opentext™

OpenText™ Intelligent Capture

Dispatcher Manager Guide

This guide describes Dispatcher Manager, which is an intelligent data recognition application enabling data capture from paper documents and the transformation of these documents into electronic images.

ECPCORE220300-CRL-EN-01

OpenText™ Intelligent Capture
Dispatcher Manager Guide
ECPCORE220300-CRL-EN-01
Rev.: 2022-June-13

This documentation has been created for OpenText™ Intelligent Capture CE 22.3.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Overview

Recognition Designer is used in both Intelligent Capture and Core Capture.

Recognition Designer is a term used throughout this guide to see Recognition Designer and Dispatcher Manager. However, the behavior is different between Recognition Designer and Dispatcher Manager. To learn about the differences, see "Comparing Recognition Designer and Dispatcher Manager" on page 15. Note also that, if you are using document types, then Dispatcher Manager cannot be used. In this scenario, you cannot create an indexing family. Instead, the Document Designer generates an indexing family within the specified recognition project in Recognition Designer when you save the document type. The indexing family contains all the index fields for the document type.

Important
Unless otherwise noted, the contents of this guide apply to both Intelligent Capture and Core Capture.

1.1 What is Recognition Designer?

Recognition Designer is an intelligent data recognition application enabling data capture from paper documents and transforming these documents into electronic images and business accessed by both individuals and processes. The aim of intelligent data capture is to identify a document and match it to a business form in the inventory, so that the appropriate set of data is extracted. For example, loan forms and cash flow statements each have a different data set to be extracted.

Document identification (also known as *classification*) enables identifying if a document is a loan form or a cash flow statement so that the appropriate data set is extracted.

Paper documents typically captured with Recognition Designer come from back-office operations (billing, maintenance, planning, marketing, advertising, finance, manufacturing, and others) and business relations with other companies (partners and suppliers/vendors).

Limitations of Core Capture

Compared to Intelligent Capture, Core Capture has some limitations in functionality.

Scripting
Core Capture does not support scripting.

Production Auto-Learning (PAL)
Core Capture does not support Production Auto-Learning (PAL). To use PAL, in Intelligent Capture, click **File** > **Project Options** > **Production Auto-Learning**.

Pre-indexed files
Core Capture does not support pre-indexed files. To use pre-indexed files, in Intelligent Capture, click **File** > **Project Options** > **Classification** > **Pre-Index**.

Manually sending projects to production
Core Capture does not support manually sending a project to production. In Intelligent Capture, click **File** > **Send to Production**. Instead, Core Capture Designer automatically prepares the recognition project for production.

OCR engines
Core Capture supports the following OCR engines:

- Advanced OCR/ICR
- Western OCR

1.2 Why Use Recognition Designer?

A capture process may include steps that trigger so called "advanced recognition"modules of Intelligent Capture. In production, each of these modules read their settings from a recognition project that is linked to a given step during setup. These settings configure the logic of the module and provide the resources for data processing.

Recognition Designer provides the environment for designing, testing, and fine-tuning recognition projects. For additional customization, users can create and edit project scripts and import templates from other projects. It includes a collection of embedded tools and wizards that simplify the task of a designer. In addition, Recognition Designer allows you to configure and use the learning techniques.

1.3 Licensing

Recognition Designer allows users to work in either *Basic* mode (using the Intelligent Capture Server license) or *Advanced* mode (using the Advanced Recognition license) depending on the selected licensing type.

Recognition Designer requires licensing to operate in fully-functional mode enabling all its features:

- *Advanced Recognition license* or *DPMANAGR* license enables using and configuring of all Recognition Designer features, such as all types of templates, all types of *OCR* engines, table fields, PAL configurations, and others.
- *Intelligent Capture Server license* enables using and configuring of certain features such as generic templates, several OCR engines, index family editing, and others.

Note: When you upgrade from an Intelligent Capture Server license to an Advanced Recognition license, restart Intelligent Capture Designer. When launched, this tool receives up-to-date license information from the server.

1.4 Comparing Recognition Designer and Dispatcher Manager

"Dispatcher Manager and Recognition Designer differences" on page 15 outlines the key differences, by feature, between Recognition Designer and Dispatcher Manager.

Table 1-1: Dispatcher Manager and Recognition Designer differences

Feature	Dispatcher Manager	Recognition Designer
Application launch	Launches from **Start > Programs > OpenText Intelligent Capture > Tools (Recognition) > Dispatcher Manager**	Launches from **Start > Programs > OpenText Intelligent Capture > Intelligent Capture Designer > Recognition screen > Open project**
Application usage	Can be used for creating and configuring a recognition project for further processing with Validation and Recognition, which are deprecated modules. A DPP file created in Dispatcher Manager can also be used with the Extraction module.	Allows you to create and edit recognition projects (*DPP* files) to be subsequently used with the Extraction module. A DPP file created in Recognition Designer can also be used with Validation and Recognition, which are deprecated modules.
Access to certain features such as optical character recognition, all templates, *PAL* configurations, table fields, and others	Access to these features is available only when an Advanced Recognition license is available enabling Dispatcher Manager or other recognition tools. Available using the Dispatcher Manager DPMANAGR license.	Access to these features is available only with an Advanced Recognition license. However, limited functionality is available with an Intelligent Capture Server license.
Opening DPPs	Those created in Dispatcher Manager	Those created in Recognition Designer or imported from Dispatcher Manager.

Feature	Dispatcher Manager	Recognition Designer
Import DPPs	Not applicable	DPPs created in versions 6.0.x, 6.5, and 6.5 SP1 can be imported into Recognition Designer. When a DPP is imported, the import process generates the document type, which in turn generates an indexing family. Also, the defined zones (or free form templates) are available in the imported DPP file. In Recognition Designer, you can also import a project created using Dispatcher Manager version 7.0. **Note:** Import of DPP projects that were created using versions 5.x and earlier is not supported.

1.5 What is Advanced Recognition?

Advanced recognition (sometimes referred to as *AR*) is the functionality of Intelligent Capture that is implemented though several Intelligent Capture modules and software components. Advanced recognition includes the following features:

Image classification
Performs the graphical layout analysis or full text analysis of the image to match that image to a particular classification template. This logic is implemented by the Classification module. Manual image classification allows the operator to match the image to a particular template manually. This logic is implemented by the Identification module.

Data extraction
Finds and extracts information from documents using various extraction settings and rules and recognition engines. This logic is implemented by the Extraction module.

Production Auto-Learning (PAL)
An automatic learning process that creates new templates for each new graphic layout entering the production flow. PAL collects the processed images and validation information during production and uses it to fine-tune the existing templates. Fine-tuning consists in creating more precise graphic templates and textual templates that become available in production automatically. This logic is implemented by the Collector module and the Production Auto-Learning Supervisor utility.

1.5.1 Advanced Recognition Modules

The following modules implement the advanced recognition functionality of Intelligent Capture and get their setup settings from a linked recognition project:

Classification
Identifies documents so they are routed to the appropriate data extraction processes.

Identification
Runs by operators to check or finalize document identification after Classification. Operators are presented only with documents that were not successfully identified by Classification.

Collector
Enables operation of the Production Auto-Learning (PAL). PAL automatically creates new graphic templates using the images collected by Collector. The newly created templates progressively replace free form templates.

Extraction
Unattended module that uses the recognition project, along with other setup options, to determine how to recognize pages when processing tasks in production mode.

For more information on modules, see *OpenText Intelligent Capture - Module Reference (ECPCORE-CMD)*.

1.5.2 Classification

Classification is the process of identifying scanned business documents to reference business forms. For example, a bank has an inventory of business forms such as cash flow statements, loan forms, check request forms, and acquisition agreements. Every day the bank does the following:

- Produces and processes new examples of these forms and sends them to customers or partners.
- Receives mail and forms from customers and partners.

The bank mail flow is composed of examples of the bank's business forms plus other documents that do not correspond to any of the forms in the bank business forms inventory.

Classification is performed by the Classification module of Intelligent Capture when it is triggered in production with a task. The task specifies the classification project with a set of templates and a set of images to classify. Typically, all images in the task belong to one logical document.

The recognition project defines a collection of templates of different types, including standard templates, HPA templates, generic templates, and text matching templates.

In turn, the Classification module implements several classification engines. Each engine looks for templates of a certain type and executes a particular classification algorithm.

"Classification Engines and Classification Templates" on page 18 summarizes the information about the classification engines available, the templates used by every engine, and algorithms executed for each type of classification.

Table 1-2: Classification Engines and Classification Templates

Classification engine	Algorithm	Templates	Description
HPA classification	High Precision Anchors (Graphic Layout Analysis)	HPA templates	Selects a feature of an image based on a similar feature that was demarcated on a model image stored in a template.
Standard classification	Full Page Image Analysis (Graphic Layout Analysis)	Standard templates	Evaluates and compares an entire image to the models stored in each template.
Handwritten classification	Handwritten Detection Analysis (Graphic Layout Analysis)	Generic templates specially added for handwritten analysis	Evaluates images to determine the percentage of handwriting they contain. If higher than a predefined threshold, an image is classified as "handwritten".
Keyword classification	Full text Analysis	Generic templates associated with keyword rules	Performs OCR and evaluates the resulting text for keywords, pattern matches, or regular expressions that were defined in a template.
Text matching classification	Full text Analysis	Text Matching templates	

The classification engines are triggered for an image in a predefined order, starting with the fastest HPA classification. If the project does not include templates of a certain kind, the corresponding engine is not triggered. Figure 1-1 explains how the classification logic executes:

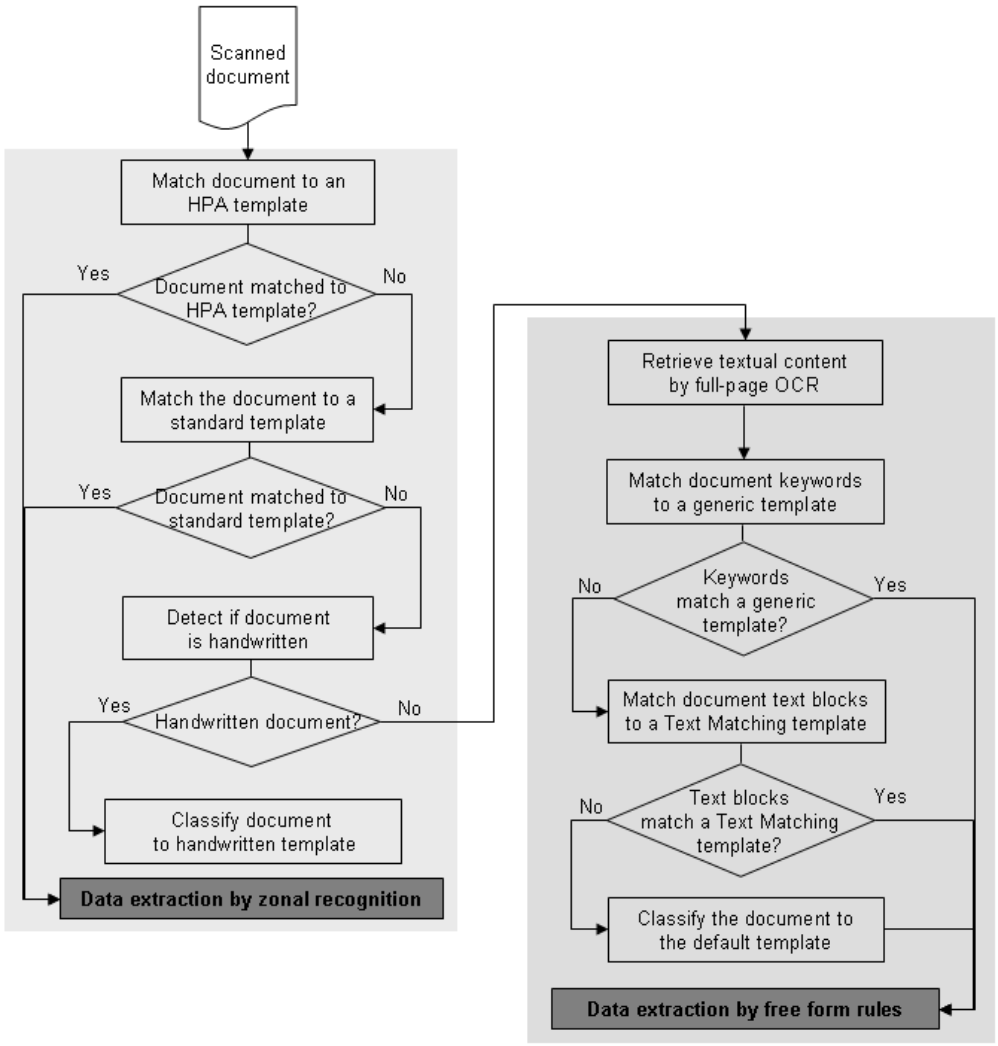


Figure 1-1: Image classification flowchart

If the image is classified to a particular template, the remaining engines are not triggered and the next image in the task is taken for classification. If the image cannot be classified by any classification engine, it is assigned to the default template as specified in the project.

1.5.3 Manual Classification

Manual classification is performed by the operator of the Identification module of Intelligent Capture. This module is only triggered with documents that were not successfully identified by the Classification module. The typical configuration of the process includes the Classification step followed by Identification.

The Identification module is triggered by a task that references the recognition project with the module settings and the list of templates. Also, the task passes in the images of the document(s) to be manually classified. For each passed in image, the operator views the list of available templates and picks up the matching template manually.

1.5.4 Data Extraction

Once a document has been classified, it typically has important data that needs to be extracted. Advanced recognition leverages the following data extraction techniques:

- Optical character recognition (OCR) for typewritten text
- Intelligent character recognition (ICR) for handwritten text
- Optical mark recognition (OMR) for data in marked fields such as checkboxes and bubbles on printed forms
- Barcode recognition for both 1D and 2D barcodes

Data extraction is the process of finding and extracting information from documents. Data extraction is performed by the Extraction module of Intelligent Capture when it is triggered in production with a task. The task references the recognition project with extraction setup settings. The task also passes in a set of images to extract data from, the templates to which those images were classified, and the index families which fields were places on those templates.

Extraction can perform both zonal and free form recognition.

- **Zonal recognition**: Applies to specific areas on a document where the same data field always resides. For example, specific tax forms or medical forms are structured documents and always contains the same specific information in the same location. Zones define data retrieval from exact locations on the documents.
- **Free form recognition**: Is used for documents where data retrieval is from different locations on different templates. For example, invoices from different companies are often structured differently and would be assigned to different templates. To extract data from these invoices, recognition must be flexible and setup of index families alone is not always sufficient. Free form recognition uses *full page recognition*, and location of data is determined in relation to keywords and associated words on the form.

A particular recognition algorithm is selected for an image based on additional template settings. Zonal recognition is triggered for images that are classified to a

template with zones and/or anchors. Free form recognition is triggered if the image is classified to a template associated with free form rules.

Documents in which data cannot be fully extracted are routed to the Completion module. In Completion, operators check and complete data extraction.

1.5.5 Using an OCR Data Cache for Classification and Extraction

Using an OCR data cache can result in fewer errors and better performance in comparison to running full OCR for classification and extraction. The OCR data cache contains text that was extracted from image files, original PDF or PDF/A files, or from PDF or PDF/A files that were converted from Microsoft Office documents. The OCR data cache is used in Classification, Extraction, Completion, and Identification.

In addition, the OCR data cache (and any associated image or PDF files) can be used for testing and debugging in your Recognition project. You can generate an OCR data cache (`<image_or_pdf_file_name>.<file_extension>.OCR`) for use with your Recognition project by testing a Standard OCR profile on a set of images and PDFs. You can also use an OCR data cache in conjunction with the Standard OCR-extracted content for tuning your Recognition project rules. Specifically, in Recognition Designer, you can use the OCR data cache as follows:

- Keyword classification testing (**Test** > **Classification Text** > **Test** > **Keyword Classification**)
- Text matching classification testing (**Test** > **Classification Text** > **Test** > **Text Matching Classification**)
- Unit testing (**Test** > **Unit Test**)
- Template testing (**Test** > **Template Test**)
- File definition testing (**Free From Designer** > **Settings** >  (**File definition test**))
- Saving and importing Free Form `*.OCR` files (**Tools** > **Free Form Designer** > **OCR Reading** > **OCR** > **Open Results File** | **Save Results**)

The OCR data cache file is loaded with its associated PDF or image files.

Note: If the OCR data cache option is not enabled or if the template, classification, or unit tests have an OCR engine other than the one required by the OCR data cache, then an OCR file generated by an associated OCR engine, if it exists, is used instead of the OCR data cache.

To generate and use an OCR data cache, in your **CaptureFlow**, you include the following modules in the given sequence:

Note: Dispatcher uses this behavior, including reusing the OCR data cache, only if the OCR data cache is created by Standard OCR.

1. Image Converter

   You use Image Converter to prepare Microsoft Office documents, PDFs, and image files for processing by Standard OCR. For a Microsoft Office document, you use Image Converter to convert it into a PDF or PDF/A and then split it into single pages. For original PDF and PDF/A documents and image files, you use Image Converter to simply split them into single pages. For all formats, you set Image Converter to keep the textual data, that is, maintain the text as text, so that Standard OCR can convert the text into an OCR data cache.

   To prepare your document for processing by Standard OCR, enable the following properties:

   - **PDF Options** > **Keep Input Textual Data**
   - **PDF Options > Output Destination = Single-page File**

2. Standard OCR

   You use Standard OCR to generate an OCR data cache, which contains the textual data extracted from the single-page PDF and image files that were outputted by Image Converter.

   For more information, see "Setting Up Extraction" on page 232 and "Setting Up Classification" on page 121.

1.6 What is Learning?

Intelligent Capture implements several intelligent data learning mechanisms that are able to analyze the existing documents and based on that analysis create classification templates without much designer effort. The learning mechanisms available with Recognition Designer include automatic learning, advanced learning, and production auto-learning.

1.6.1 Automatic Learning

Automatic learning is recommended for large volumes of images and numerous document classes. Automatic learning requires a large image base with images representing various types of documents. Automatic learning compares images two by two. An image is learned relative to all images. For each template, the system creates a folder with a subgroup of up to the first 50 images. The system selects a reference image from this subgroup, usually the central image, and this reference image is saved to the file `classifier.tif` in the template folder. During the compilation, up to 100 anchors are automatically created for each template, and the system determines the most pertinent anchors among all the images of the group. Then the system determines the top five anchors and those are the anchors used during classification. Anchors cannot be modified. Anchor information is kept for each template in the files `classifier.dpm`, `classifier.dci` and `classifier.dca` files. With automatic learning, a template code and name are manually assigned to each template.

Over-classification

Over-classification occurs when automatic learning creates two or more templates for the same graphic layout and can occur because of the following factors.

