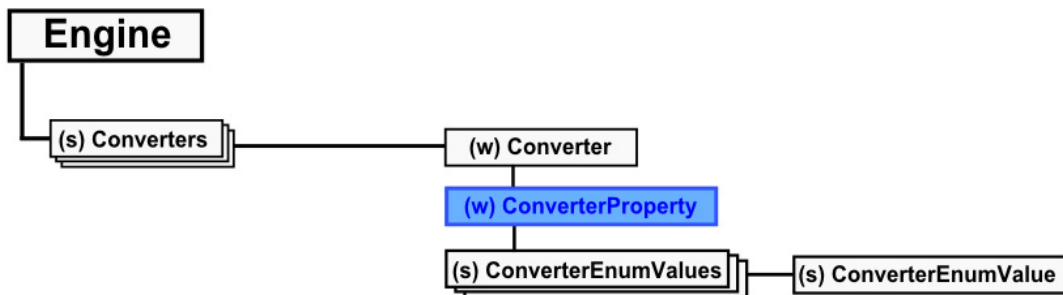
# 48

## 5.4.4 - ConverterEnumValue / ConverterEnumValues



The **ConverterEnumValue** object represents an enum value of a ConverterEnumValues collection. The **ConverterEnumValues** collection contains the available enum values of a ConverterProperty object.

## 5.4.5 - ConverterProperty



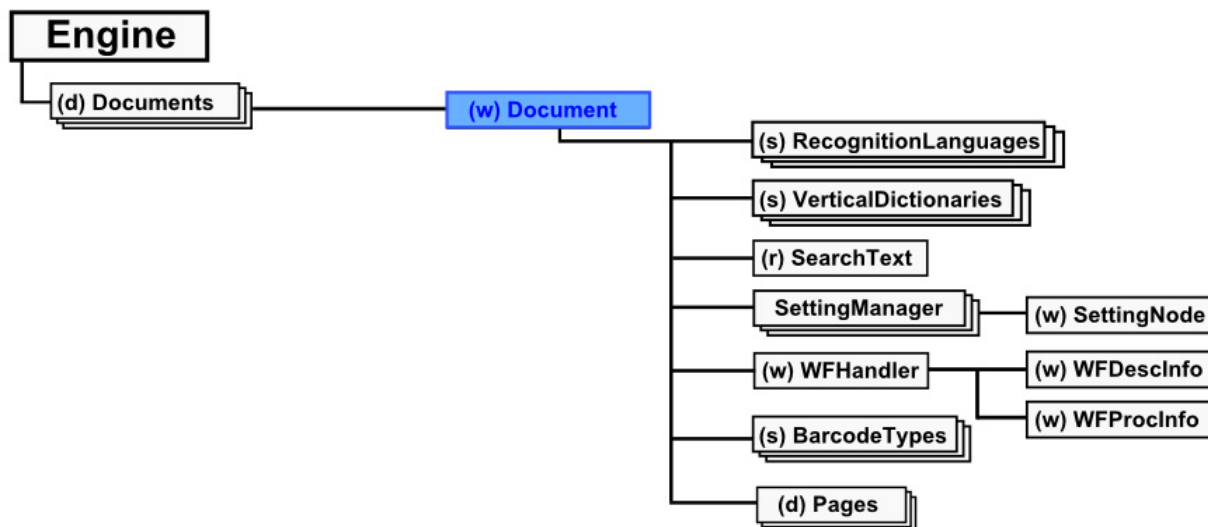
This object represents a single converter setting. The converter settings are represented by a hierarchical tree structure isolating the different setting groups, so that each tree node represents a converter setting which might have an associated value (and/or might have children settings). To read or manipulate a setting that connects to a particular ConverterProperty object, use `Name`, `FullName`, `DisplayName`, `Type`, `Value` and `EnumValues` properties.

### Note:

`EnumValues` returns a valid object only if the related setting is of type `PROPERTY_ENUM` and it can be used to query the valid enum values the particular property supports. In order to access the children settings of the object (if any), use the `Count` and the `Item` properties.

## 49

## 5.4.6 - Document / Documents



The **Document object** represents an IPRO document which encapsulates the data and the parameters required during OCR related procedures. To create a new document, invoke the Documents collection's `Add` method. To open an existing document, invoke the `Open` method. To save a document for later reference, invoke the `Save` method. The Document object has numerous properties and settings. You can save or load all settings by using the `SaveSettings` and `LoadSettings` methods.

---

**Note:**

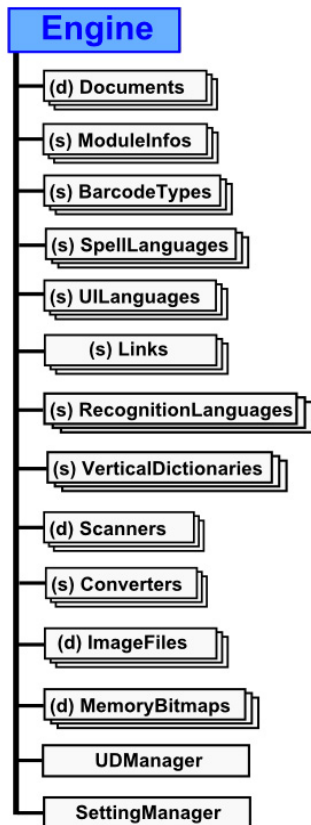
The `Save` method also saves all document level settings to the document file automatically.

---

The **Documents collection** is a sorted set of Document objects, where each element represents a document handled by the Engine. Through its properties, the Documents collection returns information about the currently open IPRO documents.

# 50

## 5.4.7 - Engine



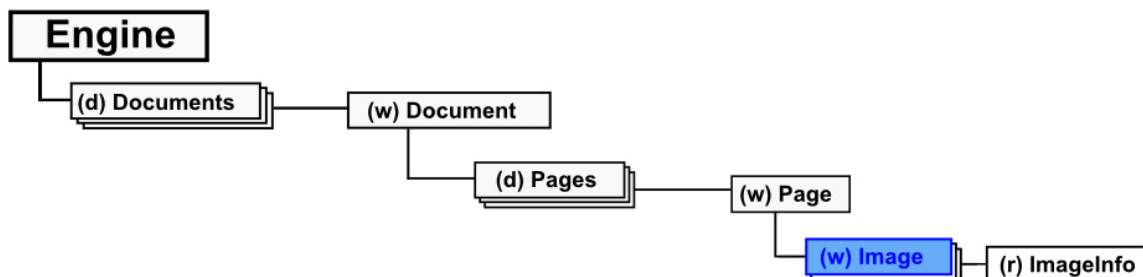
The Engine object represents the main OCR Engine class which serves many purposes such as scanning, image preprocessing, recognition, text formatting and output.

This object and its functions are referred to as the 'Engine'.

The Engine object is the top of the object hierarchy and controls all other objects in the object hierarchy. The integrating application must always explicitly create an instance of the Engine object. When it is created, the Engine object analyzes the configuration of available sub-modules and performs initialization. The valid licensing of the required modules at this time is vital, otherwise either the Engine object creation fails or one or more modules become unavailable.

The Engine must be initialized with the `Init` method. All properties and methods return an error code if initialization is improper.

## 5.4.8 - Image



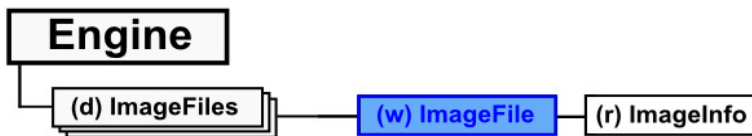
# 51

The Image object stores the page image (original) either scanned, loaded from image file or memory together with the engine-managed additional, generated images (primary (pre-processed), OCR and thumbnail). The role of this object is to facilitate access to the managed images, and also, to provide a basic operation set for image manipulation. Note that a large image manipulation library is available through the ImageFile object.

The original image is the main image, acquired from some external image source. Image preprocessing and image manipulation occurs on that image, resulting in a primary image. The thumbnail image is a smaller version of the current image. The OCR image (black-and-white) is created by the first operation that requires it (preprocessing, page-parsing, recognition). The OCR image is only available after the recognition.

Redaction is a special case for image handling. This operation writes back to primary, OCR and original images (those stored in the program). Only the original image stored in file outside the program remains untouched. In the case of scanning, no unredacted image is available. Therefore we strongly suggest that you apply redaction to a copy.

## 5.4.9 - ImageFile / ImageFiles



The **ImageFile** object represents an image file in the file system. Use this object to get information on, manipulate or convert image files using a wide set of image manipulation methods. Note that most image-processing methods are available only for particular multi-page tiff images as documented in the descriptions for the particular methods. To process other images using the available methods, save them in tiff format first, perform the operations and convert them back to the original format.

The supported image resolution is in the range of 75-2400 DPI. When accessing bitmap pixel data, every image stored is in bottom-up line order and the pixel order is BGR.

The **ImageFiles** collection contains the list of the available ImageFile objects. Using this collection, the available images can be loaded, enumerated or closed. You can find details about the image file manipulation and usage in the description of the ImageFile object in this document.

## 5.4.10 - ImageInfo



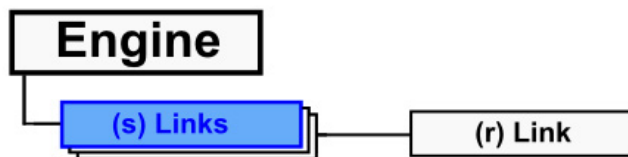
The **ImageInfo** object provides basic image information (size, resolution etc.) about the page image. Although this creatable object has read/write properties, these properties cannot be used to change the properties of an existing image or image file.

Images in IPRO are always retrieved in the following format:

# 52

- byte order: BGR
- line order: bottom-up
- line padding: double word (4 bytes) aligned

## 5.4.11 - Link / Links



The role of the **Link object** is to provide workflows with loading sources and saving targets that are not present on the local computer. These links (file sources and destinations) include document management systems (such as different versions of SharePoint, ODMA) and FTP locations.

The **Links collection** contains the list of the available Link objects.

## 5.4.12 - ModuleInfo / ModuleInfos



The **ModuleInfo object** contains information about a specific Engine recognition module even if the particular module is not available due to the lack of the appropriate licensing. Use the `Version` property to decide whether the module is available or not. Use the `InitCode` property to determine the reason if the module is unavailable.

The **ModuleInfos collection** is a sorted set of ModuleInfo objects. Each element in the collection represents an Engine module. Using these objects you can retrieve information about the available Engine modules.

Some modules must always be distributed with the integrating application, since their existence is a prerequisite of any Engine operation. Others, such as some recognition modules, the scanning module, etc. must be distributed only if the integrating application requires their services. This flexibility allows the integrator to scale the capability of the distribution file set for disk footprint sensitive applications. Using this collection can be used to log diagnostic information to help solve problems or to provide users with runtime feedback when an application is designed to optionally use a variable set of Engine modules.

## 53

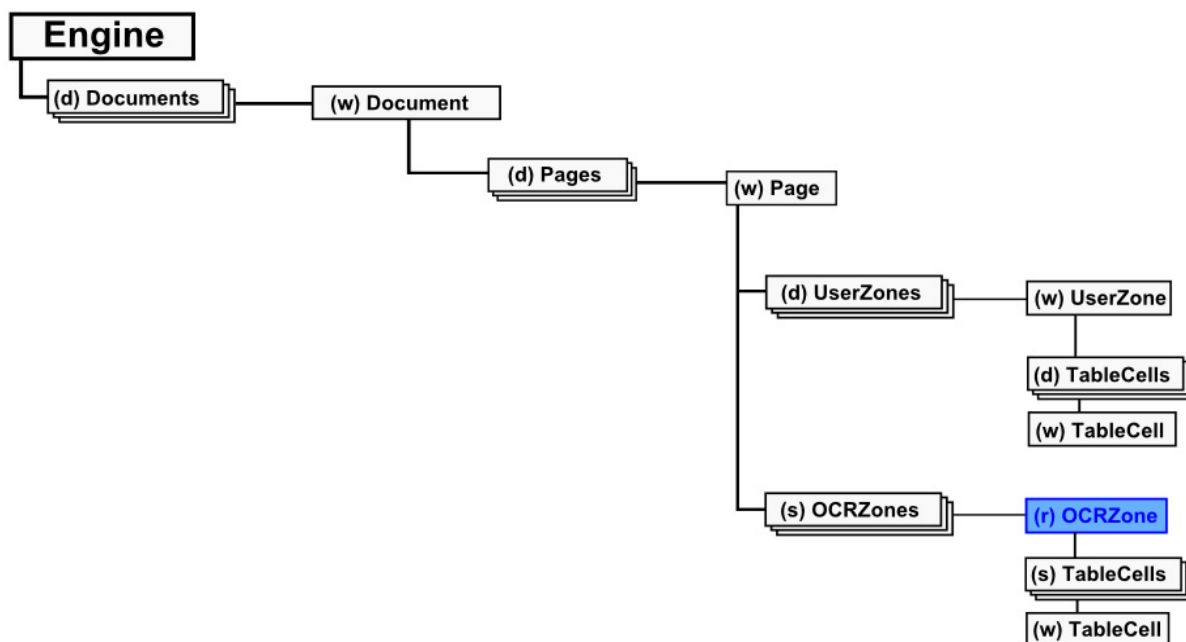
## 5.4.13 - MemoryBitmap / MemoryBitmaps



The **MemoryBitmap object** represents an image stored in the memory. This object manages bitmaps as raw data. This object can be passed to the `LoadBitmap` method of the Document object. This allows the integrating code to load an image which is stored in the memory into a particular document. All bitmaps stored by the IPRO object model are bottom-up, BGR color order, and double-word aligned bitmaps.

The **MemoryBitmaps collection** is a sorted set of MemoryBitmap objects. Using this collection, bitmaps stored in memory can be created and destroyed.

## 5.4.14 - OCRZone / OCRZones

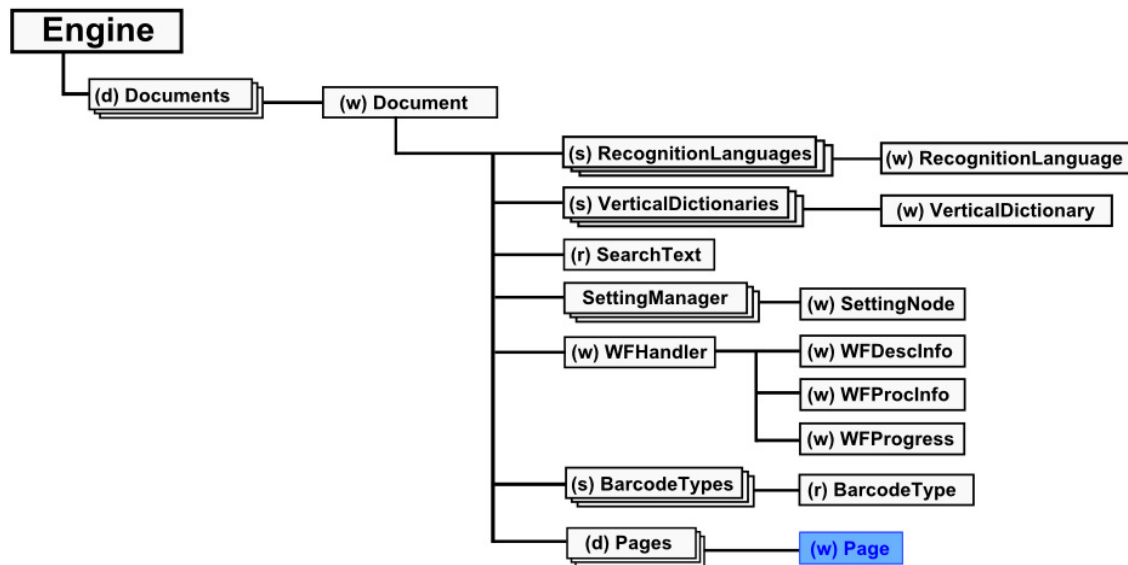


The **OCRZone object** contains information about a specific OCR zone in a page. The OCRZone object provides access to the coordinates of an OCR zone, its content type, etc. All coordinates in this object refer to the OCR image.