- Images originate from multiple sources—Images with the same layout can look graphically different if they come from different sources or printers. For example, a government form can have numerous variations if the form is printed from the web, a printed PDF, or a paper document obtained from a local office.
- Images have variable content—Images with the same basic layout can also have variable amounts of text or elements, such as tables. For example, invoices from the same vendor with a variable number of invoice rows on each invoice.

1.6.2 Advanced Learning

Advanced learning is recommended for medium volumes of images and when no relative learning of images against the other document classes is necessary. The images in the base need to be organized into sub-folders bearing the name of the document classes. Thus, advanced learning is the best option if there is already a process in place to archive scanned documents to a repository, in which case images can be exported into sub-folders bearing the name of the document class.

Advanced learning performs automatic successive learning loops at the level of each sub-folder and creates templates within each sub-folder. Each template is automatically assigned its code and name based on how sub-folders are named.

When choosing between automatic learning and *advanced learning*, mind the following:

- To prepare for advanced learning, you need to create as many sub-folders as there are document classes in the project.
- Since advanced learning does not compare the images relative to the entire image base, the resulting templates can be very different from those created with automatic learning. As a result, advanced learning may produce more templates (over-classification) than automatic learning and these templates are usually less discriminant than those created by automatic learning.
- Compilation time is longer with advanced learning than with automatic learning.

1.7 Adding the Advanced Cloud OCR Engine

Advanced Cloud OCR is a separate MSI and is available for download from the Intelligent Capture 21.4 (or later) product download site. The installer uses the standard Intelligent Capture installation folders for its destination. For installation instructions, see *OpenText Intelligent Capture - Installation Guide (ECPCORE-IGD)*. After installing Advanced Cloud OCR, you must add the Advanced Cloud OCR engine.

To add the Advanced Cloud OCR engine:

1. In Recognition Designer, click **Tools** and select **OCR/ICR Engine**.
2. Click **New**.
3. Click **Custom OCR**.
4. Type the preferred name for how the engine is displayed.
5. From the list, select **Advanced Cloud OCR**.

The Advanced Cloud OCR engine is now available to use. To set a region for processing Advanced Cloud OCR, see *OpenText Intelligent Capture - Installation Guide (ECPCORE-IGD)*.

1.7.1 Implementing LanguageHints and Debugging for Advanced Cloud OCR

Before you implement the *`LanguageHint`* and *Debugging* options, you must add the Advanced Cloud OCR engine. For more information, see "Adding the Advanced Cloud OCR Engine".

LanguageHint

If the service is having trouble detecting the correct language used in the image, you can provide a *`LanguageHint`* in the menu when creating a custom engine.

Note: *`LanguageHints`* must be in the *BCP-47* format.

Debug options

Debugging has two parameters, *`EnableDebugFile`* and *`DebugFilePath`*. Set these only if the engine is not working and you need more information for debugging purposes.

- To create a log file, set *`EnableDebugFile`* to "true."
- To stop creating a log file, set *`EnableDebugFile`* to "false."
- To save the log file, set *`DebugFilePath`* to the file path where you want to save the log file.

1.8 What is a Recognition Project?

A recognition project encapsulates the setup configuration of the advanced recognition modules and the resources required for these modules to handle their tasks. A recognition project is stored on the disk as a collection of files that are arranged in a folder structure. The components of a typical recognition project include:

1.8.1 Index Families

Index families are created and defined in Intelligent Capture Designer as *document types*. Index families are composed of index and table fields for extracting data from documents during production. The extraction behavior during production can be further customized with scripting. When defining index families, operators create fields and assign properties to fields that specify the extraction data characteristics. The index and table fields are then placed on the classification templates before production, so these templates become indexing templates.

A recognition project can be associated with one index family, or several, or none. In turn, an index family can be associated with one and only one recognition project. Index families and their fields serve for data extraction. If the project is not meant for data extraction, index families may be missing.

The number of index families needed for a project varies based on the number of documents classes being processed. A document class is presented by a set of fields to be extracted from images. A single index family can be appropriate for several templates in condition those templates belong to the same document class and a set of fields for extraction is the same for all these templates.

1.8.2 OCR Engines and Configuration Files

Recognition engines interpret images and return the equivalent text, barcode, or other information depending on the nature of the image data. Some recognition engines can recognize many types of characters and languages, while others can specifically recognize images such as barcodes. Selection of the appropriate recognition engine is important to analyze images properly and return the correct data. Intelligent Capture ships with several Optical Character Recognition (OCR) and Intelligent Character Recognition (ICR) engines. Recognition engines are designed to recognize a wide variety of specific types of printed characters.

Engine configuration files (`.reco`) are defined and associated with projects during setup. An engine configuration file is based on a single recognition engine. Each engine has customizable parameters that are saved in engine configuration files. A single engine can be represented by several configuration files, each containing settings appropriate for different projects.

Intelligent Capture includes predefined engine configuration files designed for many common tasks. These predefined files can be edited to suit specific needs, or users can create custom configuration files from these predefined files. The engine

configuration files are assigned during data extraction, so users do not select the recognition engines directly at that time. To improve recognition efficiency, configuration also enables assignment of confidence thresholds, filters, and can be based on language.

Recognition engine configuration files are assigned when fields are placed on templates. During testing, recognition engines can be reassigned for fields for help in selecting the most appropriate engine for the template.

1.8.2.1 Supported Recognition Types

The choice of an engine depends on the nature of the printed data. By matching the appropriate recognition engine to the type of characters to find on a document greatly increases data extraction success. The following recognition types are supported:

- **Machine printed**: Recognizes machine printed alphanumeric characters that have consistent, predictable shapes including fixed pitch, variable pitched, and kerned fonts. This recognition type is also referred to as Optical Character Recognition (OCR).
- **Precise machine printed**: Recognizes machine printed alphanumeric characters that have consistent, predictable shapes including fixed pitch, variable pitched, and kerned fonts. This recognition type is also referred to as Optical Character Recognition (OCR). This method favors recognition quality over speed.
- **Hand printed**: Recognizes alphanumeric characters that vary in shape, such as hand printed characters. This recognition type is also referred to as Intelligent Character Recognition (ICR).
- **Mark sense**: Recognizes checkmarks, Xs, or other marks placed in checkboxes. This recognition type is referred to as Optical Mark Recognition (OMR).
- **Barcode**: Recognizes industry-standard barcodes, including 1D and 2D barcodes. Barcodes are comprised of self-contained information encoded in the widths of printed bars and spaces.
- **Automatic**: Determines whether the characters are machine printed or hand printed, then applies that recognition type.
- **Cursive**: Recognizes handwritten characters, including signatures.
- **9 pins or 24 pins dot matrix**: Recognizes alphanumeric characters generated on a 9 pin or 24 pin dot-matrix printer.
- **MICR/CMC7**: Recognizes the code line for checks.
- **CAR/LAR**: Recognizes courtesy amount (amount in figures) and legal amount (amount in letters) on French and US checks.

Note: The recognition engine names have been changed relative to earlier versions. This name change improves the selection process, since names now indicate the appropriate use for the specific engines.

1.8.3 Classification Templates

Textual Templates

Textual templates are based on a full-text analysis of the collected data. They provide good accuracy while keeping the project size relatively small; you can also update them more easily than graphical templates.

Graphical Templates

Graphical templates are based on a graphical analysis of the collected data. They enable high-speed processing and are best used when images are reasonably stable.

Textual and Graphical Templates

Textual and graphical templates are based on both full-text and graphical analyses of the collected data. Although the project gains better accuracy by using both types of templates, the project can grow in size and performance might be slower.

1.8.4 Keyword Rules

Keyword templates and keyword-based classification serve for semi-structured and unstructured documents where graphic anchors may not work. To configure the keyword classification engine, a project defines a set of keyword rules, a unique reading zone (the whole image, an upper, middle, or lower third, or a specific area to be selected within the image), and an OCR engine to be used in that reading zone for full-page recognition.

A project can define as many keyword rules as needed. Keyword rules are composed of rule properties and keywords that are expected in a particular document class:

```
Rule A (keyword 1; keyword 2 ; keyword 3 ) + rule properties (priority, proximity
[number of characters])
```

A keyword can be a constant, an alphanumeric format, or a regular expression. Additionally, a keyword can be defined as an *isolated word* that must be found between spaces, as an *anti-keyword* to validate the template when this word is NOT found in the document, and as a *hit threshold* to evaluate a match between a recognized word and the keyword to the threshold value (%).

Rule properties include the *priority* and the *proximity*. The priority level is aimed at solving conflicts when an image matches several rules. A rule is associated with only one template which is why if an image matches several rules, it potentially matches several templates (conflicts). The proximity is an optional property that defines a maximum distance between the keywords expressed in number of characters. This property defines how near each pair of keywords in this rule can be found in the image.

During classification, if a project includes keyword templates, the keyword classification engine is activated and all incoming images, not relative to document

classes, are checked to comply with these templates. Each image is OCRed in the reading zone and rules are checked one after another. For each rule, the keywords are searched in the extracted data. A rule is true if all the keywords of the rule are found in the document. If the rule is true, the document is classified to the template that is associated with this rule. If an image is classified to more than one rule, the engine selects the one with the higher priority and classifies the image to the associated template.

1.8.5 Anchors

Anchors are used to define relative positions of fields on a template, as opposed to absolute positions on a template. When placing a field on a template, that position becomes absolute in terms of the template. If images in a production environment are offset relative to the template, due to scanning issues for example, the index position can be inaccurate for those images and recognition errors can occur. An anchor can be placed on a template relative to the field position. Anchors are placed on a static piece of information, like a logo or mark, that appears on all images in the same relative position to the index location. If one or more of the images in a process are offset relative to the template, the anchor enables OCR to find the static piece of information, determine the index position defined relative to that anchor, and extract the correct information.

Note: When more precise anchors are necessary, such as when the images in a batch are closely related or have relatively minor variations (such as checks from different banks), High Precision Anchors (HPA) and HPA templates are used.

1.8.6 VBA Scripting

To enhance and customize the advanced recognition logic of your capture process, you can create event-based scripting to be run by particular production modules.

Two scripting models are currently supported:

- **Document type scripting (.NET)**: This scripting can be implemented for production modules that support the new UIMdata object model. These modules are Extraction, Identification, and Completion.

  Document type scripting is created on the .NET platform (C# or VB.NET). This scripting is designed in a third-party programming environment, has no relation to Recognition Designer, and not associated with the DPP project.

- **Recognition (Dispatcher) scripting (VBA)**: This scripting can be implemented as part of the DPP project for Classification. VBA scripting is the only way to customize the advanced recognition modules that support the old Dispatcher object model. In old-version DPP projects, Recognition and Validation (legacy modules, no longer supported) implement only VBA scripting. The Extraction module supports both .NET scripting and VBA scripting in the part that works during data extraction (Dispatcher Batch API).

  Recognition scripting is designed using VB-COM or VB.NET in the script editor integrated with Recognition Designer.

For more information on the above scripting models, refer the following sections in *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*:

- *Document Type Scripting*: Describes how to create document type scripting
- *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*: Describes Dispatcher scripting (VBA)

1.9 Planning a Recognition Project

A common mistake when configuring data capture is to focus on document identification, or classification, and then on data extraction. This approach often leads to poor decisions which requiring rollbacks. Instead, use these steps when planning a recognition project:

1. Focus on analyzing the document classes, characteristics, and volumes. This analysis is critical for large projects, such as those processing more than 25,000 pages per day or more than 15 document classes.
2. Determine those documents that require only classification, and those documents that also require data extraction. Data extraction requires that templates be more granular than for classification only. For details, see "Analyzing the Document Flow" on page 32.
3. Consider which data extraction solution fits the business requirements to identify the classification technology to implement. For examples of various business cases, see "Examples: Addressing Business Requirements" on page 33.

1.9.1 Analyzing the Document Structure

As a first activity in implementing classification, examine the document characteristics and sort documents into the following groups:

1. Structured documents

Structured documents have exactly the same structure and appearance, meaning the same graphic layout. Data items are located at the same place on all documents. Examples of structured documents are questionnaires, tests, insurance forms, and tax returns.

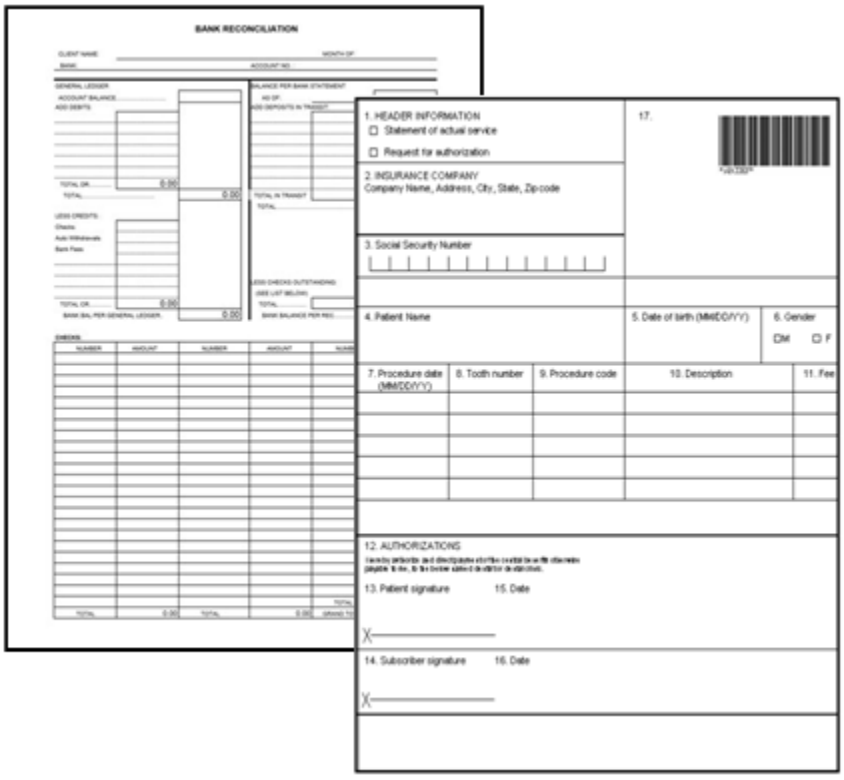


Figure 1-2: Structure document example

2. Semi-structured documents

Semi-structured documents have the same logical structure but their appearance can change. This means they contain the same data items to be extracted but data items are not present in all documents and are usually not found on the same areas of the documents. Examples of semi-structured documents are invoices, purchase orders, bills of lading, and contracts.

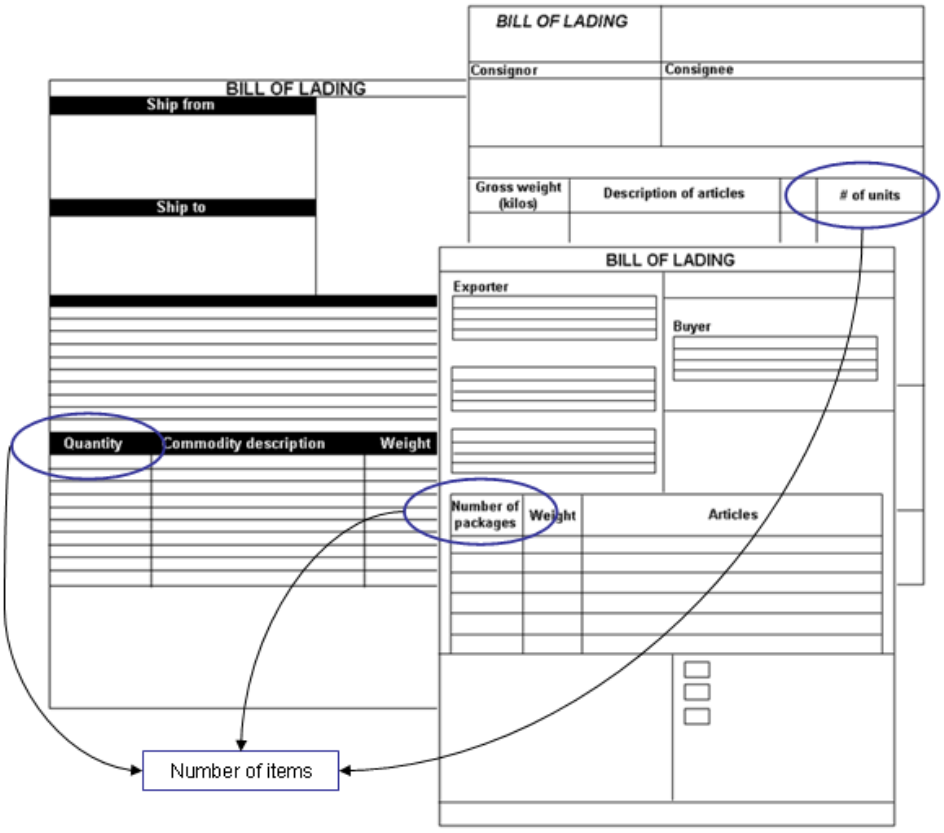


Figure 1-3: Semi-structured document example

3. Unstructured documents

Unstructured documents have flexible structure and appearance. Each page in a document has a unique structure. There is no repeatable graphic layout. Examples are letters, articles, or white mail. White mail is a collection of letters received from customers with a variety of items enclosed such as complaints, inquiries, and orders. White mail is usually enclosed by customers in their own envelopes instead of the reply envelopes provided by sellers. Another example of white mail is a business reply card with an enclosed check for payment.

Figure 1-4: Unstructured document example

1.9.2 Analyzing the Document Flow

To analyze the document flow, an efficient way is to create a spreadsheet to check and track these characteristics:

- **Nature**: Structured (application forms, medical forms), semi-structured (checks, bills, invoices) or unstructured (contracts, letters, reports, statements, attachments).
- **Lifecycle**: Collection is ongoing (for example, in a mail room) or proceeding from archives.
- **Page number**: Minimum, maximum and average number of pages.
- **Duplex**: Printed on one side or on both sides.
- **Source**: Original forms (printed by the company), photocopies, online download printed by users, FAX.
- **Version**: Existing in different versions (per state, per company department) or when a new version becomes available periodically (yearly for instance).
- **Page order**: Multiple page documents with pages coming in a definite order or in an unspecified order.

- **Data extraction**: Classification only or classification and data extraction.

For example, if the requirement is that all invoices in the document flow be classified as invoices, whatever the vendor, then one "invoice" template is sufficient. If all invoices must be classified with their vendor, then it is necessary to create one template per vendor.

To classify documents without extracting data, a unique template per document class is sufficient. For data extraction, more graphic templates are required to define and place fields accurately onto the templates for data extraction.

Table 1-3: Example: Templates for classification only vs. templates for classification and data extraction

Documents	Template for classification without data extraction	Templates for classification with data extraction
Invoices from Vendor1	Invoice_DocType	Invoice_DocType1
Invoices from Vendor2		Invoice_DocType2
Invoices from Vendor3		Invoice_DocType3

1.9.3 Examples: Addressing Business Requirements

This section includes examples to discuss the features that are recommended to address the most commonly known business requirements.

1.9.3.1 Examples with Structured Documents

Structured documents are best addressed with graphic classification (standard and *HPA*) and zonal recognition. Create a graphic template for each different graphic layout in the production flow. Then, determine how many document classes exist in the production flow, since this the number of document classes controls the number and type of index families to create. Finally, define which index families to associate with the different templates.

Dental Claims

Documents to address are dental claims coming from several known dental clinics. In this case, it is possible to create one graphic template for each known dental clinic. All clinics comply with the same nomenclature, so only one index family is necessary. Associate all templates with the same document class, or index family.

Here are the main activities to address these documents:

- Create the standard templates for all the dental claims automatically.
- Add the index family that contains all the fields of the nomenclature.
- Associate all templates with the index family.

- On each graphic template, place each field from the index family onto the template. Place each field to the precise position where the field value is found. Even if all the claims contain the same fields, they appear at different locations.

When using only graphic templates, if dental claims from unknown (new) clinics enter the document flow, there are no existing graphic templates to process them. If a number of new unknown claims are anticipated to enter the production flow after the deployment of the project, you may decide to process dental claims as semi-structured documents.

Questionnaires

Sometimes, questionnaires from different companies, working in different business domains, are sent to customers. Customers fill out and send the questionnaires back. There are as many document classes as there are companies because they all work in different business domains.

Here are the main activities to address these documents:

- Add one index family for each document class.
- Create one graphic template for each questionnaire.
- Associate each template with its respective index family.
- On each graphic template, place each field from the index family onto the template. Place each field to the precise position where the field value is found.

1.9.3.2 Examples with Semi-structured Documents

Semi-structured documents are addressed by combining graphic templates and free form templates. Capturing data on semi-structured documents involves associating documents with the correct index family by identifying the document class and extracting the appropriate fields.

This section discusses the main steps to address three examples: mixed documents (purchase orders and bills of lading), invoices, and contracts.

Additional considerations include:

- Graphic templates are more accurate than free form templates. However, if the project contains lots of graphic templates, consider that creating graphic template can take much longer than designing free form rules.
- Designing free form rules requires more expertise than designing graphic templates.
- Gaining time to put a project in production earlier is possible by creating graphic templates for the most frequently recurring documents. Then create free form templates for less recurrent documents.

Bills of Lading

A bill of lading is a document covering a shipment. Data mainly includes consignor and consignee names, description and weight of goods, rate and, total charges.

Considering a case where 85% of bills of lading come from 100 consignors and the remaining 15% come from 500 consignors, there are 600 potential graphical layouts. All 600 appearances correspond to one single document class as they all contain the same data. To address this case, create graphic templates for the 100 most recurrent consignors, as they generate most of the production flow. Use free form rules for the 500 other consignors that generate 15% of the flow.

With this strategy, 85% of the production flow is processed with the more accurate graphic templates. Design time is saved by developing free form rules for the 500 consignors instead of spending time creating 500 graphic templates.

Here are the main steps to address this sample business case:

- Create graphic templates for all the consignors that account for 85% of bills of lading.
- Create a generic template and set is as the default template. The remaining 15% bills of lading for which there are no graphic templates are classified to the default template.
- Add one index family for the bill of lading document class containing the fields for extracting from all bills of lading.
- Associate all the templates (graphic and default) to the index family.
- Place each field from the index family onto all the graphic templates. Place each field in the precise position where the field value is found.
- Create free form rules to extract data from the bills of lading classified to the default template.
- Associate the free form rules with the generic default template.
- Send the project to production.

Invoices

This example addresses a use case where invoices come from more than 100 different vendors. Extracted data is the same on all invoices, thus only one document class is needed.

Here are the main steps to address this sample business case:

- Create as many graphic templates as there are vendors.
- Add only one index family, as there is one document class.
- Associate all templates with the same index family.

Create graphic templates with zonal recognition for the most frequent invoices. For the other invoices, use keyword classification and free form templates. This way, the project is ready sooner for deployment: designing keyword classification and free form rules is time saving compared to creating graphic templates.

Contracts

Documents to address are medical insurance contracts and mortgage contracts. The two types of contracts contain different data to extract, which means that there are two document classes. Contracts are typically composed of many pages and each page mostly contains textual content. The best classification method for these documents is text matching. For information, see "Text Matching Classification Uses" on page 79.

Here are the main steps to address this sample business case:

- Create text matching templates.
- Add one index family for each document class.
- Associate the templates with their respective index family.

1.9.3.3 Example with Unstructured Documents

Unstructured documents are usually addressed by keyword classification and free form templates. This section provides recommended activities for unstructured documents, such as white mail.

White Mail

White mail includes complaints, inquiries, and orders. Each of the three document classes (complaints, inquiries, orders), must be identified and routed to the appropriate data extraction settings. There is no recurrent graphic appearance and most documents have a unique layout. For this reason, graphic templates are not a good choice.

Here are the main activities to address this sample business case:

- Add an index family for each document class and associate it to the corresponding generic templates: complaints, inquiries, and orders.
- Create keyword classification rules to identify complaints, inquiries, and orders and associate the keyword rules for complaints to the generic template created for complaints.
- Configure free form rules for each document class.
- Associate the free form rules for complaints with the generic template created for complaints. This assumes the generic template is already associated with keyword rules for identifying complaints. Similarly, associate free form rules for inquiries, and for orders, with the applicable generic template where keyword rules are identified for each document class.

1.10 High-Level Steps to a Project in Production

The following table includes the high-level steps for developing a recognition project and deploying it in the production environment.

Note: Steps related to PAL do not apply to Core Capture users.

Table 1-4: High-Level Steps for Developing a Recognition Project

Stage	Job	Related topics
1	Analyzing business requirements and planning a project	"Planning a Recognition Project" on page 29
2	Creating a project or importing an existing project in a capture system	*OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)*
3	Specifying the project settings in the **Project Options** window of Recognition Designer	"Defining Production Folders" on page 47
4	Adding index families to the project	Adding Index Families
5	Defining recognition engines	"Defining OCR Engines" on page 108
6	Creating classification templates, testing and fine-tuning classification	"Classification" on page 51
7	(Optional) Adding zones and anchors, testing and fine-tuning recognition	"Zonal Recognition" on page 134 "Testing Data Extraction" on page 198
8	(Optional) Adding Free Form rules and keywords, testing and fine-tuning recognition	"Designing Free Form Rules" on page 155 "Testing Data Extraction" on page 198
9	Configuring the Classification, Identification, and Extraction module settings in the **Project Options** window of Recognition Designer	"Setting Up Classification" on page 121 "Setting Up Classification Edit (Deprecated)" on page 127 "Setting Up Extraction" on page 232

Stage	Job	Related topics
10	(Optional) Developing the project scripting and scripting for classification	*OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)*
11	(Optional) Enabling and configuring Production Auto-Learning in the **Project Options** window of Recognition Designer	
12	Compiling the project, resolving the issues	"Compiling a Project" on page 44
13	Deploying the project to the test environment	"Deploying a Project" on page 45
14	Setting up the steps of the deployed CaptureFlow to link the project	*OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)*
15	(Optional) Developing and testing scripting for the Extraction module	*OpenText Intelligent Capture - Designer Guide (ECPCORE-CPD)*
16	(Optional) Testing and fine-tuning Production Auto-Learning	
17	Deploying the project to the production environment	"Deploying a Project" on page 45

Chapter 2

Managing Projects

The topics in this section describe the operation that you can perform on recognition projects.

2.1 Project Structure

Each recognition project must reside in its own project folder and have its files arranged in a particular folder structure. This structure is created automatically when the project is saved for the first time. Manual deletion or relocation of any of these files or folders can break the project and make it unusable in production mode.

Files	Path	Description
<ProjectName>.dpp	Intelligent Capture *<version>*/Default/GlobalData/Recognition/*<ProjectName>* (the root project folder)	Contains the main settings of the project.
<ProjectName>.dps	Intelligent Capture *<version>*/Default/GlobalData/Recognition/*<ProjectName>* (the root project folder)	Contains the settings of the production modules.
*<ProjectName>*_Migration.log	Intelligent Capture *<version>*/Default/GlobalData/Recognition/*<ProjectName>* (the root project folder)	Contains the migration log for the global OCR engine settings. This file is generated in this folder after importing a project.
	Intelligent Capture *<version>*/Default/GlobalData/Recognition/*<ProjectName>*/Cache	Contains data related to standard templates. This data is computed when the project is compiled.

Files	Path	Description
	Intelligent Capture *<version>*/Default/GlobalData/Recognition/*<ProjectName>*/IdxClasses	Contains the index families definition files (.index and .xindex) and the index families scripts (.bas). Additional script files referenced by inclusion from the project script might also reside here.
	Intelligent Capture *<version>*/Default/GlobalData/Recognition/*<ProjectName>*/Models	Contains as many subdirectories as there are templates in the project. Each subdirectory contains the following files and directories: • classifier.tif Template reference image. • index.tif Index family reference image. • \Base Image base for the standard templates. • \test Image test base for the standard templates.
	<ProjectName>\Resources	Contains the FullTextClassifier.xml file for using optional keyword rules in the project.
	<ProjectName>\Resources\OCR	Contains the OCR engine settings (.reco files), the free form definition files (.dft), the field-specific types definition files (.tft) and the full text relations, field-specific types and full text table script relations files (.bas).

Files	Path	Description
	<ProjectName>\Resources\Scripts	Contains the project script (BAS), and the breakpoint files (DBP). Script files referenced by inclusion from the project script might also reside here, but must be referenced using an asterisk in the include statement to indicate the script resides in this directory. The correct syntax for the include statement is: `#Uses "*ScriptFileName.bas`

2.2 Creating a Project

To create a recognition project, run Intelligent Capture Designer. This tool creates an empty recognition project and associates it with a particular capture system, which is a requirement for designing a capture process. All projects of a selected capture system are stored on the disk in the capture system directory at `%USERPROFILE%\Documents\Intelligent Capture <version>\<Capture_System_name>\GlobalData\Recognition`.

This folder on the disk serves for development purposes only. When the project is ready for testing, its copy needs to be sent to the testing environment. Later, the tested project copy needs to be sent to the production environment.

2.2.1 Creating a Project (Dispatcher Manager Only)

Creating a recognition project for the first time with the New Project Wizard requires definition of the **Name**, **Author**, and **Company**, and also allows addition of optional notes. The project also takes a version number. Use Dispatcher Manager to create a recognition project. The New Project Wizard performs automatic learning of documents.

To create a project:

1. Start **Dispatcher Manager**.
2. Select **File** > **New**.

   The New Project Wizard opens.
3. Click **Start**.

   The **Project Parameters** window appears. Two options are available:

   - Click **Next** to begin to create standard templates automatically by means of the wizard.

- Click **Empty project** to bypass automatic creation of standard templates. An empty project appears in Dispatcher Manager. Create templates manually or import them from an existing project.

4. Use the **Save As** command to save the project in a new project folder.

2.3 Opening a Project for Editing

You can open an existing recognition project using Intelligent Capture Designer.

After the project is open in Recognition Designer, it is locked, and the *CLK* file containing information on the user who currently works with the project appears in the project folder. Thus, only one user can work on one recognition project at a time.

2.4 Importing a Project

You can import an Dispatcher 6.x project using Intelligent Capture Designer.

2.5 Defining Project Options

To define project settings, select **File > Project Options**.

The **Project Options** window provides a collection of settings particular to the project or to a certain advanced recognition module. These settings determine how Recognition Designer and the modules handle information during production.

2.5.1 Using the Project Options General Tab

The **Project Options** window **General** tab enables setting up version of the project, location for production files, and general project information.

To define general project settings:

1. In the **Project Options** window, select the **General** tab.
2. Under **Main options**, specify a **Project name** and, if needed, add optional **Author** and **Company** information.
3. Set the **Project version No** as the version is not incremented automatically, so it can be useful to specify **major** and or **minor** version numbers.
4. Under Production, specify the **Production directory** where production files are located, **Production report directory** if reporting is enabled, and the **Backup directory for production project** as the location for backed up production files. Select **Check all templates are linked to an index family before sending to production** to ensure that the procedure to move the project to production will not be interrupted if not all templates have an associated index family assigned.
5. Click **Advanced information** to review additional project information, such as lists of templates and index families, project image data, and others.

2.6 Modifying Project Information

After you have created a new project or imported an existing one, you can modify the project information.

To modify project information:

1. From Dispatcher Manager, select **File > Project Options**.
2. In the **Project Options** window, select **General**.
3. Modify required project information in the **Main options** pane.

2.6.1 Verifying Project Size and Location

Once the project design is finished, you need to create a production project, which contains the project parameters but not the template images. The production project is saved to the *production directory*, and can also be copied to a backup directory, assuming that the project has already been sent to production at least once.

To check the project directory and the size of the project:

1. From Dispatcher Manager, select **File**> **Project Options**.
2. In the **Project Options** window, select the **General** tab.
3. To view information about the project, click **Advanced information** and review the **Project directory** path, the **Project size** (production data), and the **Total development project size**.
4. To view information about the production project, review the **Production directory** and the **Backup directory for production project** information displayed in the **Production** pane.

2.7 Deleting a Project

You can delete an existing recognition project using Intelligent Capture Designer. The project will be removed from the capture system and the project files will be permanently deleted from the disk space. If a project to delete includes index families that are defined by document types, those document types remain on the disk but become unreadable.

2.8 Saving a Recognition Project

- Select **File** > **Save**.
  The *DPP* file is saved to the project folder.

2.9 Compiling a Project

You need to compile the project before sending it to the test environment or to production. Also, recompile the project whenever adding, deleting, or merging a template, changing a template code, and when images are rotated.

The compilation is used to compute internal properties used during a classification step for Standard and HPA templates.

Note: Because of the algorithm complexity, this computation may take a long time for a large project. In addition, after adding new standard and HPA templates, the compilation must be run again. It is recommended have the compilation cache activated by default as it helps to decrease the compilation time because the compilation can read intermediate computation result from the cache.

To compile the project:

1. Before compiling a project, make sure the **Compiling cache** option is enabled.
   a. Select **File > Project Options**, navigate to the **Classification** tab.
   b. In the **Compiling cache** pane, select the **Activate** option.

   Note: If you disable the compiling cache, the compiled files created prior to disabling the compiling cache are kept in the cache directory. To empty the compiling cache, click the **Empty** button in the **Compiling cache** pane. Deactivating and purging the cache might be useful to save some disk space.
2. Save the project.
3. Select **Classification > Compile**.
4. In case of an unexpected error during the compilation, try to empty the compiling cache. This enforces Recognition Designer to rebuild the cache, which could be useful if the cache has been corrupted.

2.10 Exporting a Project (Dispatcher Manager Only)

Export operation available in Dispatcher Manager only saves the project as a ZIP file to be later used for development or production purposes. Before exporting a project from Dispatcher Manager, you need to firstly compile and save it using the following options:

- Select **File > Export the project** to compress and export the whole development project. All the project parameter files are exported together with the template image bases.
- Select **File > Export the project > Compress for production** to export the project for production. The project parameter files are exported without the image bases.
- Select **File > Project Options > General** and specify a backup directory for the production project to automatically create a backup of the project when it is moved to production.

2.11 Deploying a Project

Project deployment is necessary to send your recognition project to a testbed or to a production environment. While other capture system components can be deployed by Intelligent Capture Designer to the test or production server in a bundle, the recognition project must be deployed separately, using Recognition Designer or manually. During deployment, the compiled recognition project is copied from the design environment to a shared location accessible to all machines that will run 'recognition' modules during testing or production.

The procedure to follow applies to first-time deployment and successive deployments.

To send a project to production:

1. Create a shared production folder on the disk or in the network. Grant access rights to the users that will be running the 'recognition' modules in the network and sharing your recognition project.
2. Specify the path of the shared folder in the global options of your capture system:
    1. Open Intelligent Capture Designer and select **System** and the **Configuration** tab.
    2. Select **Other Options** in the **Configuration settings** drop-down list.
    3. Select the **Global Options** item.
    4. In the **File Management** section, specify the *UNC* path of the shared folder in the **RecognitionProjectSharedDirectory** parameter.
    5. Save the changes.
3. Compile the project as described in "Compiling a Project" on page 44.

4. Select **File** > **Project Options** and specify the **Production** options in the **Project Options** window:

   - **Production directory**: Specify the path of the shared production folder created for the recognition project.
   - **Production report directory**: If reporting is enabled, specify the path for production reports.

     If the report directory is not specified, the reports are added to the production directory by default.
   - **Backup directory for production project**: Specify the path for the backup copies of the project.

     If the backup directory is not specified, the backups copies are added to the production directory by default.

   These directories are described in topic "General Tab" on page 277.

5. Optionally, select **Check all templates are linked to an index family before sending to production**. If a template has no associated index family, the move to production will be interrupted.
6. Close the **Project Options** window.
7. Select **File** > **Send to production**. The deployment is complete when the message **Sent to production successfully** appears.
8. Set up the 'recognition' steps of your CaptureFlow to use the deployed recognition project:

   - Classification and Identification: Point out the UNC path of the shared production folder.
   - Extraction: Point out the `DPP` file name of the project.

   For more details, see a specific module guide.

At each successive deployment, a message appears prompting to compress the project. This is recommended to enable rollbacks. Compressing the project creates a backup copy (a `ZIP` file) of the previously deployed project before it is overwritten by the newly deployed project. The project can also be deployed without creating a backup copy: when prompted, select not to compress the project.

2.11.1 Defining Production Folders

1. In the **Project Options** window, select the **General** tab.
2. Under **Production**, specify the following settings:
   - **Production directory**: Specify the path of the shared production folder created for the recognition project.
   - **Production report directory**: If reporting is enabled, specify the path for production reports. If the report directory is not specified, the reports are added to the production directory by default.
   - **Backup directory for production project**: Specify the path for the backup copies of the project. If the backup directory is not specified, the backups copies are added to the production directory by default.
   - Select **Check all templates are linked to an index family before sending to production** to ensure that the procedure to move the project to production will not be interrupted if not all templates have an associated index family assigned.
3. Click **Advanced information** to review and/or export the list of templates and to view other information related to the project.
4. Click OK to save the production folder settings and close the **Project Options** window.

2.11.2 Setting Up Folder Management

Folder management is useful for splitting the batch into interrelated sub-elements so that related documents can be written to the same folders. Common examples of a folder are a multi-page invoice, enclosed documents, and other related elements. If folder assembly is not enabled, each document in the batch is considered as an individual folder.