The **OCRZones collection** is a sorted set of OCRZone objects. The order of the OCRZone objects is the same as the reading order of the recognized paragraphs.

**Note:**

The reading order takes effect only when exporting with converters using the Formatted Text formatting level (formerly Retain Fonts and Paragraphs mode).

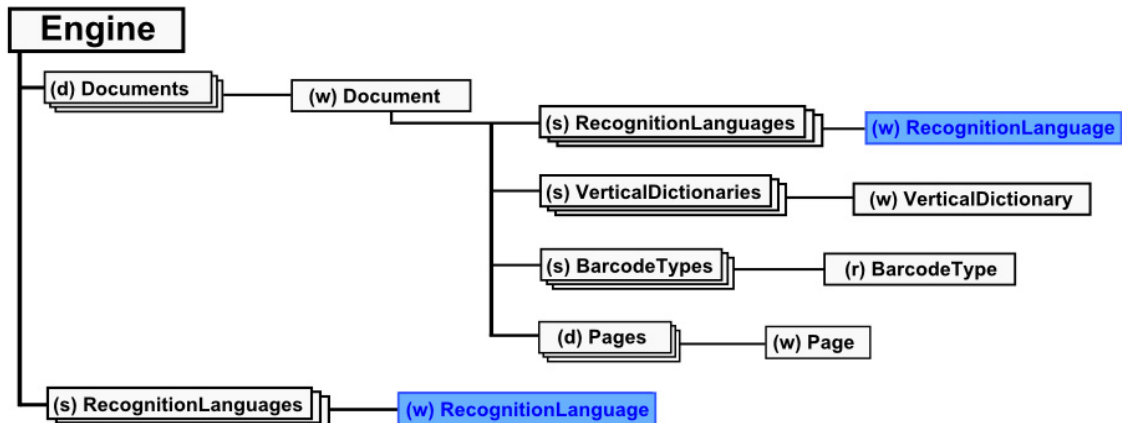
**5.4.15 - Page / Pages**

The **Page object** represents a document page. In addition to a set of processing operations, you can use the Page object to access zone, character, statistical and image objects. The UserZones collection maintains the zone set available for modification and it provides the input zone set for the recognition. The OCRZones collection is created by decomposition and recognition processes, the OCR recognized character results are assigned to the OCR-zones.

After processing methods such as `LocateZones` and `Recognize` have run, the UserZone collection is synchronized with the OCRZones collection. The OCRZone objects are read-only. Using the Image object, you can access a set of image versions maintained by the engine (such as the original, the b/w and the thumbnail images). By using the Characters collection you can access the whole recognized and optionally filtered character set. The Statistics object provides details about the recognition processes.

The **Pages collection** is a sorted set of Page objects, where each element represents a page in the document. Through its properties and methods, the Pages collection enables you to gather information about the document pages.

### 5.4.16 - RecognitionLanguage / RecognitionLanguages



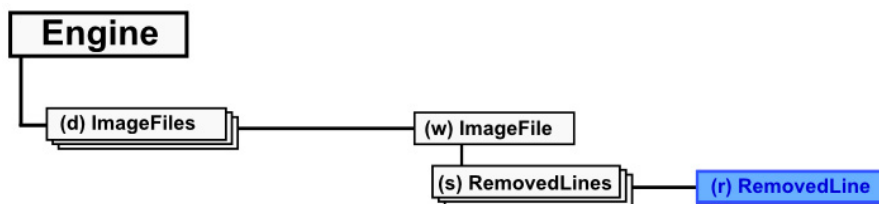
The **RecognitionLanguage** object contains information about the particular recognition language.

The **RecognitionLanguages** collection returns the collection of available recognition languages that can be switched on/off independently. The recognizable characters can be increased with additional characters by using the `LanguagePlus` property, and can be narrowed by using the `Filter` property. By default, only the English language is used for recognition (enabled).

The `Recognize` method of either the `Document` or the `Page` object uses this setting.

When the Asian OCR add-on is purchased, this Language collection includes Chinese (Traditional and Simplified), Japanese and Korean.

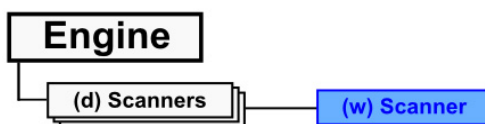
### 5.4.17 - RemovedLine / RemovedLines



The **RemovedLine** object represents a line, which was removed from the BW version of the original image. Through this object, the position and the orientation of the line can be accessed.

The **RemovedLines** collection is a set of `RemovedLine` objects representing the lines that were removed from the related BW image. The `RemoveLines` method of the `Page` object returns this collection.

### 5.4.18 - Scanner / Scanners



**Scanner objects** represent a physical scanner. The `Name` property uniquely identifies the scanner. Prior to invoking scanner methods, or accessing scanner properties, the particular scanner must be initialized calling the `Init` method. Depending on the scanner model, this may take several seconds. Each scanner manages its scanning properties individually. To load or save a scanning profile using external files, invoke the `LoadSettings` and `SaveSettings` methods. Using the `Scanner` object you can either scan images into image files – using the `ScanPages` method – or you can pass the `Scanner` object to the `Document` object's `ScanImage` method to scan pages directly into a particular document.

The **Scanners collection** lists the set of available `Scanner` objects. Each element represents a physical scanner. The scanner list is loaded from an `.ini` file. The `.ini` file can be altered using the Scanner Wizard. To run the wizard, invoke the `ScannerWizard` method. Using the wizard you can add, remove and parameterize the available scanners. By default, the `ini` file is called `scanner.ini` and stored in the current user's application data. The `scanner.ini` file can optionally be saved to the All Users application data folder, or into the application binary folder. To change the `ini` file location, update the `ConfigFileLocation` property accordingly. If the `ConfigFileLocation` property changes, the actual scanner list is destroyed, and the new `scanner.ini` file is loaded. Use the `SaveSettings` and `LoadSettings` methods to save / restore scanner parameters – like resolution or color mode – stored in the external setting file.

During scanning, the scanner drivers may show their own progress dialogs. To specify a parent window for these dialog boxes, set the `WindowHandle` property of the `Engine` object accordingly. Providing a valid window handle is highly recommended to avoid the situation when a window or message is displayed behind the main application window. The `WindowHandle` property should be set before initializing the scanning subsystem. If the parent window is subject to change after you initialized the scanning system, you need to re-init the scanning for the new setting to take effect. Update the value of the `ConfigFileLocation` property to achieve this.

---

#### Note:

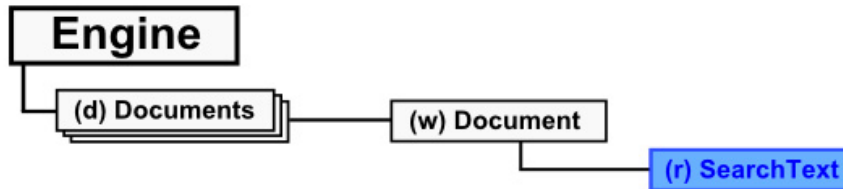
Changing the `ConfigFileLocation` property also resets all scanner properties to their default values.

---

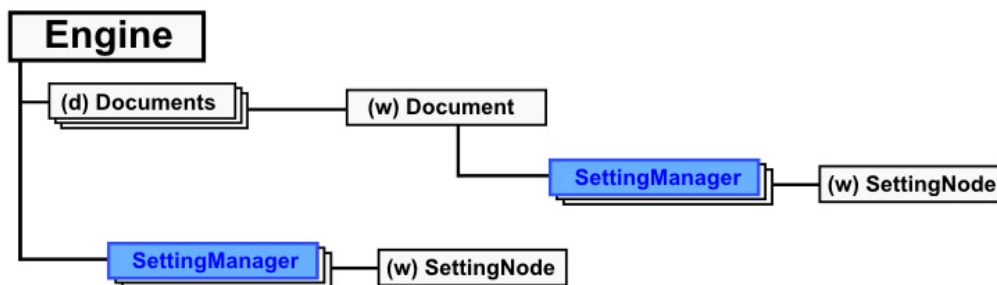
## 57

### 5.4.19 - SearchText

The role of the **SearchText** object is to allow you to carry out parameterized text searches (such as finding whole words only, searching with case matching, searching from the beginning or the end of the document etc.); navigate through your results and process them using the text marking and redaction facilities.