The folder binding field is the index field which initiates creation of a folder during classification. This option is most useful to combine multiple-page invoices based upon their invoice number. To split the document flow into folders, a comparison is made between the values read on the different documents. A folder is created whenever the folder binding field value on a document is different from the previous document, or when two documents have the same folder binding field value but different template codes.

To define the folder management settings:

1. In the **Project Options** window, select the **Folder Management** tab.
2. Select **Enable folder creation** so that documents are grouped into folders during production. If this option is not selected, each document will form one individual folder. During classification, Recognition Designer can use separators to enable folder assembly. For information on how to set up

separators, see "Assigning Separators to Templates to Enable Folder Assembly" on page 119.

3. Under **Folder creation using a folder binding field**, select **Enable a folder binding field** to select the index field to be used as the folder binding field. The list box contains the index fields to be pre-indexed as defined in the **Classification** tab, **Pre-index following fields**.

4. The **Folder populated field** enables assignment of one or more field values to all documents in the folder. For example, if an invoice date is identical on all the invoice pages of all the documents in a folder, then it would be useful to automatically assign this value, after validation, to all the documents in the folder.

   a. Click the **+** button.

   b. In the **Folder populated field** window, select from the **Available fields** and click **Add**. The available fields are grouped by index family. As soon as a field is added to the list of **Selected fields**, it no longer appears in the list of available fields.

   c. Click **OK**.

2.11.3 Recommendations for Deploying the Project

This section offers recommendations for first-time deployment and successive deployment.

First-time Deployment

- When you send a recognition project to production in Recognition Designer, the entire development environment (all folders) is copied to the selected production folder. However, Recognition Designer does not copy the image base of the graphic templates as these images are not used in production. The deployed *<ProjectName>*\Models sub-folder keeps only reference images (classifier.tif and index.tif) for each template. It is recommended to keep to this approach when deploying the recognition project manually. The structure of the project folders is described in section "Project Structure" on page 39.
- It is recommended to deploy all other capture system components to the production server after the recognition project has been deployed and available for use. This deployment order is recommended to prevent the batch creation before the recognition project can be found in the specified shared location.

Successive Deployments

The following recommendations apply to redeploying a project with modified templates:

- Before you send the updated project to production, stop the 'recognition' steps that use the current project. Restart these steps when redeployment is complete.

- Before you send the updated project to production, make sure that the current project is not in use by any batch. All the batches using this project must be completed or deleted before redeployment.

These recommendations prevent any document in a batch from being classified and then extracted using different versions of a template. This can happen because of the difference in how 'recognition' modules access templates in production. Classification and Identification load all templates from the project at startup and use these templates until the batch is completed. The updated templates become available for the next batch. Extraction loads a template each time a template is required to process a document. If the project is redeployed, Extraction can access the updated templates immediately.

Chapter 3

Classification

This section explains how to set up and tune the different classification methods. Indications of expected classification are provided to help you compare between the different methods. For each method, recommended uses are indicated.

Textual classification locates the set of characters (labels and its value) that appear in the same place in two different documents. If the numbers of matching words are high, they may belong to the same class. Such algorithm takes small variations in text (for example, <*Invoice*> and <*Invoice:*>) into account as well as *floating fields* (when the page has a table in it, fields below the table are relative to the bottom of the table).

When starting a project, you may also want to read the *Addressing Business Requirements* section to decide which method best fits your document types.

Before setting up any classification method, a preparation phase is recommended namely to analyze the document flow, build images bases and deal with scanned images possible issues.

3.1 Choosing the Document Identification Technologies

After the data extraction technologies are planned for all document classes, choose the document analysis technologies that are appropriate:

1. Analysis of the graphical layout of the document
2. Analysis of the textual content of the document

The following table indicates the combinations of classification and extraction technologies mostly used depending on the nature of documents.

Table 3-1: Classification and Extraction Technologies Based on Document Structure

Nature of Documents	Data Capture Technologies
Structured documents	Standard and *HPA* classification with zonal recognition.
Semi-structured documents	Standard and HPA classification with zonal recognition. Keyword classification and text-matching with free form recognition.

Nature of Documents	Data Capture Technologies
Unstructured documents	Keyword classification with free form recognition.

Analysis of Graphical Layout

The Classification module identifies documents by analyzing their graphical layout. It recognizes a document that looks like another one seen before (global image analysis) or that contains a specific pattern, like a logo (local image analysis). This type of analysis is recommended for structured documents.

Graphical layout analyzed using the following technologies:

- **Standard classification**: Identification of documents using a global image analysis (graphical layout analysis). Intelligent Capture features automatic learning to automatically create standard graphic templates based on new documents entering the system.
- **HPA classification**: Identification of documents using a local image analysis so that identification is based on a specific local area, such as a logo or a title. Intelligent Capture enables converting standard templates to HPA templates.
- **Handwritten classification**: Identification of documents based on handwriting content.

Analysis of Textual Layout

The Classification module analyzes the keywords and text blocks present in the documents. It recognizes a document that contains a specific set of keywords or a similar sequence of characters (text blocks). This type of analysis is recommended for semi-structured and unstructured documents.

Keyword analysis is performed in Classification using the following technologies of document identification:

- **Keyword classification**: Identification of documents based on keywords. Requires a full text recognition engine to extract document information (keywords). Requires preparing settings for searching keywords in the document.
- **Text matching**: Identification of documents based on text blocks. Requires a full text recognition engine to extract and match document information (text blocks).
- **Textual templates**: Production auto-learning algorithm-based templates.

For example, in an incoming mail project, the following ratios have been observed and are given for indications only: standard and HPA classification are used to classify 55% to 90% of documents, keyword classification 5% to 20% of documents and Text Matching 15% to 40% of documents.

Find examples of business requirements with their most appropriate classification methods in section "Examples: Addressing Business Requirements" on page 33.

Processing Speed

Classification performance varies greatly depending on many variables including hardware, settings, and image quality.

Global or local image analysis technologies (HPA and standard) are much faster than the keyword analysis. This is because keyword analysis requires a full-page *OCR* step to retrieve the textual content of the images before keywords can be searched. Full-page OCR takes 1 to 10 seconds per image depending on the number of characters in the image, complexity of the image layout, and the *CPU* speed.

The overall processing speed with HPA and standard templates depends on the number of templates and the number of HPA anchors.

The table next indicates expected classification speeds for each classification method. These ranges are provided for information only and should be used exclusively for comparing classification methods. Understand that because text-based classification requires full text OCR prior to classification, this step adds additional processing time of 1 to 2 seconds per image.

Table 3-2: Classification Speed

Classification method	Speed
HPA	540 images per second depending on the number of HPA templates and anchors.
Standard	530 images per second depending on the number of standard templates.
Handprint	1020 images per second depending on the number of pixels in the image.
Keyword	320 images per second depending on the number of keyword rules, not including OCR time. Allow 1 to 2 seconds to perform full-page OCR prior to classification.
Text-matching	320 images depending on the number of reference images, not including OCR time. Allow 1 to 2 seconds to perform full-page OCR prior to classification.

3.2 Preparing Image Bases

Image bases are large amounts of images that are representative of the document classes in a production flow. Image bases are required for automatic learning, for fine-tuning classification templates, and for testing classification.

3.2.1 Recommendations for Creating Image Bases

The quantity of images recommended for the image base depends on the variability of documents. The best recommendation is that images must be representative of the document flow. The quantity of images matters but is less important than ensuring that the types of documents anticipated in the workflow are present in the image base.

Note: PDF files can also be used as a basis for templates.

For example, if the project has only two forms, 100 images is ideal for each form but 30 might be sufficient to create templates. Fine-tuning template settings requires more images. When fine-tuning, focus on the templates that are used most frequently in production. For these templates, high classification and recognition rates are expected. For these templates, the image base must contain more images than for templates that are not often present in the production flow. For these infrequent documents, it is acceptable to have them processed by operators.

Ensure that the image base has images for all document classes, and for all versions of document classes if some documents have several versions. Generally, gather at least ten images per document class and ten images per version of the document class. If there are multiple sources for a document class, gather a minimum of three to five copies of each source. Ideally gather the equivalent of several days of production documents.

Compose the following bases from the overall image base:

- **Learning base**: Used exclusively to create the templates with automatic learning. Never used to fine-tune templates.
- **Evaluation base**: Used to fine-tune classification templates. Images in the evaluation base must be sorted per document classes. Fine-tuning usually includes merging templates, converting some to *HPA*, and adjusting the pre-classification and decision thresholds.
- **Test base**: This base is optional. It is useful to evaluate the expected classification rate on the first day of production. It is mostly recommended for daily production volumes of 25,000 documents or more. It is usually a subset of the evaluation base. Use it to carry out a final test of the templates before deploying the project. Sort images by document class.

Consider the following recommended volumes of images in the different image bases:

- **Learning base**: Contains 50% - 90% of the overall image base. At least 5000 images, one day of production. or 50 images per document class variation, whichever is the smaller. The base should not exceed 40,000 images. If there are many images available per document class, allow for 50% of images in the learning base and 50% in the evaluation base. If there are few images available per document classes, allow for 90% of images in the learning base and 10% in the evaluation base.
- **Evaluation base**: Contains 10% - 50% of the overall image base. Allow for no more than 1,000 images per document class. Usually contains 1,000 - 5,000 images. Do not exceed a total of 10,000 images.
- **Test base**: Allow for 10 to 100 images per document class variation. Do not exceed a total of 5,000 images.

3.2.2 Improving Image Quality for Learning

Automatic learning is sensitive to the quality of scanned images. Defaults of scanned images can cause automatic learning to create too many templates for a given set of images. This excess of templates is known as over-classification. It usually occurs when some images look different graphically but are similar and belong to the same template. Over-classification happens with skewed scanned images, when black borders are present on scanned images, or when scanned images exist in different sizes for the same layout. Black borders around the edge of the image often result from photocopiers, scanners, and fax machines. Scanners, photocopiers, and fax machines can reduce the image beyond its native size or enlarge it. All these variations can result in over-classification.

It is a best-practice recommendation to run automatic learning with scanned images already processed by the image correction step planned for production. This way, the images used for automatic learning are representative of the production flow. This step is important to get correct classification and data capture. For Recognition Designer, use the Image Processor module.

Note: A resolution of 200 *DPI* is recommended, although a resolution of 300 DPI may be required for all images if some or all contain small print. The quality of the images greatly affects the accuracy of classification and data extraction.

Here are recommendations to solve some of the most common problems encountered with scanned images:

Black Borders around the Edge of the Image

Implement the Image Processor step to clean up black edges.

Downsized or Upsized Images

Solutions depend on the business requirements: classification only, classification, and zonal recognition or, classification and free form recognition.

- **Classification only**: Use keyword classification whenever possible. Keyword classification is based on a textual analysis of the document and not on a graphic analysis, so images can be classified whatever their sizes.
- **Classification and zonal recognition**: If two templates for the same layout are created because there are two sizes of images, keep one template for each image size and duplicate the fields on all the templates. Zonal recognition is preceded by a standard or *HPA* classification, which is based on a graphic analysis that requires a constant image size. The image size must be the same as the template for accurate field detection on the image.
- **Classification and free form recognition**: Create a generic template to be associated with keyword classification and free form rules.

A production solution is to implement the Image Processor step to clean up scanned images and to scale images in preparation for data capture operations.

Skewed Scanned Images

In production, most distorted images cannot be classified. Some slightly skewed images are classified but data extraction usually gives poor results.

- When creating templates with automatic learning, keep only the templates that are created with corrected or enhanced images and delete all templates created with skewed scanned images. Reduce this effort by running automatic learning on scanned images already processed by an Image Processor module to be implemented in production.
- In production, implement the Image Processor step to deskew scanned images.

3.2.3 Project Resolution

All images in a recognition project (including Production Auto-Learning) must have the same resolution referred to as the *project resolution*. A resolution of 200 *DPI* is recommended, although a resolution of 300 DPI may be required for all documents if some or all contain small prints.

Notes

- In Production Auto-Learning, images that do not match the project resolution are not collected (that is, they are skipped).
- In production, the resolution of images can be different from the resolution of the associated project.

You use **Image Analyzer** to verify the image resolution and convert images to the required project resolution if required.

Creating a recognition project for the first time with the New Project Wizard requires selection of reference images that are used to automatically create standard templates. The resolution of these reference images determines the project resolution. To add new templates to the project with the Project Update Wizard,

these images must match the project resolution. For additional information, see "Converting Images to the Project Resolution" on page 57.

The project resolution is shown in **File > Project Options** > **General** > **Resolution (in dpi)**.

3.2.4 Converting Images to the Project Resolution

1. Select **Tools > Image analyzer**.
2. Select **File > Load images** or **File > Load tree**.
3. Select **File > Analyze**. When results are displayed, changes can be made by selecting another display type from the toolbar, such as display by size, resolution, bits per pixel, or by status.
4. Right-click the image and select **Change Resolution** to convert it.

3.3 Standard Classification

Standard classification is image-based and one of the fastest classification methods. Standard templates are best for structured forms where information is found in predictable locations.

When creating standard templates in the recognition project, you have an option to get them from the base of images automatically, or to create them manually. The option you choose depends on the number of documents that are available for automatic analysis. If an insufficient number of documents are available, or if a custom template is needed, choose to create standard templates manually.

After the project has been run in testing or production, you can update the project templates to accommodate images that were not classified to exiting templates.

3.3.1 Creating Standard Templates Automatically

The **Project Update Wizard** enables creation of new standard templates in an existing project. This tool analyzes a set of unsorted images and automatically creates classification templates.

To create standard templates:

1. Select **File** > **Update project**.
2. Select **Start**.
3. In the **Select a Base of Images** window, select images using the toolbar buttons:
   -  Loads selected images from a directory.
   -  Loads all the images from a directory.
   -  Loads all the images from a directory and subdirectories.

- Unloads the selected images.
- Unloads all the images.

Note: The images you use must be at the same resolution as the project.

4. Select **Update**. The wizard automatically creates the templates, and the **Classification View** is refreshed. Just before the project update process starts, **Image Analyzer** runs automatically.

3.3.2 Recommendations for Automatic Learning

Usually, running one to five iterations of automatic learning is recommended, depending on the volume of images. Deciding to stop learning iterations depends on the results of the tests as described in this topic. When using automatic learning, it may be necessary to run only one iteration and then allow Production Auto-Learning to create templates during production.

Steps for efficient learning iterations:

1. Open the project in Recognition Designer.
2. Select a base of images that contains 5,000 to 30,000 images from the learning image base. The maximum number of images the Update Project Wizard can process in each iteration is 40,000 images. An image base with more than 100,000 images requires one to four learning iterations.
3. Start learning. Click **Learn**.

   Note: Learning takes approximately 20 minutes for 5,000 images and 2 to 3 hours for 20,000 images on a single 2.3 GHz *CPU* machine with 2 *GB RAM*.

4. When learning completes, click **End** to view the list of the newly created templates in the template list (to the left of the Recognition Designer main window).
5. Save the project.
6. Select all the templates in the template list and give them the same template code (for example “TEST”). At this stage of the learning phase, it is too early to specify the templates codes. This will come later when fine tuning the templates. In addition, compilation runs faster if all templates have the same code.
7. Review each template against its image base to detect and delete wrong templates. As a general rule, the more templates in the project, the more wrong templates should be eliminated at this step. Wrong templates are the result of over-classification (two or more templates created for one document class).

A fast option to eliminate wrong templates is to delete templates that have very few images in their image base. The following are general indications; they may not apply to all projects:

- Eliminate all templates with only 2 images in their image base if the project has more than 300 templates.
- Delete all the templates with 4 images in their image base if the project has more than 1,000 templates.
- Delete all the templates with 5 images in their image base if the project has more than 2,000 or 1,000 templates in some cases.

The appropriate way to handle wrong templates created by over-classification is to merge them or give them the same template code but neither of these solutions is recommended at this early stage. Instead, it is recommended to wait until the last learning iteration before fine tuning templates by merging them or using template codes.

8. Compile the project. Select **Classification** > **Compile**. Compiling takes approximately 15 minutes for 1,000 templates that have the same code and up to 24 hours for 5,000 templates that have different codes (for example, 10 different codes). These durations are given for indications only.

9. Select **Test** > **Classification Test**. The classification test takes approximately 10 minutes for 5,000 images. Run the test keeping the images that are automatically selected. These images are the sample images used to create the templates. They are stored in the template folders.

   This test is a sanity check to ensure automatic learning went well. The classification rate is usually 100%. If the classification rate happens to be 99% or 98%, this may be due to an image that slightly differs from the other images in the image base of the template and on which the internal anchors, placed automatically by automatic learning, do not exactly match the internal anchors on the other images. Having a classification inferior but close to 100% is not a problem and does not require resolution.

10. Export the images listed on the **To confirm** and **Not classified** tabs. Select **File** > **Export the following images**. Add these images to the set of learning images so that they are included in the next learning iteration.

11. Select **Test** > **Classification Test**. Run the test selecting the images from the evaluation base, or for a quicker test, use the test base. Check the classification rate to decide whether another learning iteration is needed. To make this decision, use the following values as a guide:

    - If less than 50% of documents are classified, consider running other learning iterations until 75% or more of the documents are classified. This is likely to happen if the project mostly contains semi-structured or unstructured documents.
    - If more than 90% of documents are classified, consider running one more iteration. This is likely to happen if the project mostly contains structured

documents. If more than 98% of documents are classified, you may decide to stop the learning phase at this stage; this is likely to happen if the project has a limited number of structured forms only.

Track the results of the classification test in a spreadsheet: Derive a curve to track the percentage of classified document versus the number of templates in the project. An asymptotic curve trend means that the curve turns flat. This means that creating more templates, through more learning iterations, cannot improve classification any more. When the curve trend is asymptotic, consider fine tuning the templates as described in the section "Tuning Standard Templates" on page 64.

12. To perform an additional learning iteration, select **File** > **Update Project**. The Project Update Wizard enables running the automatic learning feature on an existing project.

13. Select a base of images that contains 5,000 to 30,000 images from the learning image base and ensure the selected images contain the unclassified images exported from the **To confirm** and **Not classified** tabs during the classification test.

14. Start a new learning iteration: Select new images from the learning image base and perform the steps described earlier in this procedure.

15. After the last learning iteration, run a classification test from the evaluation image base. Use the evaluation base to fine tune and test the templates. Learn to fine tune the templates in the section "Tuning Standard Templates" on page 64.

3.3.3 Creating Standard Templates Manually

The **New Template Wizard** enables creation of one standard template at a time. Use this option when image sets are gathered and sorted and you know which templates to create.

To create a template:

1. Click **Classification View**, right-click, and select **New template** from the templates list.

2. In the **New Template Wizard**, click **Start**.

3. In the **Select a Base of Images** pane, select the images to use. A minimum of ten images is recommended, but fewer are allowed. When using fewer than ten images, you must confirm creation of template creation with the available images.