### 5.4.20 - SettingManager



The **SettingManager** object allows you to manage the settings of the Engine and the Documents. Settings are named values which control the behavior of different functions on the lower levels of the SDK. They are organized in a tree-structure. The setting tree contains a large number of settings, but you should use the SettingManager only to manipulate those that are documented in the RecAPI online help.

In a setting tree, every setting is attached to a node. (The nodes do not necessarily have associated setting values. For example, some nodes might be created to act as the parent of a set of child nodes having associated setting values.) Individual settings are unambiguously identified by their 'path' name, which is made up by concatenating the names of the nodes from the root node down to the setting node. The node names are separated by dots.

Several setting trees can co-exist at a time. Upon its creation, the Engine creates the first setting tree. All other setting trees are created as copies of the current settings of the Engine. Each newly created Document receives such a copy of the Engine's settings. This way the settings of the individual documents are independent of each other and also from the settings of the Engine.

When a document is saved to an OPD file, the Document settings active at that time are also saved automatically. However, you should explicitly save the Engine settings if needed.

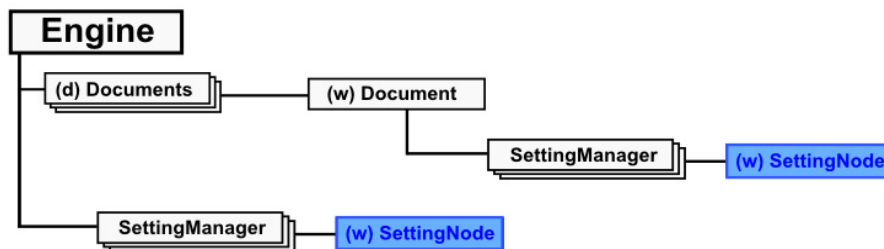
# 58

Scanner and Converter object settings present special cases. Although they do appear in setting trees, several of their individual setting values may depend on others. Therefore it is recommended that you work with Scanner and Converter object settings via these objects themselves, and not by utilizing the SettingManager.

Scanner and Converter objects can be queried only from the Engine. These objects modify the Engine settings. For document-specific Scanner and Converter setting modification these objects provide some helper methods. To let document-specific Scanner and Converter settings have an effect in your workflows, you must set `SP_EXP_USEENGINECONVERTERS` and `SP_EXP_USEENGINESSCANNERS` workflow step parameters to false.

The setting manager supports the following setting types: integer, floating point, Boolean, string, integer array, floating point array, Boolean array and enumerated. Saving and loading the entire setting tree or an arbitrary sub-tree is also supported.

## 5.4.21 - SettingNode



The SettingNode object provides access to a particular node and its setting if it exists. You can use it to get or set the value of a setting, retrieve information about a setting's flags and the default value, and to enumerate the child nodes.

## 5.4.22 - SpellLanguage / SpellLanguages

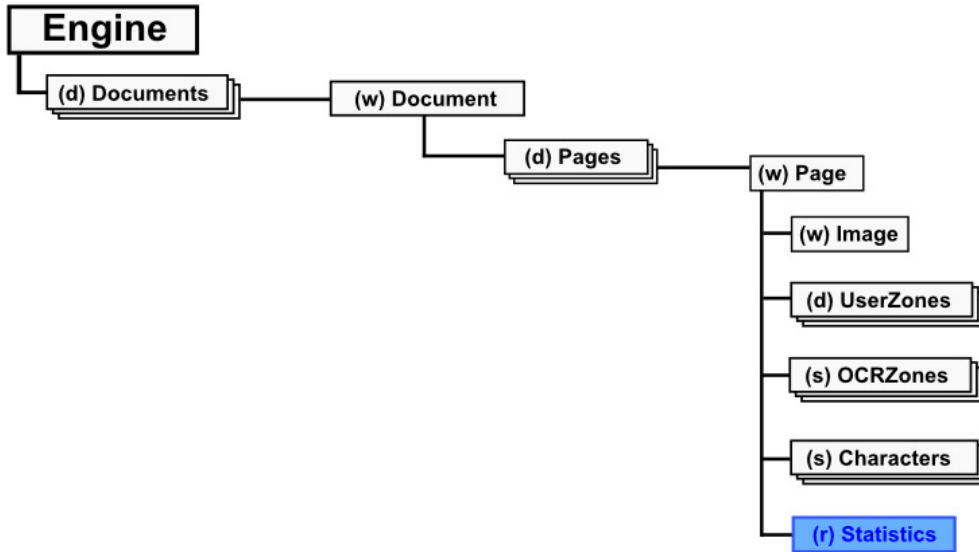


The **SpellLanguage** object contains information about a specific spelling language.

The **SpellLanguages** collection is a sorted set of SpellLanguage objects, where each item represents an available spelling language (i.e. one with a Language dictionary available in the current engine configuration). Through its properties, the SpellLanguages collection enables you to retrieve information about the installed spelling languages.

# 59

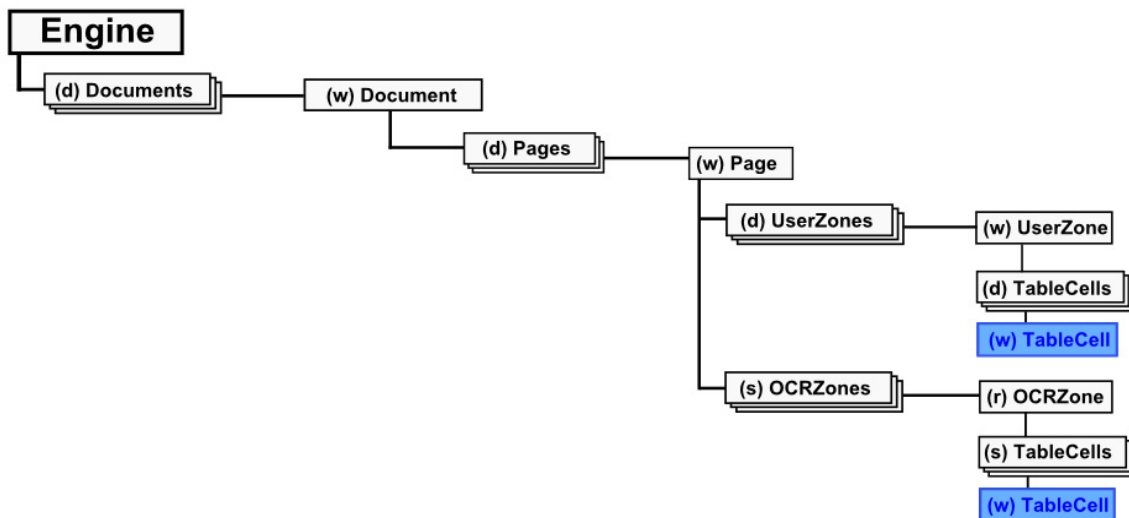
## 5.4.23 - Statistics



The **Statistics** object provides statistical information (general, accuracy and timing information) about the processing of the page.

If the page has not been recognized, all properties of the related Statistics object return zero (0). The Statistics object does not reflect the time spent in the `Progress` or `ProgressEx` events fired by Document object.

## 5.4.24 - TableCell / TableCells



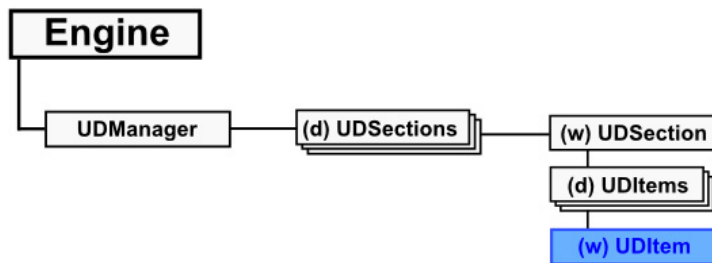
# 60

The **TableCell** object represents a single cell in a table.

The **TableCells** collection is a sorted set of TableCell objects.

If the parent of this collection is a UserZone object, the content of the collection can be modified. If the parent of this collection is an OCRZone object, the collection's items are available for enumeration in read-only mode, but they cannot be modified. The members of the collection are sorted by the top and left coordinates.