4. Click **Create template** to proceed to template creation. **Image Analyzer** runs automatically. Alternatively, click  to run the **Image Analyzer** to check the image format and resolution.

5. At the **Setting the invariant zones** step, select **Setting zones** to define invariant zones that constrain Recognition Designer to the zone you specified. In most cases, the system automatically defines those zones however.

   - In the **Define Invariant Zones** window, click to delineate the zones which are typical of the template. Those graphical zones must be wide enough to cover all the invariant zones of the document. If graphical zones are too small, the template cannot be created and a warning message appears.

6. Click **Create template** to proceed to template creation.

7. Select **New template** to keep the wizard open and create another template or select **End** to close the wizard. Click **Cancel** to close the **New Template Wizard** without adding the template to the project.

3.3.4 Creating Templates Based on Template Codes

The **Advanced Learning Wizard** creates new templates from an annotated image base, that is, a base in which the template code is known for each image. Images must be representative of the production flow. To produce high quality templates, a minimum of 10 images, and preferably 20 images, should be available for each template in the image base.

To create templates based on template codes:

1. Select **File** > **Advanced Learning**.

2. Select **Start**.

3. Create a new annotated image base or open an existing one.

   - To create an annotated image base, use the toolbar to select images and create a subdirectory by document code. An image document code is the name of the directory in which it is located. Class the images in their respective subdirectories using Windows Explorer.
   - To open an existing annotated image base, use the toolbar to select images individually, by directory or by tree. Use **Image Analyzer** to control image format and resolution. The number of selected images displays at the top left and the image list displays grouped by document code. The group line indicates the document code followed by the number of corresponding images. The selected image is displayed in the right pane of the wizard. Select several files at the same time and all or a part of the selection can be deleted.

4. Select **Learn** to start the automatic advanced learning. During the automatic learning process, information on the templates is displayed as they are being created. If the selected images are not homogeneous, a warning message appears enabling you to define a common resolution for all the images in the base. Select **Cancel** to stop the process at any time.

   After the templates have been created, the window displays the number of created templates. If the project already contains templates named “A (x)”, the

numbering begins at "A (Max(x)+1)". Multiple selections are allowed. Delete templates by selecting them and clicking  at the top right of the window.

5. At the bottom of the window select **Create a new directory for each document code** to create a logical directory per group of templates associated with the same code. If this option is selected, for each document code of the generated templates, a logical directory is added to the current directory. If this directory exists, the templates are placed in it. If this option is not selected, the new templates are placed in the current logical directory. The new templates are automatically selected to enable the user to move them easily to another directory.
6. Select **Next** to go to the last step. The final screen appears when the learning has finished.
   a. Select **End** to add the templates to the project.
   b. Select **Cancel** to close the **Advanced Learning Wizard** without adding the templates to the project.
   c. Select **Previous** to return to the previous window in the wizard.

3.3.5 Merging Standard Templates

Merging templates can be useful when Recognition Designer has automatically created numerous templates from a base of images when only one or two templates are needed. When automatic learning occurs in the **Project Update Wizard**, or during production auto-learning, distinct but graphically similar templates may be automatically created. These templates are good candidates for merging. Only standard templates can be merged. Merging is not compatible with *HPA* or generic templates.

 **Note:** An alternative to merging two templates is to assign all the similar templates the same template code.

To merge templates:

1. Click the **Classification View** button on the toolbar.
2. Select the templates to merge from the **Template list**.
3. Right click the **Template list** and select **Merge templates**. If the message `Cannot merge templates` appears, the templates are not similar enough to be merged. The newly created template assumes the following settings:
   - **Field settings**: Field positions and settings are kept as defined in the first template with an associated index family. For example, if Template 1 has no associated index family and Template 2 has an associated index family, field settings defined in the index family associated with Template 2 are kept for the newly created template.
   - **Template name**: The new template takes the name of the template with an associated index family among the merged templates, assuming that only

one template has an associated family. If no index family is associated, or if different families are associated, the name of the first selected template is kept.

- **Index family**: See the examples in the following table to understand how index family settings are applied during a merge:
  - If all the merged templates have the same index family, it is assigned automatically to the new template (Example 1).
  - If only one of the templates merged has an index family or if several templates have the same index family, it is assigned automatically to the new template (Example 2).
  - If the merged templates have different index families, no index family is assigned to the new template (Example 3).

Table 3-3: Index family Assigned to a Template Created by Merging Multiple Templates

Merged templates	Example 1	Example 2	Example 3
	Assigned family	Assigned family	Assigned family
Template 1	Family a	No family	Family a
Template 2	Family a	Family a	Family b
Template 3	Family a	No family	Family a
Template 4	Family a	Family a	Family b
Newly created template	Family a	Family a	No family

Related Topics

"Template Properties" on page 295

3.3.6 Converting a Standard Template to an HPA Template

Converting a standard template to *HPA* is a way of reducing classification conflicts. Learn more in the recommendations on tuning templates, in the topic "Reducing the Number of Standard Templates by Converting Them to HPA" on page 70.

To convert a standard template to an HPA template:

1. Select the name of the standard template in the **Classification View**.
2. Right-click and select **Convert to HPA Template**. A confirmation message is displayed.

3.3.7 Tuning Standard Templates

This section suggests recommendations for fine tuning the standard templates created automatically. Recommendations depend on the business requirement and the nature of documents.

Business Requirement

As a general rule, if the requirement is classification only, use standard templates and apply template codes. If the requirement is classification plus data extraction, use *HPA* templates. This is described in the section "Achieving Business Logic with Template Codes" on page 65.

Nature of Documents

Suggestions for fine tuning depends on the nature of the document. These suggestions are summarized in the following table and described later on.

- Solve over- and under-classification by merging templates and converting the templates to HPA as described in "Solving Over-Classification and Under-Classification" on page 66.
- Consider keyword classification as a backup solution for graphic templates that fail as described in "Keyword Classification Uses" on page 88.

Table 3-4: Recommended Fine-tuning Depending on the Nature of Documents

Nature of documents	Recommended fine-tuning
Very structured forms	• Merge templates • Convert to HPA • Apply template codes • Keyword classification as backup
Structured forms with attachments. Different versions of document classes or multiple-page documents	• Merge templates • Convert to HPA • Apply template codes • Keyword classification as backup
Mailroom with forms	Create a generic template and use keyword classification or Text Matching
Mailroom with semi-structured document classes, attachments, multiple-page documents and different versions of document classes	Create a generic template and use keyword classification or Text Matching
Mailroom with mail and correspondence	Create a generic template and use keyword classification or Text Matching

3.3.7.1 Achieving Business Logic with Template Codes

Templates created automatically reflect the graphical characteristic of images but does not necessarily reflect any business logic. Documents that are graphically similar may be required to be classified with separate templates in a specific business logic. In this case, applying template codes enables matching classification to the business logic. Also, learn how template codes behave in the classification engine in the section "Solving Conflicts with Template Codes" on page 67.

Using the example of checks documents, here are suggestions for setting template codes and achieving various business logic for processing checks. As a very first step, process all checks with automatic learning to create one graphic template for each check. Two business requirements are considered: classification only and classification plus data extraction. For the different requirements, these are suggestions to use when applying the template code:

Classification Only

- Specify the same code to all the templates if the business logic is to identify that a document is a check, irrespective of the bank. In this case, even if the classification engine has a doubt (meaning that several potential templates are in conflicts), it does not send the document to Identification for manual classification.
- Specify a specific code to each bank if the business logic is to differentiate banks. In this case a check needs to be classified to the correct bank. If the engine has a doubt, the document will be sent to Identification to be manually classified.

Classification and Data Extraction

- Specify the same code to all the templates if the logic is to extract the same data across banks. In this case, only one indexing family is created for all fields and it is applied to all the different "check" templates. Zonal recognition is possible by placing the fields accurately on each template.
- Specify a specific template code to each template if the logic is to extract different data items for each bank. In this case, one indexing family is required for each bank and it is essential that each check is classified to the specific template that is associated with the appropriate indexing family. Zonal recognition is possible by placing the fields accurately on each template.

Note: If checks look similar because several banks use the same graphical layout to print their checks, use *HPA* templates to differentiate between banks and provide each template a specific template code so that the appropriate indexing family is applied.

3.3.7.2 Solving Over-Classification and Under-Classification

Over-classification occurs when too many templates are created and under-classification occurs when too few templates are created. Both over-classification and under-classification can occur when documents are similar graphically and automatic learning cannot place graphic anchors precisely enough.

When several templates are created for very similar documents (over-classification), convert standard templates to an *HPA* template and place anchors precisely. By positioning anchors precisely, HPA enables an accurate differentiation between graphically similar documents. Example 3-1, "Solving over-classification" on page 66 explains how HPA can solve over-classification, while Example 3-2, "Solving under-classification" on page 66 explains how HPA can solve under-classification and meet the business logic.

Example 3-1: Solving over-classification

One example is with a form for which scanned images exist with different qualities (printed and fax), the difference in quality may cause the images to be graphically different and this generates over-classification. In this case, using HPA is recommended. Another example is with semi-structured documents such as bank details which may be different graphically but need to be processed with the same data extraction settings (same index families to extract the same data). In this case again, automatic learning may create different templates which do not fit the business requirement. Again the solution is to create an HPA template and place HPA anchors on areas that are similar on the different bank details so that all bank details are classified to the same template and proper data extraction can be applied.

Example 3-2: Solving under-classification

One example is when several vendors use layouts that are graphically similar and that contain the same information but the information items are positioned at slightly different places. In this case, automatic learning will create only one template but this does not enable correct data extraction because items of information to be extracted cannot be found as they are positioned slightly differently depending on the vendors. In this case, create HPA templates to accurately differentiate the images and place the fields accurately for data extraction. Learn to place efficient HPA anchors in the section "Recommendations for Anchors Position and Settings" on page 76.

3.3.7.3 Solving Classification Conflicts

A conflict occurs when classification returns several results (several templates) representing more than one document class for a given image. In other words, the image potentially matches several templates with different codes. Solving conflicts involves helping the engine to make the right decision between the several candidates or making the templates more discriminative. This is possible by using template codes and by converting some standard templates to *HPA* templates. Basically, understand that:

- If conflicts come from over-classification, you will probably use template codes to solve this type of conflicts.
- If conflicts come from two templates potentially matching an image (because the templates are not discriminant enough), you will probably convert both templates to HPA and place anchors on discriminant areas.

Finally, another solution is to tune the standard pre-classification and decision values. This solution is explained later although it is usually not necessary in most projects.

3.3.7.4 Solving Conflicts with Template Codes

There are two levels for identifying a template: by specifying a unique template NAME and by specifying a template CODE. The template name is mandatory. Consider that it is mainly for administration purposes as it is not taken into account by the classification engine. The template code is optional but it is essential as the classification engine relies on the template code to deal with conflicts and make a final decision.

It is a best practice to use template codes to reduce the number of conflicts in Classification. Learn also to use template codes to fit the business logic in the section "Achieving Business Logic with Template Codes" on page 65.

Understanding Template Codes and Conflicts

- If an image matches two templates, and both templates have the same template code: there is no conflict and the image is classified to the template having the best decision rate above the decision threshold. Learn more about decision rates in the section "Tuning Standard Pre-classification and Decision Rates" on page 68.
- If an image matches two templates, each having a different template code: there is a conflict and the image is routed to Identification where an operator manually selects the correct template.

Example of How Template Codes Act on Conflicts

Image1 potentially matches five templates (t1, t2, t3, t4, t5). First, the classification engine isolates templates whose decision rate is higher than the decision threshold. In this example, these are t1, t2 and t3. Secondly, the engine considers the template codes:

- t1, t2, t3 have the same template code: there is NO conflict and Image1 is classified to the best template which is the one with the higher decision rate.
- t1 and t2 have the same template code but t3 has a different template code: there is a conflict and Image1 is sent to Identification.
- t1, t2 and t3 have a different template code: there is a conflict and Image1 is sent to Identification.

3.3.7.5 Using HPA Templates to Solve Conflicts and Detect False Positives

This is covered in the section "Solving Conflicts and Detecting False Positives" on page 75.

3.3.7.6 Tuning Standard Pre-classification and Decision Rates

This section provides recommendations on testing and adjusting both the pre-classification and decision threshold values to reduce the number of conflicts and speed up graphical classification.

3.3.7.7 Understanding Pre-Classification and Decision Rates

Classification is carried out in two steps for graphic templates, that is standard and *HPA* templates.

- As a first step, a pre-classification phase is carried out to retain a subset of best candidate templates. This step is very fast as only 1/128th of the image pixels are computed. This is possible by processing the images in memory and applying an automatic reduction of the number of pixels on the images.
- As a second step, decision rates are calculated for the best candidate templates. This step takes longer than pre-classification as the decision is computed on a full-scale image (that is without reduction of the number of pixels), on a pixel per pixel level for each automatic anchor (for the top five anchors). For standard templates, the anchors are placed automatically; these anchors cannot be modified (neither their number, nor their position). For HPA templates, project designer can place as many HPA anchors as required and fine tune them.

The pre-classification and decision threshold values are adjustable for both standard and HPA templates.

- In standard templates, the pre-classification threshold value and the decision threshold value apply to all the standard templates of the project. It is strongly recommended to first test the default values of the pre-classification and decision threshold as they are usually suitable for most projects.
- In HPA templates, a different pre-classification rate is set for each HPA template and a specific decision threshold rate is set for each HPA anchor.

The pre-classification algorithm takes a few microseconds to process a single template. For this reason, the number of templates in the project does not

significantly impact the classification processing time. The decision algorithm relies on the analysis of the automatic or HPA anchors, and therefore is more time consuming depending on the number of anchors and the searching zones defined for each of them. In conclusion, to process one candidate template, it is the number of candidate templates that impacts the processing time. For this reason, reducing the number of candidate templates is a solution to significantly speed up classification. To do so, set a higher pre-classification threshold value to reduce the number of candidate templates. With this solution there is a risk of reducing the number of classified documents (since fewer candidates are analyzed by the decision algorithm).

3.3.7.8 Tuning Pre-classification Rate for Standard Templates

In most projects, it is not necessary nor recommended to try to tune the pre-classification rate. The pre-classification algorithm is reliable and changing it may generate more conflicts or false positives. However, in some specific projects, you may want to try tuning this value which is why we are providing indications here. Tuning involves analyzing conflicts closely as well as the documents from the **To Confirm** tab when running a classification test.

To tune the pre-classification threshold value:

1. Run a classification test keeping the default pre-classification rate. Select a set of images from the evaluation base (select **Test** > **Classification Test**)
2. When the test results appear, select **Display** > **Advanced Information**. The **Pre-classification** and **Decision** columns appear. They indicate the pre-classification and decision rates.
3. Click the **Not Classified** tab.
4. Click the heading of the **Pre-classification** column to sort per increasing or decreasing pre-classification rates.
5. Detect the documents that have the pre-classification rate close to the pre-classification threshold, so for example, between 66.6 and 69.9.
6. Set a lower pre-classification rate threshold but not lower than 65 or this would generate too many conflicts. To adjust the pre-classification rate, select **File** > **Project Options**. Click the **Classification** tab and click the **Advanced engine parameters** button.
7. Perform the classification test again and verify whether the documents you had detected during the first test are now passing pre-classification. If they do, you should see them in the **Classified** tab or in the **To Confirm** tab.

3.3.7.9 Tuning Decision Rate for Standard Templates

This topic explains how to test classification and tune the decision threshold value accordingly.

To tune the decision threshold value:

1. Run a classification test keeping the default decision rate (80%). Select a set of images from the evaluation base (select **Test** > **Classification Test**).
2. When the test results are displayed, select **Display** > **Advanced Information** to display the pre-classification and decision rates (**Pre-classification** and **Decision** columns).
3. Click the **Classified** tab and detect the documents that are classified to the wrong templates and with a decision rate close to the threshold value, which means a value close to 66. For those documents, set a higher threshold value for the decision rate and run a classification test again. To adjust the pre-classification rate, select **File** > **Project Options**. Click the **Classification** tab and click the **Advanced engine parameters** button.
4. Run the classification test again. Click the **To Confirm** tab and examine the conflicts. For the documents that have conflicts, examine the best candidate template. The best candidate is the one with a decision rate that is closer to the threshold value, that is between 60 and 80. Disregard candidates with a rate lower than 60. For the best candidate template, set a lower decision rate and run a classification test again.

Note: Select a template candidate and then select **Display** > **Template viewer** to display an image pane with the template reference image.

3.3.7.10 Reducing the Number of Standard Templates by Converting Them to HPA

This topic offers recommendations for selecting standard templates that can be converted to *HPA* templates to reduce the number of templates. The aim is to reduce the amount of work involved in placing fields for data extraction on templates. As a general rule, focus first on documents that exist in different versions (graphical variations) and on which data extraction is made with the same index family. Typically, these documents are addressed by different templates (one for each graphical variation) and have the same template code (same index family).

To select the standard templates to convert:

1. Focus on the “top” templates that have the same template code. As an indication only, consider that top templates are the standard templates with more than 50 images in their reference image base. The aim is to identify series of templates having the same code. For example, a series of 10 top templates having the same code means that there are 10 variations of the same template. To identify those templates, one solution is to sort the templates by template

code in the **Classification View**. Another solution is to select **Project Options > General > Advanced Information** and list the templates by codes.

2. Convert one of the 10 variations to an HPA template.

3. Place the anchors. Learn how in the section "Recommendations for Anchors Position and Settings" on page 76.

4. Review the other templates to identify other graphic variations with the same code and convert them to HPA as necessary. Some series of template variations can eventually be replaced possibly by only one HPA template or several but always fewer than the initial variations.

5. Test the converted HPA template:

   a. In the **HPA Template Editor**, select **All** as **Minimum hit number**.

   b. Run the test on the image base of standard template converted to HPA.

   c. Ensure to obtain 100% classification.

6. If necessary, tune the anchors positions and sizes until reaching 100% classification. If necessary, try lowering the pre-classification threshold down to 60% on a specific anchor. Learn more in the section "Tuning HPA Templates" on page 75.

3.4 HPA Classification

HPA stands in for High Precision Anchors. As a general rule, consider that HPA is best to accurately differentiate between documents that have similar graphic layouts and that could result in false positives with standard templates. Understand that contrary to standard templates in which the graphic anchors are positioned by the automatic learning (and cannot be tuned), HPA enables positioning and tuning anchors precisely. Such a precision in anchor positions makes HPA a very flexible method to differentiate documents or to not differentiate them. For documents that are similar graphically, place an anchor on a similar area to NOT differentiate them or place an anchor on a discriminative area to differentiate them. This is how HPA can solve over-classification or under-classification occurring with standard templates.