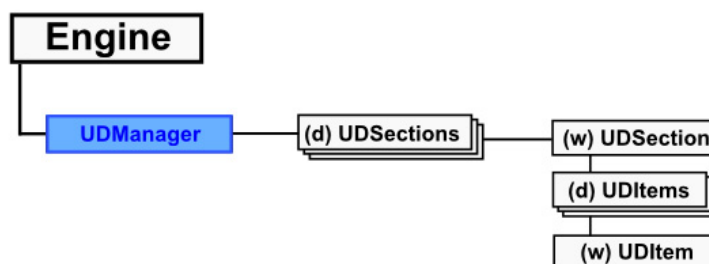
## 5.4.25 - UDItem / UDItems



The **UDItem** object contains information about a specific User Dictionary item associated with a particular UDSection.

The **UDItems** collection is a sorted list of the UDItem objects (string literals) that are assigned to a user dictionary section. Through its properties, the UDItems collection enables you to retrieve information about the underlying UDItem objects. The string literals contained by the collection are sorted alphabetically and followed by the regular expressions.

## 5.4.26 - UManager



The **UDManager** object is used for creating and maintaining user dictionaries. Note that only a single user dictionary can be opened for maintenance at a time. This user dictionary is referred to as active.

Each user dictionary contains one or more UDSections and each UDSection contains a number of UDItems. Any UDItems can be a literal string. Each valid dictionary must contain at least one UDSection and UDItem object.

# 61

User dictionaries are used with the Document object's `SetUserDictionary` method. However, be sure that the checking subsystem is enabled.

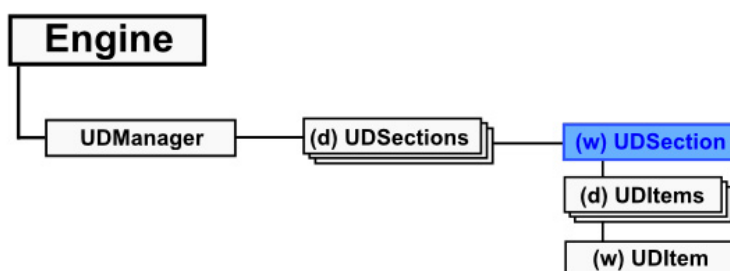
---

**Note:**

The checking subsystem can only use a single user dictionary, optionally with more than one section. A particular section can be specified when setting a user dictionary for a document or a zone.

---

## 5.4.27 - UDSection / UDSections



The **UDSection** object is deprecated in Nuance OmniPage Capture SDK 20.1.

## 5.4.28 - UILanguage / UILanguages

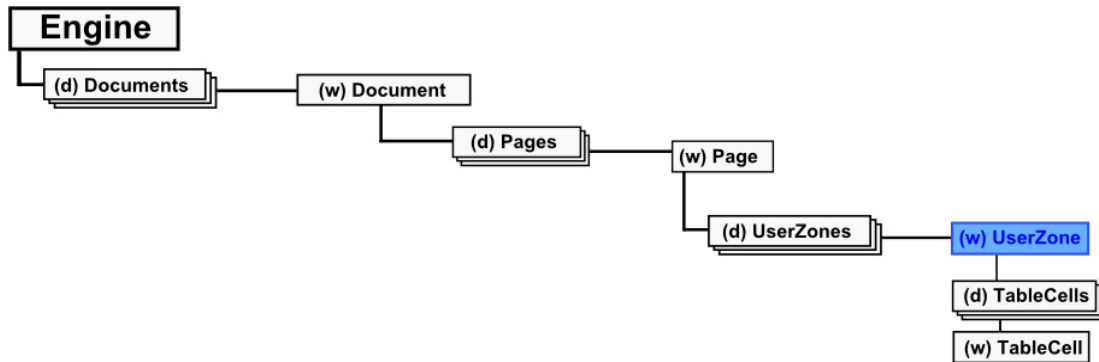


The **UILanguage** object contains information about a specific Visual Toolbox User Interface language.

The **UILanguages** collection is a sorted set of the available **UILanguage** objects. Each member item represents an available Visual Toolbox User Interface language (currently English, French and German). Through its properties, the **UILanguages** collection enables you to retrieve information about the available User Interface languages. This property primarily controls the Visual Toolbox UI language, although it also sets the IPRO related UI language for the ScannerWizard.

# 62

## 5.4.29 - UserZone / UserZones



The **UserZone object** contains information about a particular user zone on the page. The UserZone object provides access to the coordinates of a user zone, its content type, etc. All coordinates of this object refer to the original image.

The **UserZones collection** is a sorted set of UserZone objects, where each element represents a user zone on the related page. Through its properties, the UserZones collection enables you to access and manipulate the user zone set which provides input for recognition.

After zone location or recognition, the OCRZones collection is created and the UserZones collection is synchronized accordingly.

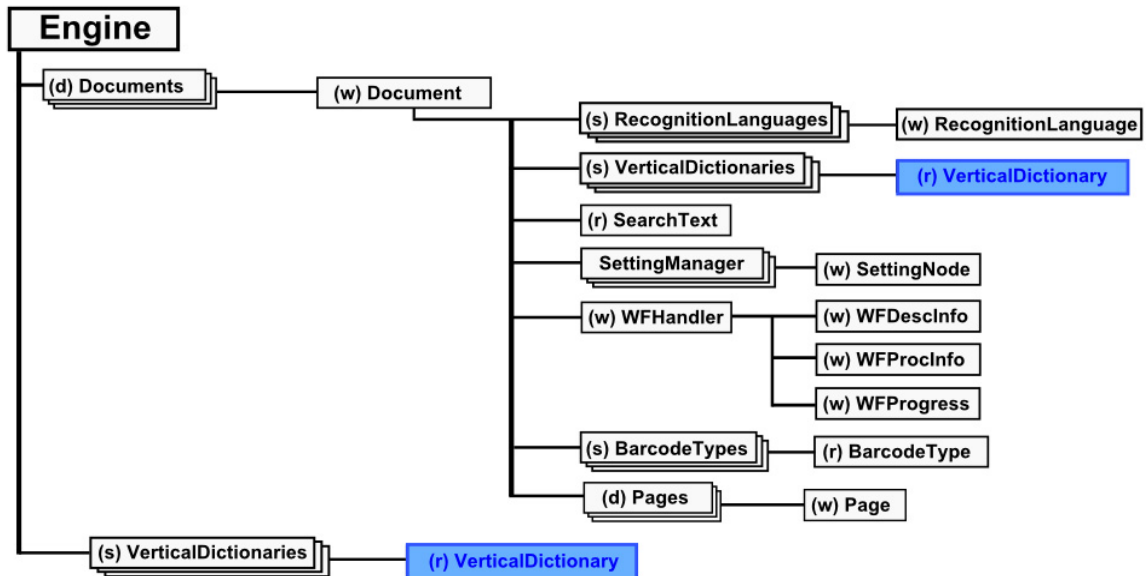
---

### Note:

In processing methods that alter the user zone set, the collection is stored for later reference automatically to provide a way to recall the original UserZones set using the `RestoreUserZones` method.

---

### 5.4.30 - VerticalDictionary / Vertical Dictionaries



The **VerticalDictionary object** contains the name and the description of the particular vertical dictionary. Each vertical dictionary can either be disabled or enabled if the parent of the containing VerticalDictionaries collection is a Document object.