Since HPA enables placing anchors manually, it does not require any image base. Only one image is sufficient to create an HPA template. In terms of processing speed, classification is as fast with HPA templates as with standard templates when default settings of the decision and pre-classification threshold values are kept. In terms of fine tuning, HPA templates are more flexible as the pre-classification threshold value is adjustable for each template and decision rate is adjustable for each HPA anchor. These two parameters can dramatically influence the processing speed. The higher the pre-classification threshold, the faster the processing time. The anchors also influence the processing speed: the fewer the anchors, and the smaller the searching areas, the faster the processing speed.

3.4.1 Creating HPA Templates

HPA templates are created manually by placing anchors on a template page and assigning settings to each anchor. HPA templates are displayed on a green background in the template list displayed in **Classification View**.

To create an HPA template:

1. Click **Classification View** on the toolbar.
2. Right-click the template list and select **New HPA Template** from the contextual menu.
3. Select a reference image that will be the new HPA template. The requirements to the image include:
   - The reference image format must be supported. Valid image formats are listed in the topic "Advanced Recognition Supported Image Formats" on page 426.
   - The reference image must have the same resolution as the project. To view the project resolution, view the **General** tab in the **Project Options** window.
4. Click **Open**. The template is created and added to the list of templates in Recognition Designer. The template opens for editing in HPA Template Editor.

Related Topics:

"Editing HPA Templates" on page 72

3.4.2 Editing HPA Templates

This section describes the settings of *HPA* templates. Read this section to understand the main settings, then learn to optimize the settings in the section "Tuning HPA Templates" on page 75 which includes recommendations for anchors position and settings, for reverse anchors and for adjusting the threshold values.

To edit an HPA template:

1. Click **Classification View** on the toolbar.
2. Select the HPA template to edit and right-click.
3. Select **Edit HPA Template**. The HPA Template Editor appears displaying the selected template.
4. Click  to draw an anchor on the image. Select an anchor that is typical of the template, and therefore a good discriminator. Two to three well-defined anchors with appropriate settings should be sufficient.
5. Set the **pre-classification threshold** for each anchor. This threshold, ranging from 0% to 100%, is the matching rate required between a document and the

HPA template for the document to be classified as a potential match. A high pre-classification threshold is recommended and is set to 70% by default.

6. Set the following parameters for each high precision anchor:

   - Set a **Minimum hit number** to define the conditions required for the document to match the HPA template. This value specifies how closely anchors must match the template anchors for the document to be classified. Set this value to require that all anchors match, or to allow documents to be classified when fewer anchors match. If you want a specific anchor to always be included in the **Minimum hit number**, then set it as **Mandatory**.
   - Set an **Anchor size**. Small anchors are recommended. Anchor size must not exceed 1 x 1 cm. If the size exceeds the recommended size, warning messages are displayed at the bottom of the **HPA Template Editor**.
   - Set a **Search zone** to define the boundary zone for searching the anchor. The default value is set in the **Project Options**, in the **Classification** tab. A small search zone is recommended, with a minimum value of 30 x 30mm is recommended.
   - Select **Reverse answer** so the anchor will be considered valid when the matching rate is inferior to the anchoring threshold. Learn from an example in the section "Recommendations for Reverse Anchors" on page 77.
   - When you select **Strengthen** the anchor is thickened before searching begins. This is useful to facilitate searching when the anchors are printed in thin characters.
   - **Mandatory**: For a document to be classified, the template anchor must match the anchor in the document. Mandatory anchors are searched for first. Furthermore, mandatory anchors are always included as one of the minimum number of anchors required for classification (as specified in **Minimum hit number**). You could use mandatory anchors to differentiate between multiple templates that are very similar. For example, if two invoice forms have very similar layouts and columns for purchased items, but the company names are different, then you could use each name as a mandatory anchor for each template.
   - Select the **Binary conversion threshold** option to setting parameters for a color document that it is not displayed in the interface. Enable this option to adjust the binary conversion threshold manually, or clear this option to enable automatic adjustment. Preferably, select an anchor that is highly contrasted, but if the automatic adjustment gives no satisfactory results check the **Binary conversion threshold** and adjust the value until the anchor image contrast is correct.
   - The **Anchoring threshold** defines the matching rate that is required between the anchor and the processed image pattern. The default value is set to 70%.

3.4.3 Testing HPA Templates

This section describes the main steps of the test process and how to use the testing interface. Learn to tune settings in the section "Tuning HPA Templates" on page 75.

To run a unit test on the *HPA* template:

1. Select **Classification View** in the main toolbar.
2. Select the template in the **Template** list and right-click.
3. Select **Edit HPA Template**.
4. In the **HPA Template Editor**, select **Test** > **Evaluation on Image Base**. The images from the template image base are loaded automatically.
5. Select an image to run the analysis on. Alternatively, run the analysis on all the images using the **Test** menu or by pressing **F9**. Results are displayed in the left pane. For more information on test window menu and options, see "HPA Template Test" on page 355.

To visualize the anchors found in each image:

1. Select a document from the **File** list.
2. Select an anchor from the **Show detail for list** box.

Note: Scanning double-sided documents can produce inverted images.

To change the template reference image:

1. Select an image from the list of images in the test base.
2. Right-click the image name.
3. Select **Use as Referenced Image** in the menu.

To export images, there are two export options:

1. Select an image from the list of images in the test base.
2. Right-click the image name.
3. Select **Export Images** in the menu:
   - **Export Images** > **In the Template Bank** copies the selected images to the template image base (the images are copied without any confirmation message).
   - **Export Images** > **In a Folder** copies the selected images to an existing or a new folder.

3.4.4 Tuning HPA Templates

The objective for testing and tuning of *HPA* templates is to optimize the template and to solve conflicts and detect false positives. Ultimately, it is best practice to export the images that cannot be classified with HPA and consider processing them with keyword classification.

3.4.4.1 Solving Conflicts and Detecting False Positives

This section suggests a testing process and tuning of pre-classification and anchoring thresholds and of other parameters to solve conflicts and false positives.

To solve conflicts and detect false positives:

1. In the **HPA Template Editor**, test individual templates. Set the **pre-classification threshold** to 75% (default is 70%).
2. Tune the values of the **Pre-classification threshold** and **Anchoring threshold**. The pre-classification rate is calculated for each anchor individually. The decision rate (anchoring threshold) is the average of all the anchors on the template. Recommendations are:
   - **Pre-classification threshold**:
     - Always run a first test with the default value (70%).
     - To solve conflicts with documents that are graphically similar, increase the value to 75%.
     - To solve conflicts with documents that exist in various versions (one template per version), lower the threshold to 65% if there are fewer than 50 variations and down to 60% if there are fewer than 10 variations. However a better solution is to add more *HPA* anchors.
   - **Anchoring threshold**:
     - In most cases, keep the 70% default value.
     - Always test anchors individually before tuning them.
     - Apply minor adjustment: 67% to 73%.
     - Consider lowering the decision threshold to 68% if the image is degraded (lack of black pixels). Placing an anchor on a degraded area of the image is not recommended. However, if the entire image is degraded or the best discriminative anchor is in a degraded area, try lowering the anchoring threshold.
3. Set the **Minimum hit number** to a value that represents two thirds of the anchors. For example, if there are three anchors, set this value to two.
4. Select images of the current template AND images of other documents. This is essential to detect false positives. Consider using the learning base that served for automatic learning.

5. Run the test and look for false positives. If there are false positives, make the template more discriminative. To do so, add a new anchor or adjust the **Minimum hit number** to a value that represents all the anchors implying that all the anchors must hit for the document to be classified.
6. Run the test again on the learning base. When there are no false positives left, continue tuning the anchors until reaching 100% of classified documents.
7. After testing individual templates, close the **HPA Template Editor**. Select **Test** > **Classification Test**, run a test on all HPA templates on the images of the evaluation base (same evaluation base as the one used to create and tune the standard templates).
8. Export the images that are not classified into a sub-folder. For these images, consider using keyword classification.

3.4.4.2 Recommendations for Anchors Position and Settings

To position efficient anchors on an *HPA* template, position two to four anchors to discriminate the document. These anchors prevent false positives and conflicts. Conflicts occur when several templates can match a document because of graphical similarities. False positives occur when a document is classified to a wrong template because anchors are placed on areas that are not discriminant enough or there are not enough anchors. If there are no false positives, consider setting only two anchors to match instead of three. In the **HPA Template Editor**, set the **Minimum hit number** so that two thirds of the anchors must match.

For all anchors:

- Keep anchors size always lower than 10 mm; usually 2 to 6 mm. The smaller, the better.
- Place anchors away from the document edges and away from handwritten areas.
- Place anchors on patterns that may appear several times within the search area.
- Place anchors on small portions of text or graphic elements such as logos:
  - Keep a 50%/50% balance between black and white pixels.
  - For anchors placed on a graphic, select an element that does not vary from one image to another whatever the image quality.
  - For anchors placed on text, select small fonts with light characters. Select a small portion of a word and do not select the outside of the letters as illustrated in the following screenshot.

The following screenshot illustrates some of the recommendations for anchor position and settings. It contains 3 text anchors, placed away from the document edge. On this example, the **Minimum hit number** is set to 2, which means that two thirds of the anchor must match; this is sufficient if there are no false positives. The **pre-classification threshold** is set to 75% which is usually recommended for initial testing. For information about learning to tune this threshold, see "Tuning HPA Templates" on page 75.

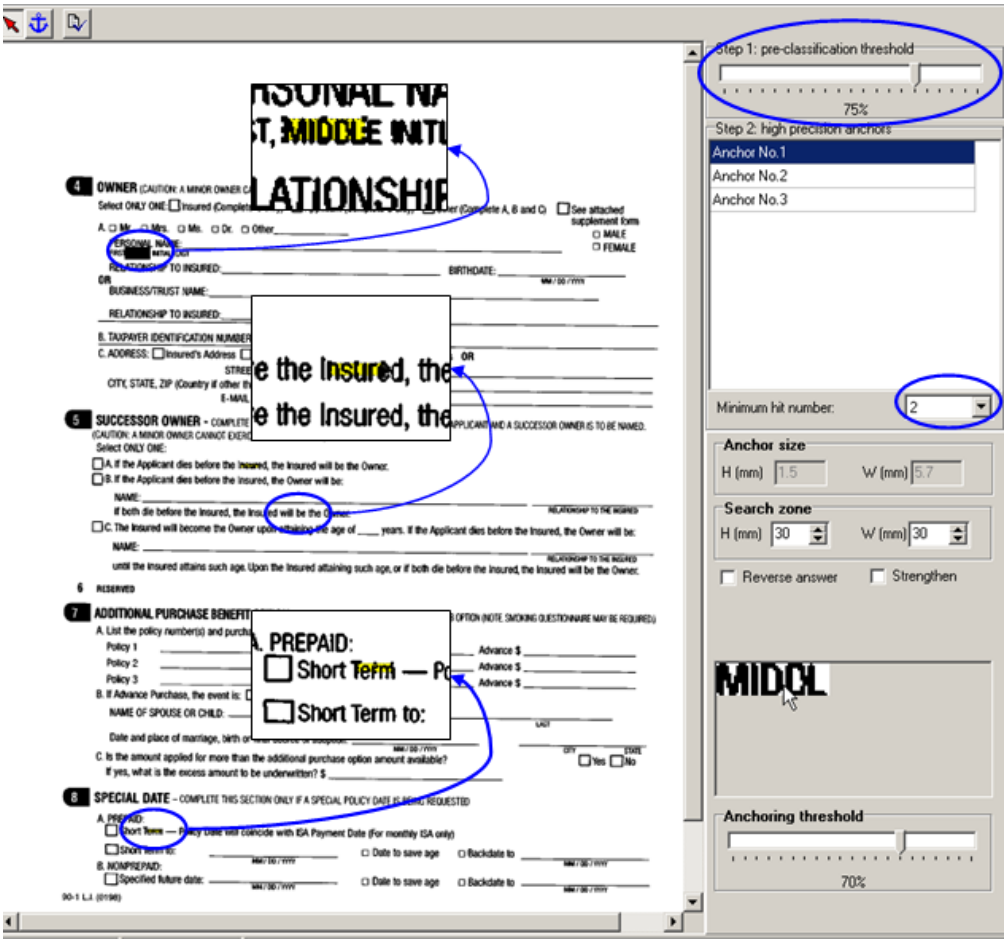


Figure 3-1: Recommended anchor settings

3.4.4.3 Recommendations for Reverse Anchors

A reverse anchor comes in addition to the other anchors when it is necessary to differentiate between documents that are 75% to 95% similar. The idea is to place a reverse anchor on a graphic or text pattern that is present on one template but that must NOT be present on another very similar template. In the **HPA Template Editor**, select **Reverse answer** to set an anchor as reverse anchor. On the template image, the reverse anchor displays as a red square.

Examples of Using Reverse Anchors

The reasons for differentiating very similar templates depend on the business requirements. Here are two examples illustrating different data extraction requirements:

- A similar life-insurance form is used by two insurance companies. One company uses the basic form. Another company uses the same basic form but with an ID

in the upper right corner. The template “Life_Ins_EF7” has the ID “EF783887” in the upper right corner while the template “Life_Ins_Other” is the basic form without ID. The business requirement is to extract data from both templates but NOT the same data. In this case, the only solution to differentiate the two templates is to create two *HPA* templates, one with the ID (“Life_Ins_EF7”) and one without the ID (“Life_Ins_Other”). To do this:

1. In the **HPA Template Editor**, place a normal anchor on the ID “EF7” on the template that has the ID. The anchor must be placed only on “EF7” and not on the whole ID as the ID may change. In this example, we assume that “EF7” is the invariant element of the ID.
2. On the template that does not have the ID, use the same image, that is the one that has the ID on it, and place a reverse anchor on the ID “EF7”.
3. In the **Index View**, select the template that does not have the ID on it (“Life_Ins_Other”), right-click and select the option **Change the Index Image** and then select the image that does not have the ID on it. It is important to use a different image in the **Index View** and in the **HPA Template Editor** because the reverse anchor functionality is based on the fact that an element must not be present.
4. Create two index families and associate each HPA template with its respective family.
5. Specify each template with a specific template code. Set the **Minimum hit number** so all anchors must match. Or add other reverse anchors (which is not possible in this example).

- A similar template is used by two vendors. The business requirement is to extract the same data on the template irrespective of the vendor. However, one vendor sends printed documents, while the other vendor sends documents by fax. Fax images usually have graphical elements, such as lines, that make them different from the original printed document. To differentiate the two types of images:

  1. Create one HPA template for the printed images and one for the fax images. For both templates, use the fax image as reference image in the **HPA Template Editor**.
  2. On the HPA template for the fax image, place a reverse anchor on an invariant graphic element of the image such as a line created by the fax; this element is not present on the printed images. During classification if the line is not found on the image, the document is classified with the template that is created for the printed document and if the line is found, it is classified with the template created for the fax images.
  3. Finally, in the **Index View**, select the HPA template for the printed image and change its index image: replace the fax image by the image of the printed document. Associate both HPA templates to the same index family since the same data is extracted on both templates. Both templates can have the same template code.

3.5 Text Matching Classification

Text matching classification is mostly directed at semi-structured and unstructured documents, specifically, at documents that include large text strings indicative of a particular document class. The predefined blocks of text are searched in images after full page recognition.

3.5.1 Understanding Text Matching Classification

Text matching uses a learning mechanism to create templates based on the textual content of the documents. Text matching learns a text *signature* for a template. A signature is a set of large text strings that is representative of the template. Compared to keyword classification, text matching uses long text strings whereas keyword classification relies on keywords. This is one of the reasons why text matching is best for documents that contain mostly text and for which keyword classification may cause too many false positives.

During text matching classification, the full page *OCR* engine reads the data from each image and text matching tries to match the document textual content to the signatures of the templates. If a match is found, the image is associated to the template associated with the corresponding text matching signature. To associate a signature with its template, text matching relies on a folder structure which you have to prepare before launching the signature learning.

Note: Unlike the automatic learning feature that is available for graphical classification templates, text matching does not classify or categorize the templates automatically. This is why you have to create the folders with the correct images in them. Text matching assumes that each image in a folder contains the text signature that you want to associate with a template. Creating the image base is a very important step for successful learning. Learn recommendations in the section "Preparing the Image Base" on page 82.

3.5.1.1 Text Matching Classification Uses

Text matching classification is mostly directed at semi-structured and unstructured documents. When choosing between keyword classification and text matching classification, consider the following:

Cases Suitable for Text Matching

- Documents with a large proportion of recurrent textual content (wording) such as in legal documents.
- Semi-structured or unstructured documents having 60% or more of recurrent wording across all the documents of a given document class.

Cases NOT Suitable for Text Matching

- When the percentage of variable wording is greater than the percentage of recurrent wording on the page.

- When only a small relative percentage of wording is characteristic of the document class. For example, two documents, a “NOTE” and a “SECURITY INSTRUMENT” may contain the exact same paragraphs so they have up to 90% common wording and the only discriminating wording (10%) is found in their header. In this case, Text Matching classification will end up with conflicts because 10% is insufficient to differentiate between the two documents.
- When the recurrent wording is interrupted by non recurrent wording and is therefore spread over multiple pages. Since text matching works on a page level, scattered paragraphs of recurrent wording may prevent text matching to learn or classify the page. A workaround exists and is documented in the topic “Tuning Text Matching Classification” on page 86.

3.5.1.2 Learning Mechanism

For each image in the image base folders, the learning mechanism runs a full page *OCR*, extracts text strings and analyses the text strings to create templates. Processing time is proportional to the project size: whenever an unknown image is loaded, a full page OCR is run and the extracted set of strings is compared to that of the existing templates. The more the templates, the longer it takes to analyze a new image. Here is how text strings are processed to create templates:

- If the image text strings do not match any existing TM references, a new TM reference is built for the image. The new TM reference is added to the current *TM* template.
- If the image text strings match an existing TM reference of another template, a new TM reference is not created. The image is in conflict with existing TM references. A new TM template is also not created. Text Matching Designer has an option to view the images in conflict together with the conflicting templates. These images can also be exported (**Tools** > **Export skipped images in conflict**).
- If the image text strings match an existing TM reference of two other templates, a new TM reference is not created and the image is ignored by the system. This happens when an image has been previously learned. For example, no new template would be created in the case of a folder that contains only images already learned. Such a situation is prevented by a careful preparation of the image folders.
- If the image text strings match an existing TM reference of the current template, no new TM reference is created and the image is “skipped”, meaning that it is not taken into account. Text Matching Designer has an option to view the skipped images. These images can be exported (**Tools** > **Export skipped well classified images**).