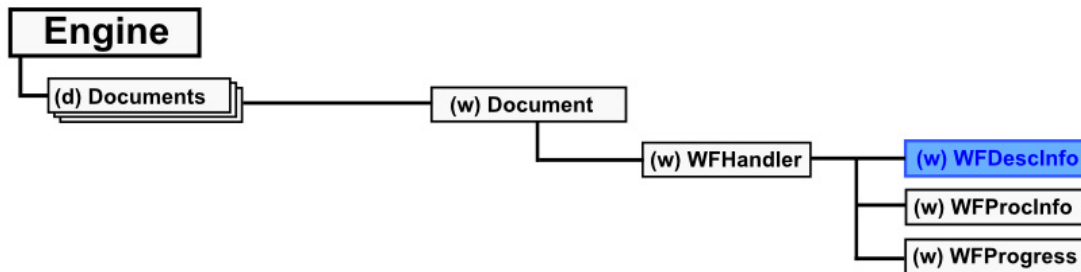
The Capture SDK supports the following vertical dictionary set:

- Dutch Legal Professional Dictionary
- Dutch Medical Professional Dictionary
- English Financial Professional Dictionary
- English Legal Professional Dictionary
- English Medical Professional Dictionary
- French Legal Professional Dictionary
- French Medical Professional Dictionary
- German Legal Professional Dictionary
- German Medical Professional Dictionary

The **VerticalDictionaries collection** is a sorted set of the available VerticalDictionary objects. This collection can be accessed through the Engine and Document objects. Through the Engine object, you can only query the available dictionary set and its properties. Through the Document object, you can enable or disable an arbitrary combination of the available vertical (also referred to as professional) dictionaries.

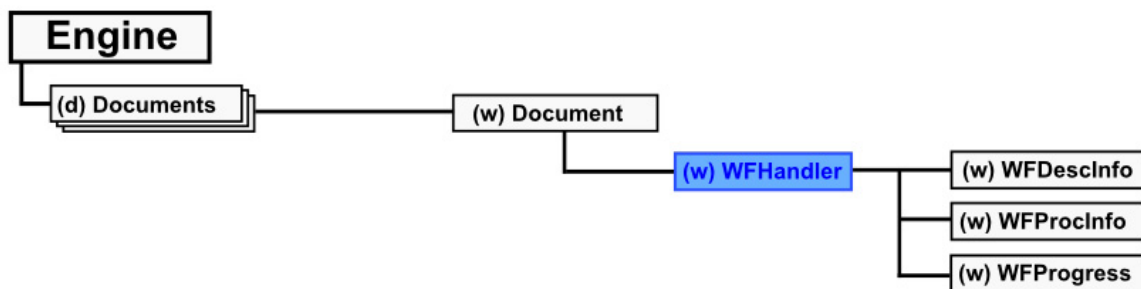
# 64

## 5.4.31 - WFDescInfo



The **WFDescInfo** object is provided by the WFHandler object. It is used to acquire various kinds of static information about the current workflow (information that remains unchanged even while the workflow is running).

## 5.4.32 - WFHandler



The **WFHandler** object contains a wide range of workflow handling related functionality. Its two main purposes are

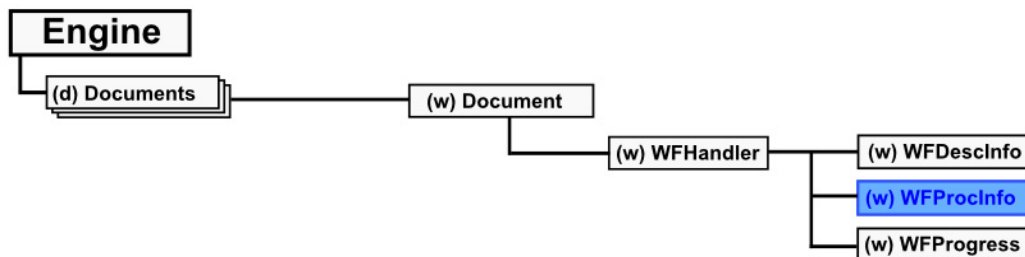
- running existing workflows created with the Workflow Assistant
- programmatically compiling and running workflows.

In either case, you can specify parameters for individual workflow steps. In addition to the traditional steps (loading, scanning, recognizing, exporting, etc.) the current application version offers additional ones, such as Form Data Extraction and Text Marking, including redaction.

To query dynamic workflow information (execution state) or static information (origin, friendly and full file name, etc.), you can also use WFProcInfo and WFDescInfo provided by this object.

# 65

## 5.4.33 - WFProcInfo



The **WFProcInfo** object is provided by the WFHandler object. It is used to acquire various kinds of dynamic (changing) information about the currently running workflow.

# 66

## 6 - Visual Components

### 6.1- Changes

With the introduction of the MediumWeight Visuals in SDK v18, you are likely to face the most comprehensive set of changes in case you used the Visual Toolbox of earlier SDK versions. This chapter gives you an overview of both legacy controls from v18 on and the new ones.

### 6.2- Introduction

This help system provides a description of the Visual Toolbox (also known as 'Visuals') as well as a recently developed set of ActiveX controls named MediumWeightVisuals. These provide all the visual controls necessary to create customized user interfaces for Windows applications benefiting from the services of the IPRO interfaces. While IPRO provides invisible imaging, recognition and document management services, these ActiveX controls focus on typical user interface elements requiring complex programming tasks to display images, correct preprocessing and decomposition mistakes or view and modify recognition results. The controls visualize the contents of the objects managed by IPRO (image, recognized text, and so forth). They provide different levels of presentation, adjusted to the underlying object hierarchy. In addition to their basic tasks, the controls work in harmony with IPRO and with each other on multiple multi-page documents.

The interfaces of the Visual Controls are available in English, French and German. See the IPRO topic on ["UILanguage / UILanguages" on page 61](#).

### 6.3- Overview

As stated above, these ActiveX controls can be divided into the following main groups:

- Visuals
- MediumWeightVisuals

Legacy visual controls work only with an in-process IPRO server, whereas MediumWeight Visuals enable the integration of and out-process IPRO server as well.

To facilitate the proper interworking, a control (either legacy or MediumWeight) needs to be attached to the relevant IPRO object (in practice, either a document or an engine), and it is ready to go. There are a few issues to consider when attaching a control to an engine or a document:

- Controls providing document view (for example, **Thumbnail** and **Details**) can be attached only to a Document object.
- Controls attachable to both engines and documents (for example, **Setting view**, **Recognition language view**, or **Dictionary view**) provide information depending on what they are attached to. Controls attached to an engine show all engine-specific information, while those attached to documents show document-specific information. The information displayed can differ between the engine-attached and document-attached version of the same control.

# 67

The system-level information of IPRO is the session 0, stored as a template. Each document is provided a session from the current document session and there is no inter-visibility among document sessions. The sessions requested from the various engines can show values that differ from the document session values.

The default interfaces of the framework are simple, and are mostly not used. For both types of controls, the main type of interface is the event interface. This interface has two types, both providing the same tasks: the disp interface and the custom interface. The main differences between these are:

- **Disp interface:** usually denotes an ActiveX control. Its advantage is security; it is always called on a UI thread, the client-side does not have to be thread-safe. Its disadvantage is the very slow execution.
- **Custom interface:** this type provides very quick action, but as the call does not necessarily arrive on a UI thread, there is a slight possibility of the event becoming lost.

The disp interface is considered to be the default interface offered for objects.

All controls of the program implement a number of pre-defined interfaces, thus providing a unified user interface for the various controls.

The actual visual appearance of the visual controls takes its defaults from the operating system.

## 6.3.1 - Visuals

This section lists the Visuals controls (dropped ones in *italics*), the still-available ones complete with their default interfaces and CLSIDs.

- **Character Set Control (CSC):** *dropped from v18.0. Its functionality is covered by the MediumWeight Visual Control Character Bar.*
- **Converter Manager Control (CMC):** *dropped from v18.0. Its functionality is covered by the MediumWeight Visual Control ConverterView.*
- **Document ManagerControl (DMC):** *dropped from v18.0. dropped. Its functionality is covered by the MediumWeight Visual Control DetailView.*
- **Font Matching Control (FMC):** supported in v18 and point releases as well with a change of CLSID.
  - default interface: IFmc
  - CLSID: {CD9FDAAF-5C7C-40CC-80F2-1487FF7D9EBD}
- **Image View Control (IVC):** supported in v18 and point releases as well with a change of CLSID.
  - default interface: IIvc
  - CLSID: {1E2420A3-7BA3-4E84-84CB-03AFBE689F96}
- **Scanner Parameter Control (SPC):** supported in v18 and point releases as well with a change of CLSID.
  - default interface: ISpc
  - CLSID: {462EF4BE-787E-4ED1-BEE2-EF1CD37F4863}
- **Text Editor Control (TEC):** supported in v18 and point releases as well with a change of CLSID.

# 68

- default interface: ITec
- CLSID: {219A1D06-3B8F-4EE1-BF15-C257A2CE623C}
- **Thumbnail View Control (TVC):** dropped from v18.0. Its functionality is covered by the MediumWeight Visual Control ThumbnailView.
- **WorkflowControl (WFC):** supported in v18 and point releases as well with a change of CLSID.
  - default interface: IWfc
  - CLSID: {6C3DBC07-594F-4161-A9D4-D5F1400E42F7}
- **Workflow Viewer Control (WFVC):** supported in v18 and point releases as well with a change of CLSID.
  - default interface: IWfvc
  - CLSID: {4FBD86A3-A56F-4302-88EC-3BC37BBCECA8}

## 6.3.2 - MediumWeightVisuals controls

- **CharacterBar (former CSC):**
  - default interface: ICharacterBar
  - CLSID: {13E35487-6ED8-478c-9396-27B7EFDB28DE}
- **ConverterView (former CMC):**
  - default interface: IConverterView
  - CLSID: {6BE0B22B-70F9-408F-97D5-5985B07EB855}
- **ConvertersView:**
  - default interface: IConvertersView
  - CLSID: {4679CD03-CF13-4C5B-A3BF-B0AC134FF90D}
- **DetailView (former DMC):**
  - default interface: IDetailView
  - CLSID: {A99D49C8-AB2C-4BBD-828B-F5E7341EFE21}
- **DictionaryView:**
  - default interface: IDictionaryView
  - CLSID: {BCE24B12-5F99-4e53-ACC5-C45F0C90FED7}
- **DocumentView:**
  - default interface: IDocumentView
  - CLSID: {A45578C1-B538-4461-BB29-79B6386DA1E2}
- **LanguageView:**
  - default interface: ILanguageView

# 69

- CLSID: {BB012969-EB2D-4777-922E-27F1F5D93DC8}
- **SettingView:**
  - default interface: ISettingView
  - CLSID: {AD3AFC3E-6BFE-41D4-A8A2-45E6AEA44500}
- **ThumbnailView (former TVC):**
  - default interface: IThumbnailView
  - CLSID: {9A9041B0-3938-467D-9A12-66FE06BD8D78}
- **WFInfoView:**
  - default interface: IWFInfoView
  - CLSID: {65C23F0A-2346-42B4-8CBA-68C650EFDDCF}
- **WorkflowBar:**
  - default interface: IWorkflowBar
  - CLSID: {26872619-DBE2-4abc-B982-5D072AED9677}

# 70

## 7 - Index

### 3

3D deskew 34

### A

Abbreviations 1

Accessing Help 5

Accuracy 59

Accuracy vs speed data 10

Activation steps 3

ActiveX Image Viewer Control 34

Alternatives for recognized characters 38

Anchors 36

Appending TXT outputs 40

Auto-zoning 34

### B

BAR 37

BAR\_AMP 37

Barcode Types 46

Barcodes 40

Batch Manager 40

Before installing 4

Binary output 40

### C

Call-back functions 38

Cells in tables 37

Changes 66

Character Set Module 38

Character tips 38

Characters 46

Check boxes 35

# 71

- Checking module availability 9
- Circle texts 35
- Code Page Handling Module 38
- Comb fields 35
- Concurrent scanning processes 33
- Confidence reporting 38-39
- Converter objects 58
- Converters 47
- CSV output 39
- Cutting book pages 34

## D

- Deadlock situations 43
- Default speed-accuracy trade-off 10
- Deskewing 34
- Detection of lines and frames 34
- Development Kit choices 1
- DirectTXT Output Converter Module 30, 39
- Distributing Applications 8
- Distribution file set 9
- Distribution log file 9
- DOC 40
- DocuDirect 40
- Document-level processing 31, 40
- Documentation 5
- Documents 49
- DOT 37
- Dutch Vertical Dictionaries 38
- Dynamic Linking of RecAPI 41

## E

- Electronic documentation 6
- Empty forms 35
- Engine 50
- Engine combinations 10

# 72

Engine settings 57  
English Vertical Dictionaries 38  
Error Handling Module 32  
EULA i  
Evaluation Kits 3

## F

Fast parsing 31  
Fast recognition output 39  
Faster image file handling 33  
Fastest recognition 10  
Fillable fields 36  
Filled form recognition 36  
Filling method 37  
Financial Dictionary 63  
Fixed Volume licensing 3  
Flavors of PDF 38  
Flowing Page 41  
Flowing text zones 35  
Form converters 47  
Form Data Extraction (FDE) 64  
Form Recognition Module 35  
Form templates 35-36  
Form zones 35  
Formatted page output 31, 40  
Formatted Text 40  
Frame and border detection 34  
Framework .NET 5  
French Vertical Dictionaries 39  
FRX 37  
FTP 52

## G

General Operations Module 31  
German Vertical Dictionaries 39

# 73

Getting Help 5  
Graphics zones 35

## H

Help 5  
HIMGFILE handle 33  
Hint file for scanning 33  
HNR 37  
HyperThreading 31

## I

Identifiers for Settings Collections 32  
Ignore zones 36  
Image-on-text PDF 39  
Image-only PDF 38  
Image Converters 47  
Image despeckling 34  
Image File Handling Module 32  
Image Handling Module 33  
Image pre-processing 34  
Image resolution 51  
ImageFiles 51  
Insert recognized pages 40  
Installation 5  
Inversion 34  
IPRO 43  
IPRO Object Model 44  
ITest 40

## K

KernelAPI 30  
KernelAPI Modules 31  
KernelAPIS 30  
Kit variants 1  
kRecScanInit 33

# 74

## L

- Language dictionaries 38
- Language Module 38
- Languages 61
- Layout retention 40
- Legacy parsing 31
- Legal Dictionary 63
- License agreement 5
- License Handling Tool 3
- Licensing the product 3
- Line detection 34
- Linking of RecAPI 41
- Links 52
- Logical Form Recognition 35

## M

- Manual zoning 34
- Manual zoning for forms 36
- MAT 37
- Medical Dictionary 63
- Memory space 34
- MemoryBitmaps 53
- Microsoft .NET 5
- Modifying images 34
- Modules - checking availability 9
- Modules in KernelAPI 31
- MOR 37
- Most accurate recognition 10
- MTX 37
- Multi-core processors 31
- Multi-language recognition 37
- Multi-page document handling 40
- Multi-page image file formats 33
- Multiple scanning processes 33

# 75

## N

Node 32, 57  
Non-setting nodes 32  
Normal PDF 38

## O

Object Model 44  
OCR zones 34  
OCRZones 53  
ODMA 52  
OEM Licenses 3-4  
OmniForm 35  
OMR 37  
One-step functions 40  
Online help 3, 6  
OPD files 40, 57  
Optimal recognition resolution 34  
Overview 1

## P

Page-level processing 31  
Page operations 40  
Pages 54  
Parallel scanning processes 33  
Parsing choices 31  
Path 32  
PDF 40  
PDF output 39  
PDF recognition 37  
Performance comparison 10  
Pixel order 51  
Pizzabox shape 34  
Plain Text 30, 40  
PLUS2W 37

# 76

PLUS3W 37  
Predetermined paths 41  
Preparing a distribution file set 8  
Preparing for installation 4  
Preprocessing of images 34  
Processing filled forms 37  
Professional dictionaries 38

## Q

Queuing scanner calls 33

## R

Readme files 6  
RecAPI 30  
RecAPI .NET support 31  
RecAPI dynamically linked 41  
RecAPI statically linked 41  
RecAPIPlus 31, 40  
RecAPIPlusS 31  
RECERR 32  
Recognition Data Handling Module 38  
Recognition Modules 9, 37, 52  
Recognition of filled forms 36  
RecognitionLanguages 55  
Recognized character alternatives 38  
Release Notes 4  
RemovedLines 55  
RER 37  
Resolution 34  
Resolution enhancement 34  
Root node 57  
Rotation 34  
RTF 40  
Running workflows 64