3.5.1.3 Text Matching Mechanism

During classification, the system runs full page *OCR* on the image to extract the image text strings and compare them with the TM references of the TM templates. As a result of this comparison, the two best candidates are retained. Best candidates are those with the highest matching rates to the TM template. These ratios give an indication of the similarity between the tested candidates and the TM references of the TM template. In the following example, TM1 and TM2 are two different templates and are the two best candidates and that R1 and R2 are the ratios of TM1 and TM2 respectively. The example explains the decision mechanism of how the text matching parameters are applied. The three parameters described in this example are: **Matching threshold** (mT), **Minimum difference between two best candidate results** (MinDelta) and **As a second attempt, lower the matching threshold by** (SecondChance).

- If R1 ≥ mT > R2 and if R1 matches template TM1 while R2 matches template TM2, then the image is classified to TM1.
- If both R1 and R2 ≥ mT and if R1 matches template TM1 while R2 matches template TM2, then the parameter **Minimum difference between two best candidate results** (MinDelta) is applied as follows:
  - If R1 - R2 > MinDelta, then the image is classified to TM1.
  - If R1 - R2 < MinDelta, then the image is in conflict between TM1 and TM2.
- If R1 and R2 < mT and R1 matches template TM1 and R2 matches template TM2, the image is not classified.

3.5.1.4 High-level Steps to Implement Text Matching Classification

The high level steps for implementing text matching classification include:

1. Establish the top priority document classes. These are the classes which represent 80% of the activity.
2. Prepare the image base for the top priority document classes. Dedicate time to prepare an accurate image base.
3. Run text matching learning and tune the text matching settings until reaching the maximum accuracy. For documents that cannot be classified by text matching, consider creating subclasses as explained in the topic "Solving Conflicts" on page 86.
4. Focus on the next documents in the order of priority, so for example process medium-priority document classes. For documents that cannot be classified by text matching, consider manual classification.
5. Consider low-priority documents and decide whether they should be learned by text matching or classified manually. The decision depends on the number of documents and their frequency knowing that it is possible to create a text matching template with only one image.

3.5.2 Preparing the Image Base

Text matching is a technology that classifies a page based on the textual content (wording). Text matching includes an automatic learning mechanism that runs a full page *OCR* on an image base and processes the OCR results to extract the typical wording or typical sentences on the images to create the associated templates. However, contrary to automatic learning of standard templates, text matching does not perform relative learning of text matching templates. For this reason, a preparation phase is required before templates can be learnt. To avoid having to solve too many conflicting images, dedicate enough time to accurately prepare the image base.

Recommendations to Prepare the Image Base

Here are the recommendations to follow and information to know while preparing the image base:

- Sort the images and organize them into folders: a folder must contain the images that are typical of a document class. Text matching creates a template for each folder. Text matching considers that the full page OCR results from all of the images in a folder relate specifically to a template. Ultimately, text matching creates one template per folder. See the example provided in this section.
- Restrict the folder tree structure of the image base to two sub levels maximum.
- Have the image base represent several days of production, if possible, although text matching requires only one image to create a text matching template. Allow for at least one day of production.
- Ensure a large image base to reduce false positives even if it results more conflicts.

Note: The graphic layout can vary across documents of the same document class as long as the wording is similar.

An Example With a Three-Page Document:

For a three-page mortgage document, create a folder for Page 1 and a folder for page <*N*>. Place all of the mortgage Page 1 images in the Page 1 folder and the remaining Page 2 and Page 3 images in the Page <*N*> folder. For each folder, text matching will derive a set of strings, ultimately the template signatures.

3.5.3 Creating Text Matching Templates Automatically

Text Matching Designer automatically creates text matching templates and text matching reference images for one or more document pages. When a directory containing images is selected, each image present in the subdirectory is analyzed. A text matching template is created for each subdirectory contained within the selected directory. If there are no subdirectories, a text matching template is created for the selected directory. After all directories or subdirectories have been processed new text matching templates and/or text matching references are added to the templates list.

In the Text Matching Designer, the automatic results are displayed in the **Learning details** pane, in the following tabs: **Summary**, **Skipped images**, and **Images in conflict**. To show/hide the **Learning details** pane, select **Display** > **Display learning details**.

When a reference image is too similar to a reference image already defined in a text matching template, it is skipped and this result appears in the **Skipped Images** tab. When a reference image matches two templates, there is a conflict. Images in conflict are displayed in the list of the reference images in the left pane of the Text Matching Designer, preceded by  . When a reference image matches at least two reference images already in conflict, it is automatically skipped since Recognition Designer only manages conflicts between two images. Those skipped reference images are listed in the **Images in Conflict** tab.

To create a text matching template automatically:

1. Select the **File** > **Project Options** > **Classification** tab and select **Enable Text Matching Classification**.
2. Select the recognition engine for the text matching classification:
   a. Select **Classification** > Text Matching Designer.
   b. From the Text Matching Designer, select **File** > **Project Options**. The **Project Options** window appears.
   c. In the **Text Matching** tab, click  in the **OCR** pane. Select a recognition engine that supports full text recognition. For information on engine types that support full text recognition, see "Recognition Types Supported by Recognition Engines" on page 429.
   d. Select the image filters to improve *OCR*. Learn the purpose of each filter in the topic "Text Matching Tab" on page 281.
   e. Keep the default values of the parameters for now. Tuning will happen during testing as explained in "Tuning Text Matching Classification" on page 86.
3. Create a text matching template automatically:

a. From the Text Matching Designer, select **File** > **Learn**. The directory selection window appears.

b. Select a directory that contains one or more images in formats supported by the application. The images can be in one or more subdirectories of the selected directory.

c. To export skipped images for use at a later date or for comparison purposes, select the **Tools** menu from the Text Matching Designer.

3.5.4 Creating Text Matching Templates Manually

To create a text matching template manually:

1. Select **File** > **Project Options** > **Classification** tab and select **Enable Text Matching Classification**.

2. Select the recognition engine for the text matching classification:

   a. Select **Classification** > **Text Matching Designer**. The Text Matching Designer appears.

   b. From the Text Matching Designer, select **File** > **Project Options**. The **Project Options** window appears.

   c. In the **Text Matching** tab, click  in the **OCR** pane. Select a recognition engine that supports full text recognition. For information on engine types that support full text recognition, see "Recognition Types Supported by Recognition Engines" on page 429.

   d. Select the image filters to improve *OCR*. Each filter is described in topic "Text Matching Tab" on page 281.

   e. Keep the default values of the parameters for now. Tuning will happen during testing as explained in "Tuning Text Matching Classification" on page 86.

3. Create a text matching template manually:

   a. From the Text Matching Designer, select **Edit** > **Create new TM template**.

   b. Select one or more images in the **Open** window. The **Creation of a New TM Template** window appears.

   c. Type in a name for the new text matching template. An empty template is created.

   d. Select the created template and select **Edit** > **Add TM reference(s)** to add text matching references to the template. The **Open** window appears.

   e. Select one or more image files and select **OK**. In the Text Matching Designer, results are displayed in the **Learning details** pane, in the following tabs: **Summary**, **Skipped images**, and **Images in conflict**. To show/hide the **Learning details** pane, select **Display** > **Display learning details**.

Note: When a reference image is too similar to a reference image already defined in a text matching template, it is skipped and this result appears in the **Skipped Images** tab. When a reference image matches two templates, there is a conflict. Images in conflict are displayed in the list of the reference images in the left pane of the Text Matching Designer, preceded by . When a reference image matches at least two reference images already in conflict, it is automatically skipped since Recognition Designer only manages conflicts between two images. Those skipped reference images are listed in the **Images in Conflict** tab.

f. Select the **Edit** menu from the Text Matching Designer to cut and paste text matching references from one template to another.

3.5.5 Testing Text Matching Classification

When learning completes, carry out a test to detect the conflicts.

To test text matching classification:

1. From the **Text Matching Designer** menu, select **Test** > **Test**.
2. Select **Test** > **Run** to run the text matching classification test.

   The **Classification Test Results** window appears.
3. View the results of test text matching classification in the following tabs:

   - **Classified**: Displays images that are successfully classified to the project templates
   - **To confirm**: Displays unclassified images for which the system finds several candidate templates
   - **Not Classified**: Displays unclassified images for which the system does not supply a list of possible templates

   For the last two cases, the Identification operator will have to select a template manually.
4. Analyze the results and apply tuning as explained in "Tuning Text Matching Classification" on page 86.

3.5.6 Tuning Text Matching Classification

This topic offers recommendations to tune text matching classification parameters. Note that whenever you change the settings, you need to run the whole learning process again as described in the section "Creating Text Matching Templates Automatically" on page 83.

3.5.6.1 Reducing the Number of Templates

The ultimate goal of tuning is to reach the best possible classification rate with fewer templates and fewer conflicts. In the first step, it is recommended to learn all images using the default settings. If an image is not classified to the correct document class, it is important to learn it anyway even if it creates a conflict. This conflict will serve as a barrier against false positives.

To reduce the number of templates, follow these recommendations:

- If more templates than required are created due the recurrent wording being interrupted by non recurrent wording and being spread over multiple pages, try increasing these two advanced parameters: **Maximum difference between number of characters in document and TM reference** and **Maximum percentage difference between number of characters in document and TM reference**.
- If more templates than required are created because pages are very different (heterogeneous documents), it is possible to create fewer templates by processing only the pages that have more textual content. To do so, increase this parameter: **Minimum number of characters recognized in document needed to perform learning**.
- To reduce false positives, tune the matching threshold. Never set a matching threshold above 60% as in this case almost all images would be learnt as a new template.

3.5.6.2 Solving Conflicts

When text matching completes the learning process on the image base, it indicates the images that are in conflicts in the Text Matching Designer. Solutions to fix conflicts during the learning process are:

- Delete the template(s) causing the conflict.
- Review the classification rates and adjust the **Matching threshold** until the image is classified with the correct template.
- Avoid solving all conflicts. Consider that conflicts can act as barriers against false positives. Indeed, conflicts are presented to operators for manual classification so ultimately the conflicting image is correctly classified whereas false positives are not.
- Classify with sub-classes those images that are so similar that text matching cannot differentiate them. Then use pre-indexed fields to retrieve field values in a

script and classify the images to the correct templates. For example, some documents are very similar but need to be classified differently depending on their issue date: some with the template “before_march” and some with the template “after_march”. Text matching cannot classify them to two different templates. In this case, create a TM template “before_or_after_march” to classify all these similar documents. Use a pre-indexing field “issue_month” to retrieve the month from the documents. During classification, these documents are classified with “before_or_after_march”. Apply a script to this template. The script retrieves the value of the field “issue_month” and depending on the date, the script can classify the documents to the correct template: “before_march” or “after_march”.

3.5.6.3 Speeding Up Processing

The learning process takes from several minutes to several hours depending on the size of the project. The Text Matching Designer features advanced parameters to set a lower number of characters in the extracted text strings and therefore to accelerate processing speed. This topic describes the parameters that are intended to speed up both learning and classification processes and those that are available to improve accuracy.

To speed up processing by defining advanced parameters:

1. Select **Classification** > **Text Matching Designer**. The Text Matching Designer appears.
2. From the Text Matching Designer, select **File** > **Project Options**. The **Project Options** window appears.
3. Click **Advanced**. The **Text Matching Advanced Parameters** window appears. The parameters are:
   - The **Minimum number of characters recognized in document needed to perform learning** option sets the limit size of *OCR* strings for the image to be learnt or classified. By default, the minimum string length is 50 characters. This is to prevent images from being processed if they contain text strings that are too short.
   - The **Maximum difference between number of characters in document and TM reference** option sets a minimum threshold of difference that an image must not exceed with respect to the TM (text matching) reference. This is to prevent images whose OCR content is far from that of the TM reference to be processed. When this happens, the image is not processed and some processing time is saved. For example, if the document is composed of 250 characters and the text matching reference is composed of 100 characters, and if you define a maximum number of 70 characters in the option, then the document will not be classified since the difference between the document and the text matching reference is higher than 70 characters. This option is best used for short OCR strings.
   - The **Maximum percentage difference between number of characters in document and TM reference** option is similar to the option **Maximum**

difference between number of characters in document and TM reference except that it uses a relative value instead of an absolute one (percentage). This option is recommended for large OCR strings. See an example of how this parameter can be used in the section "Tuning Text Matching Classification" on page 86.

- The **Maximum number of mismatched characters allowed** option is to be understood as a correction option for the differences of processing between OCR engines. Depending on the engines, special characters, spaces or commas may be processed differently and this can cause differences with the TM references. The value of 20% is known to be appropriate for the Western OCR engine.

Note: Select **Default** to restore the initial default values for all these settings.

3.6 Keyword Classification

Keywords are separate words or combinations of words on a page that are indicative of a particular document class. Keyword classification searches the predefined keywords in images after full page recognition.

3.6.1 Understanding Keyword Classification

When performing keyword classification, the full page OCR is run on the defined reading zone. Keywords are searched in the *OCR* results and are compared to the rules. The image is classified if it matches one or another of the rules associated to the template.

- If the image matches a rule that is associated with more than one template, then there is a conflict between two or more templates. The algorithm applies the priority levels set for the rules. The image is classified with the template which has the highest priority rule.
- If the image matches a rule that is associated with different candidate templates and if this rule has the same priority level on all candidates, then the conflict is not solved and the image requires manual classification.

3.6.1.1 Keyword Classification Uses

Keyword classification is mostly directed at semi-structured and unstructured documents. In certain cases, it can be a backup solution for graphic templates.

Unstructured Documents

Keyword classification is the best method to classify semi-structured and unstructured documents where too many variations exist to be processed through graphic classification. Even if it is possible to create one graphic template for each variation, this has to be balanced with the design effort to create so many graphic templates and the delay before deploying the project. In this case, keyword

classification associated with free form rules for data extraction is preferred. General recommendations for setting up keyword classification are:

- Build general rules if the project requires only classification.
- Build page-specific rules if data extraction is required.

Backup for Graphic Templates

Consider keyword classification as a backup solution for structured documents which have failed the graphic classification or poor quality documents such as faxes. Graphic templates, in particular HPA, are very sensitive to the quality of printed characters. Poor quality documents such as faxes can contain areas where pixels are so degraded so that some graphic anchors may not match. If the project includes high volumes of documents to classify, it is worth considering create a generic template associated with keyword rules to backup an HPA template and increase the classification rate. Graphic classification of structured documents usually can yield up to 95% classification rate. For graphic templates that fail, backup them with keyword classification to get closer to 100% classification. This backup solution is worth considering mostly in mailroom applications when very high volumes of documents are classified and little data extraction is performed. To implement a backup solution:

- Create one generic template associated with keyword rules for each document class.
- Focus on finding keywords in a small reading area to avoid full page reading and reduce processing time. The page header is usually a good searching area to retrieve the most representative keywords.
- Give the generic template associated with keyword rules, the same name as that of the graphic template but preceded by a hyphen or any other character so that the “paired” generic and graphic templates are clearly visible in the list of templates. Absolutely follow this recommendation if you decide to use code-oriented keying. This will help you ensure you give the same template code to the paired generic and graphic templates.

3.6.1.2 Keyword Classification Settings

Reading Zone

The reading zone is not relative to document classes but there is a unique reading zone defined for the project. The reading zone can be the whole image, a third (upper, middle or lower) or a specific area to be selected within the image. If no data extraction is performed after classification, it is possible to save processing time by selecting keywords to be found in the page header (upper third) or the page footer (lower third) so as to avoid full-page reading. However, understand that data extraction requires full page reading. Keyword classification is usually combined with data extraction based on free form rules and full page reading is usually required. In this case, it is best to select also full page reading for keyword classification. Indeed, the *OCR* data from the Classification module are passed in the

form of OCR files to the Extraction module for data extraction so there is no need to carry out OCR reading again. In this case, if keyword classification is made on one third of the document, full page reading needs to be performed anyway in Extraction which ultimately consumes more processing time.

OCR Engine

The OCR engine is not relative to document classes but there is a unique OCR engine selected for keyword classification in the project. The default confidence threshold value depends on the selected engine. It is recommended to keep the default value as least in the first place as it is usually selected to best fit the engine. Keyword classification requires a full text engine. To save processing time during subsequent data extraction with free form rules, it is recommended to use the same OCR engine settings for both the free form rules and the keyword classification rules.

Note: The free-form image filters are automatically applied to keyword classification even if they cannot be selected in the keyword classification editor. Free form image filters are selected in the **Project Options**, in the **Recognition** tab.

Keyword Rules

Rules are very flexible in that they are composed of keywords and each keyword has its own properties. The number of rules in the project is unlimited. Create as many rules as required and associate them with one or several templates. However, only one OCR engine can be selected for keyword classification. During classification, a rule is true if all the keywords of the rule are found in the document. If the rule is true, the document is classified to the template that is associated with this rule. Rules also have the following two properties to address known pitfalls:

- **Priority** - Because an image can match several rules, there can be conflicts in classification. To solve conflicts, a priority level can be set for each rule. The priority is aimed at solving conflicts when an image matches several rules. Understand that a rule is associated with only one template which is why if an image matches several rules, it potentially matches several templates (conflicts). Learn how priority is applied to solve conflicts in the description of the algorithm next.
- **Proximity operator** - Optionally, a maximum distance between the keywords of a rule can be set. The search uses the boolean NEAR operator, a proximity operator. Distance is expressed in number of characters. This search is intended for some very specific documents where the keywords can be found in different areas of the document but need to be found near to each others for the document to match a given template. If the rule involves more than two keywords, the distance is calculated between pairs of keywords and none of the calculated distances must exceed the defined maximum distance. In some cases, this property may be more flexible than regular expressions. For example, this search is known to have been used in a mailroom project to identify a change of address. Rules used were: “moved” + “address” with a distance of 5 characters

between keywords and “new” + “address” also with a distance of 5 characters between keywords. Rules suggested here are preferred because it not recommended to use “new address” in case a term may be present between “new” and “address” and in this case the rule would not match. Finally, defining a distance prevents the system from retrieving all occurrences of “new” and “address” so that only the change of address is ultimately retrieved.

Keywords

A keyword can be a constant, an alphanumeric format, fuzzy regular expression, or a regular expression. A keyword if it is a constant can be case sensitive. A keyword can be given a name that displays in the name of the rule. The rule name is automatically composed of the names of the all the keywords that compose the rule (rule name cannot be modified). Giving a meaningful name to a keyword is recommended if the keyword format is a regular expression. Keywords also have the following three properties to address known pitfalls:

- **Isolated word** - A keyword can be set as an isolated word, meaning that it must be found between spaces. This is very useful namely for short terms that may be found within a bigger term. For example the term “search” is found in “searching” and “searchable”.
- **Hit threshold** - A keyword can have a threshold value. The higher the threshold, the closer the read keyword should be to the specified keyword. For example, when searching he keyword constant “amount” with a threshold value set to 90%: if the OCR engine returns “amovnt” with a confidence of 91%, the returned value is accepted.
- **Anti-keyword** - A keyword can be set as an anti-keyword. An anti-keyword serves to validate the template when it is NOT present in the document.