# 77

## S

- Scanner Setup Wizard 33
- Scanners 56
- Scanning 9
- Scanning Module 33
- SearchText 57
- SET tools 34
- Setting node 57
- Setting tree 57
- Setting types 58
- SettingManager 57
- Settings Collections 32
- Settings Manager Module 32
- Settings values 32
- SharePoint 52
- Single-page images 33
- Software License Agreement i, 9
- Speed vs accuracy data 10
- Spell Checking Module 38
- Spelling languages 58
- Spreadsheet 41
- Standard parsing 31
- Static libraries 41
- Statistics 59
- Steps for installation 5
- Suggestions for recognized words 38
- Superior parsing 31
- Suspicious recognition results 39
- System Requirements 12

## T

- Table detection 34
- Table Recognition Module 37
- Table zones 35

# 78

- TableCells 59
- Templates for zoning 34
- Temporary Fixed Volume Licenses 9
- Testing Applications 9
- Text Converters 47
- Text Marking 64
- Text output 39
- Thread safe scanning 33
- Time-limited licenses 3
- Timing information 59
- Trade-off settings 10
- True Page 41
- Types of settings 32

## U

- User-written callback functions 38
- User Interface language 61
- User zones 34, 62

## V

- Variants of kit 1
- Vertical Dictionaries 38, 63
- Voting engines 37

## W

- WFDescInfo 64
- WFHandler 64
- Wizard for Scanning 33
- Word accuracy values 10
- Word suggestions 38
- WordML 40
- WordPerfect 40
- Workflows 40, 64

# 79

## **X**

XLS 40

XML output 39

XML output for form processing 36

## **Z**

Zone file 34

Zone Handling Module 34

Zone types 35

Zones 34, 53, 62

Zoning for forms handling 36

[www.nuance.com](http://www.nuance.com)

Copyright © 2017 Nuance Communications, Inc. All rights reserved. Subject to change without prior notice. Nuance, the Nuance logo, and OmniPage are trademarks or registered trademarks of Nuance Communications Inc. in the United States and/or other countries. Microsoft and Windows are registered trademarks and/or trademarks of Microsoft Corporation in the United States and other countries. All other company names or product names may be trademarks of their respective owners. This program is protected by US and International copyright laws.

# 80

## 8 - Appendix - Sample Barcode Images

### 8.1- Existing 2D types newly supported by BAR

#### 8.1.1 - BAR\_QR: QR Code (Quick Response Code)



#### 8.1.2 - BAR\_DMATRIX



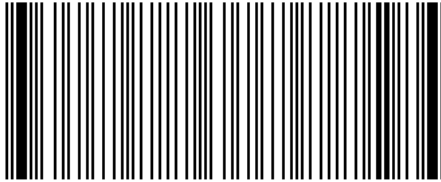
#### 8.1.3 - BAR\_A2of5

#### 8.1.4 - BAR\_2of5



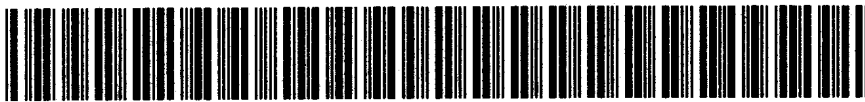
# 81

## 8.1.5 - BAR\_C93

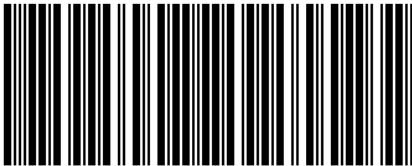


74481587448158

## 8.1.6 - BAR\_C39\_NSS



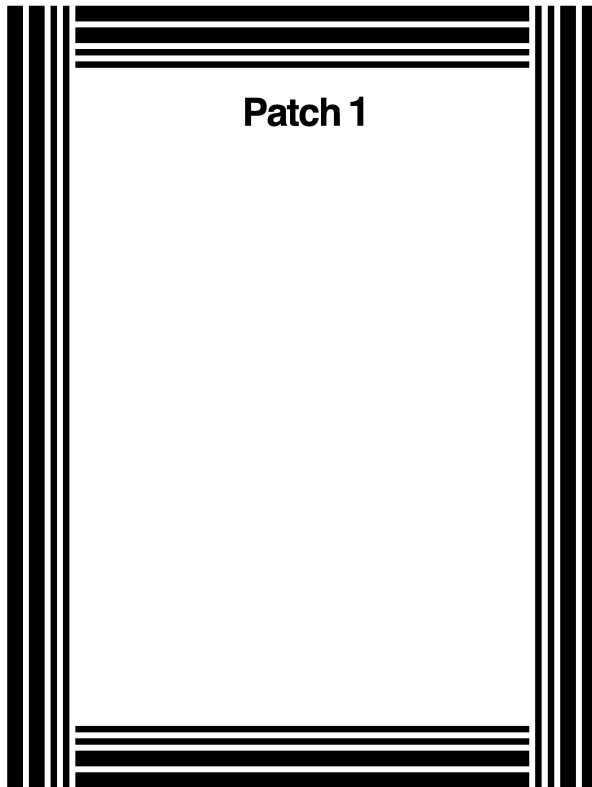
## 8.1.7 - BAR\_MAT25



40106654010665

# 82

## 8.1.8 - BAR\_PATCH



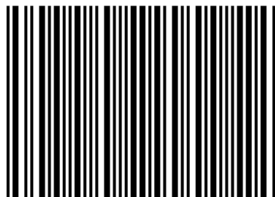
## 8.1.9 - BAR\_PLANET



# 83

## 8.2- New 1D barcode types supported by BAR only

### 8.2.1 - BAR\_CODE11



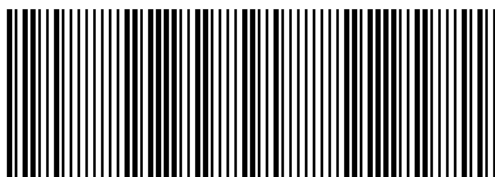
699601669

### 8.2.2 - BAR\_ITALIAN25



12652325871-0

### 8.2.3 - BAR\_MSI



64037986403798

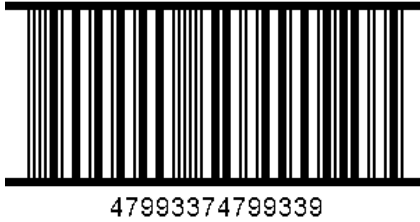
### 8.2.4 - BAR\_BOOKLAND



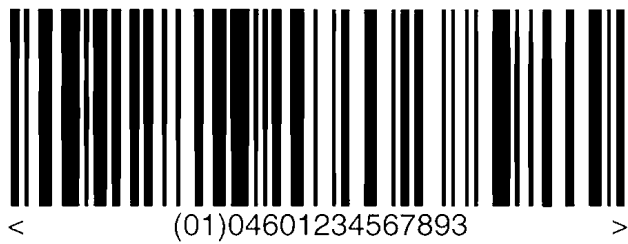
9 787442 521744

# 84

## 8.2.5 - BAR\_ITF14



## 8.2.6 - BAR\_EAN14



## 8.2.7 - BAR\_SSCC18



## 8.2.8 - BAR\_DATABAR\_LIMITED



# 85

## 8.2.9 - BAR\_DATABAR\_EXPANDED



(01)9001234567890(3103)012233(15)991231

## 8.2.10 - BAR\_4STATE\_ONECODE



44123123456123456789

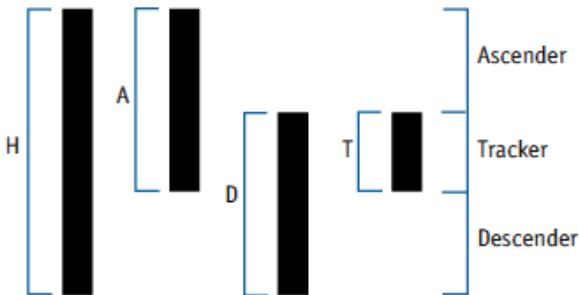
## 8.2.11 - BAR\_4STATE\_AUSTRALIA



8765432101234560000

# 86

## 8.2.12 - BAR\_4STATE States and Values



*The four states of the barcode*

Bar	Elements	Bar State Value
T	Tracker	3
D	Tracker, Descender	2
A	Tracker, Ascender	1
H	Tracker, Ascender, Descender	0

## 8.3- Types supported by BAR\_AMP only

### 8.3.1 - BAR\_4STATE\_DK1

# 87

## 8.3.2 - BAR\_AZTEC



## 8.3.3 - BAR\_C32