Recommended Number of Keyword Rules

Ideally create no more than 100 rules per project; do not exceed 200 rules. More than 200 rules are difficult to maintain. To help limit the number of rules, analyze the document spectrum of documents to be classified with keyword classification and create three categories: most frequent, frequent, and others. Build up to 3 rules for very frequent documents, up to 2 rules for frequent documents and 1 rule for the others.

Detection of the Document Type: An Example

Rules must identify the document type so that it is associated with the appropriate data extraction settings (often, free form rules). Analyze the document title on the top of the page. Select constants (usually terms) away from handwritten characters and away from page edges. Ensure terms are specific in the document; they may be repeated several times in the search area. For example: APPLICATION+LIFE INSURANCE APPLICATION+ COMPANION. The first keyword (APPLICATION) confirms the document type. The second and next keywords (LIFE INSURANCE APPLICATION) are to discriminate against other possible candidates for the document type. Set these keywords as non case sensitive and with a hit threshold of

80%. Set a proximity of characters (option **Keywords nearby**) of 100 characters and a **maximum** priority.

Detection of Page Numbers - Examples

Detecting page numbers is useful for multiple-page documents to associate the appropriate data extraction settings. If there are no other elements to discriminate better between the different pages of the document, detecting page numbers is a possibility. For example, if the document has three pages, create three generic templates, one for each page and then build rules to identify the page number. When creating the rules, target a classification rate close to 100% for the first page as it is generally used for data extraction. For the first page, create three rules. The second and following pages are less frequently used for data extraction so they usually require two rules. This is a good tip to limit the number of keyword rules in the project. Here are three examples of rules to detect page numbers:

- **Detection of document type and page 1**: Detect the document title and the page number on the top of the page by creating a rule with 3 keywords (logical "AND"). For example: LIFE INSURANCE APPLICATION + (?i)Page[\.:\s]*1\s + POLICY. The first keyword is a constant to confirm the document type. Select 2 or 3 words from the document title, set it as case sensitive with a hit threshold of 80%. The second keyword is a regular expression to confirm this page is "Page 1". The third keyword is a constant to confirm that this page is really "Page 1". Select 2 or 3 words, set them as non case sensitive with a hit threshold of 80%. For Rule 2, set a proximity of characters of 50 characters and a **high** priority.
- **Detection of document type and pages 2 and next**: Build a rule with 2 keywords, a constant to confirm the document type and a regular expression to confirm the page number. The first keyword can be composed of 2 or 3 words found in the document title. Set them as case sensitive with a hit threshold of 80%. The regular expression is to confirm the page number. You can use for example a regular expression such as (?i)Page[\.:\s]*1[0-9]*\(?i)Page[\.:\s]*[02-9][0–9]*. For the rule, set a proximity of 50 characters and a **standard** priority.
- **Confirmation of page number**: Build a rule with 3 or 4 keywords (logical "AND"). The first keyword is a regular expression to confirm the document type. For example, \d{2}-\d{4} \(\d{4}\) to retrieve a format or ID of type "90-1234 (9999)". The second keyword is a regular expression to confirm that this page is NOT page 1. Use the same regular expression for the page format as for the rule to detect page 1 (see previous paragraph) but select the **excluded** option. The third and fourth keywords are constants to confirm that this page is NOT page 1. For each keyword, select 1 or 2 words that are specific of Page 1, select the option **excluded** and set these keywords are non case sensitive with a hit threshold of 80%. For Rule 2, set a proximity of 50 characters and a **standard** priority.

Keyword Classification Algorithm

The main steps of the keyword classification algorithm are:

1. Full page OCR is run on the defined reading zone.

2. Keywords are searched in the OCR results and are compared to the rules.
3. The image is classified if it matches one OR another of the rules associated to the template.
4. If the image matches a rule that is associated with more than one template, then there is a conflict between two or more templates. The algorithm applies the priority levels set for the rules. The image is classified with the template which has the highest priority rule. If the image matches a rule that is associated with different candidate templates and if this rule has the same priority level on all candidates, then the conflict is not solved and the image requires manual classification.

3.6.1.3 High-level Steps to Implement Keyword Classification

The high level steps for implementing text matching classification include:

1. Set up the OCR parameters to configure the keyword classification engine. These parameters will apply to all images in production, irrespective of the document class.
2. Create as many generic templates as document types in the production flow. Associate each template with a reference image.
3. Create keyword rules for each generic template. The rules are used to detect if a recognized document represents a particular document class and can be assigned to the template linked by those rules.
   a. Create a rule and specify its priority: standard, high, or maximum.
   b. Select the generic template to be associated with the rule.
   c. Add keywords. Perform keyword testing.
   d. Test the keyword rule.
4. Test the keyword rules. Edit the rules if necessary.

3.6.2 Setting the OCR Parameters

1. Open the recognition project in Recognition Designer.
2. Select **Classification** > **Keyword Classification**.
3. In the **Edit Keyword Classification** window, click the **OCR parameters** button.
4. Specify the OCR parameters:
   a. In the **Reading zone** area, specify the zone on the image in which recognition will be performed. The options are: **Full Page**, **Upper third**, **Middle third**, **Lower third**, **Custom size (mm)**.

      To specify a custom zone:

      - Select the X and Y coordinates of the left upper corner of the reading zone

- Specify the height and width of the zone relative to the left upper corner.

The specified reading zone will be colored on the sample image.

b. Select an **OCR Engine** that supports full text recognition. Find the list of supported full text recognition engines in section "Recognition Types Supported by Recognition Engines" on page 429.

c. Specify the OCR confidence threshold in a box next to the selected `.reco` file. If no threshold has been specified, all characters will be recognized with a default threshold value. For details on how the confidence threshold works, see "Recognition Engine Confidence Threshold" on page 110.

For preliminary testing, it is recommended that no threshold be specified.

5. Click **Close** to exit the **Edit Keyword Classification** window. To save the settings, click the "save" icon in Recognition Designer.

3.6.3 Preparing Templates for Keyword Classification

Before you start designing keyword rules, you need to create a generic template for each type of document in the flow, such an invoice, a bill, or other. Every keyword rule created for identification of a particular type of document must be attached to a corresponding template. In production, keyword classification performs full text recognition from the image and applies all available keyword rules to the extracted content. The matching rule indicates a template to which the image will be classified.

To create a generic template:

1. Open the recognition project in Recognition Designer and click **Classification View**.
2. Right-click in the **Template** list area and select **New Generic Template** from the content menu.
3. In the system dialog, navigate to the image to be assigned to the generic template. Recognition Designer requires an image attached to a generic template even if this image is not used as a reference for classification. Select the image and click **Open**.

Note: Ensure the image format is supported. Find the full list of supported image formats in "Advanced Recognition Supported Image Formats" on page 426.

A new generic template appears in the **Template** list.

3.6.4 Creating and Editing Keyword Rules

1. Open the recognition project in Recognition Designer.
2. Select **Classification** > **Keyword Classification**.
3. In the **Edit Keyword Classification** window, click the **Edit rules** button.
4. To create a rule, click the **+** button located below the grid in the **Rules** pane. A new line appears in the grid. All other panes and controls are empty.

   To edit an existing rule, click it in the **Rules** list. All other panes and controls display the settings of the selected rule.
5. In the **Edit a rule** pane, expand the **Template** drop-down list and select a generic template for which you create a rule.

   The **Template** list only displays generic and text matching templates created in the project.
6. In the **Edit a rule** pane, set the priority level for the current rule: **Standard**, **High** or **Maximum**.

   If a document matches several keyword rules (in other words, if a document matches several templates), the document will be classified to the template that matches the rule with the highest priority level.
7. To add a keyword to the rule, or edit the existing keyword, click the **+** button below the **Keywords** grid.

   The **Define Keyword** window opens.
8. In the **Category** pane, select a keyword category:

   - **Constant**: The engine will search a specific term, amount or date.

     Select **Case sensitive** if required.

     For constants, specify the **Hit threshold** percentage. The higher the threshold, the closer the characters should be to the specified value. For example, if the threshold is 80%, 80% of the characters must match. If the threshold is 100% then 100% of the characters must match.
   - **AN format**: Accepts alphanumeric characters. Type a valid format syntax. For example, “1N5A7X2C” indicates that a value with one numeric, five alphabetic, seven alphabetic or numeric, and two characters of any type will be valid.
   - **Regular expression**: Use regular expressions to search patterns. Learn about regular expressions in “Regular Expressions” on page 96.

     Fuzzy regular expression: Unlike the **Regular expression** option, fuzzy regular expressions take advantage of the **Hit threshold** option, which enables a single regular expression to have a wider range of text matches. For more information, see “Fuzzy Regular Expressions” on page 100.
9. (Optional) Select **Isolated word** to find an isolated string that is preceded and followed by spaces.

10. (Optional) Select **Excluded** to create anti-keywords, that is, keywords that are valid if they are not found in the document.

11. After creating a keyword, click **Add**. After editing, click **OK**.

12. Test the keyword as described in section "Testing Keywords" on page 103.

13. Test the keyword rule as described in section "Testing Keyword Rules" on page 103.

3.6.4.1 Regular Expressions

Regular expressions are used to build rules for keyword classification and to build free form rules with Free Form Designer.

A regular expression is a string used to describe or match a set of characters, according to certain syntax rules. The expression, for example, can search for specific characters, the position of characters in a string, or specific grouping of characters. Regular expressions are useful for identifying partial or entire strings during recognition, and validation. They can be used to search for Unicode characters. The tables included here give some common elements and examples. For a complete description of regular expressions, see the Microsoft *MSDN* Website and search on regular expressions.

This table gives examples of regular expressions used to verify some common items. These regular expressions are built using "Regular Expression Syntax Elements" on page 97.

Table 3-5: Regular Expression Examples

Regular Expression	Usage	Example
`[A-Za-z]+\ ?[A-Za-z]*`	One or two consecutive words (allowing for mistyped lowercase initial)	Development Engineer
`\d{3}-\d{3}-\d{4}`	Social security number (in different notation)	234-543-1234
`\(\d{3}\)\d{3}-\d{4}`	Phone number (in different notation)	(555)555-1234
`(?i)(jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec) \d{1,2} (\d{4}|\d{2})`	Date format	Feb 10 2005
`acct|account`	The abbreviation "acct" or the word "account"	acct, account
`\d{1,2} ?/ ?\d{1,2} ?/ ?(\d{4}|\d{2})`	Date format with optional space around slashes	10/2/05 10 / 2 / 05

Regular Expression	Usage	Example
`\d{1,2} ?\. ?\d{1,2} ? \. ?(\d{4}|\d{2})`	Date format with optional space around dots	10 . 02 . 2005 10.02.2005
`(?i)invoice ?`	The word "invoice", regardless of case	INVOICE, Invoice, invoice, followed by one or no spaces
`[\dO]+( ?[\.,;o] ? [\dO]{2})?`	Value format with possible decimal separators, and exactly two digits after the separator	123456789.00, 1234,56, or 45.00
`[\dO]+ ?[\., ;o]( ? [\dO]){3}( ?[\., ;o] ? [\dO]{2})?`	Value format with possible thousands separators and decimal separators, and exactly two digits after the separator	100 000.00
`[a-zA-Z0-9]+[@][a-zA-Z0-9]+[.a-zA-Z0-9]+`	E-mail address	JohnDoe@abcxyz.com
`\p{<IsCyrillic>}`	Set of Cyrillic characters	Any Cyrillic character
`(?<=\$)\d+`	Any amount preceded by "$" value. The regular expression retrieves the amount without the "$" value.	123
`\d+(,\d{2})(?!FRF)`	Any amount with a comma and two decimals and not followed by "FRF" value	1.918.25

3.6.4.2 Regular Expression Syntax Elements

Regular expression syntax elements are grouped here by the type of characters to search for and the specific recognition or validation task to perform.

- *Quantifier* elements are regular expression elements that specify numbers of the indicated item.
- *Character Class* elements specify specific characters, groups of characters, words, digits, and others.
- *Grouping* elements designate the way in which characters or syntax elements are grouped in regular expressions.
- *Literal* elements either identify tabs or line ending characters, or allow use of quantifier, character class, or grouping element items as literal characters.

Table 3-6: Regular Expression Quantifier Elements

Quantifier Elements	Description
?	Zero or one occurrence of previous element.

Quantifier Elements	Description
*	Zero or more occurrences of previous element.
+	One or more occurrences of previous element.
{<*n*>}	Specifies exactly <*n*> matches; for example, (pizza){2}.
{<*n*>,}	Specifies at least <*n*> matches; for example, (abc){2,}.
{<*n*>, <*m*>}	Specifies at least <*n*>, but no more than <*m*>, matches.
*?	Specifies the first match that consumes as few repeats as possible (equivalent to lazy *)
+?	Specifies as few repeats as possible, but at least one (equivalent to lazy +).
??	Specifies zero repeats if possible, or one (lazy ?).
{<*n*>}?	Equivalent to {<*n*>} (lazy {<*n*>}).
{<*n*>,}?	Specifies as few repeats as possible, but at least <*n*> (lazy {<*n*>,}).
{<*n*>, <*m*>}?	Specifies as few repeats as possible between <*n*> and <*m*> (lazy {<*n*>,<*m*>}).

Table 3-7: Regular Expression Character Class Elements

Character Class Elements	Description
[...]	Matches any one character between the brackets. For example [abc] would find any occurrence of the characters a, b, or c.
[^...]	Matches any one character not between the brackets. For example [^xyz] would find any occurrence of characters except x, y, or z.
-	Use of a hyphen (-) allows specification of contiguous character ranges. For example, [<*3*>-<*7*><*a*>-<*f*>] matches any digits between 3 and 7 (inclusive) and any characters between a and f (inclusive, and lower case).
.	Any character except newline. Equivalent to [^\n].
\w	Any word character. Equivalent to [a-zA-Z0-9_].

Character Class Elements	Description
`\W`	Any non-word character. Equivalent to `[^a-zA-Z0-9_].`
`\s`	Any whitespace character. Equivalent to `[\t\n\r\f]`.
`\S`	Any non-white character. Equivalent to `[^\t\n\r\f]`.
`\d`	Any digit. Equivalent to [0-9]. For example, Social Security number = `\d{3}-\d{3}-\d{4}`.
`\D`	Any character other than a digit. Equivalent to `[^0-9]`.
`\p{`*<name>*`}`	Matches any character in the named character class specified by *<{name}>*. Supported names are Unicode groups and block ranges. For example, `Ll, Nd, Z, IsGreek, IsBoxDrawing`.
`\P{`*<name>*`}`	Matches text not included in groups and block ranges specified in `{`*<name>*`}`.
`\A`	Matches the start of the text.
`\Z`	Matches the end of the text.
`\x`	Matches a hexadecimal character.
`(?i)`	Makes the pattern to the right case-insensitive.

Table 3-8: Regular Expression Grouping Elements

Grouping Elements	Description
`|`	Or. For example, *<exp1>*`|`*<exp2>* matches the expression exp1 or the expression exp2.
`(...)`	Use with repetition characters.
(?=*subexpression*)	Positive lookahead assertion. Matches expressions followed by a subexpression.
(?!*subexpression*)	Negative lookahead assertion. Matches expressions not followed by a subexpression.
(?<=*subexpression*)	Positive lookbehind assertion. Matches expressions preceded by a subexpression.
(?<!*subexpression*)	Negative lookbehind assertion. Matches expressions not preceded by a subexpression.

Table 3-9: Regular Expression Literal Character Elements

Literal Character Elements	Description
\n	Finds new line indicators, equivalent to
\r	Finds return indicators, equivalent to \x0d.
\t	Finds tabs, equivalent to \x09
\f	Matches a form feed, equivalent to \x0c.
\a	Matches an alarm bell, equivalent to \x07.
\e	Matches an escape, equivalent to \x1b.
$	Matches the end of a line.
\<*regular expression element*>	Use the backslash in front of any of the regular expression elements when using that character as a literal. Here are the elements that, to use as literals, must be preceded by the backslash. • \ • $ • . • ^ • ? • * • + • (•) • [•] • { • |

3.6.4.3 Fuzzy Regular Expressions

Like the standard **Regular expression** option, fuzzy regular expressions are used to build rules for keyword classification, zonal recognition, and to build free form rules with Free Form Designer. However, unlike the **Regular expression** option, fuzzy regular expressions take advantage of the **Hit threshold** option, which enables a single regular expression to have a wider range of text matches. Therefore, a fuzzy regular expression can compensate for small but consistent OCR inaccuracies.

You must accurately specify both the regular expression and hit threshold to have a successful fuzzy regular expression. Hit threshold specifies the percentage of the characters in a fuzzy regular expression that result in a match; that is, the higher the hit threshold, the closer the string must be to the regular expression. This means that

if the threshold is 80%, then only 80% of the characters in a string must match the regular expression. For example, if the regular expression for a US social security number is the following:

```
\d{3}-\d{2}-\d{4}
```

and the hit threshold is set to 91%, then one character in an otherwise matching string does not have to be a digit or a hyphen. In this case, the following string is a match:

```
a62-25-7294
```

If a character is not recognized, then a question mark (?) is substituted for it in the output.

Note: The hit threshold for fuzzy regular expressions should be less than 100%; otherwise, the results are equivalent to its corresponding standard regular expression.

In addition, fuzzy regular expressions have the following usability advantages over the **Regular expression** option:

- Less complex regular expression construction and fewer regular expressions
- Reduced maintenance
- Application to a wider range of OCR engines
- Less scripting

The following regular expression elements are supported:

- Regular Expression Quantifier Elements

Quantifier Elements	Description
?	Zero or one occurrence of previous element.
*	Zero or more occurrences of previous element.
+	One or more occurrences of previous element.
{<*n*>}	Matches the previous element exactly <*n*> times.
{<*n*>,}	Matches the previous element at least <*n*> times.
{<*n*>,<*m*>}	Matches the previous element at least <*n*> times, but no more than m times.

- Regular Expression Alternation Elements

Alternation Elements	Description
|	Or. For example, *<exp1>*| *<exp2>* matches the expression *<exp1>* or the expression *<exp2>*.

- Regular Expression Grouping Elements

Grouping Elements	Description
(*<subexpression>*)	Captures the matched *<subexpression>* and assigns it a one-based ordinal number.

- Regular Expression Character Class Elements

Character Class Elements	Description
`.`	Any character except newline. Equivalent to `[^\n]`.
`\w`	Any word character. Equivalent to `[a-zA-Z0-9_]`.
`\W`	Any non-word character. Equivalent to `[^a-zA-Z0-9_].`
`\s`	Any whitespace character. Equivalent to `[\t\n\r\f]`.
`\S`	Any non-white character. Equivalent to `[^\t\n\r\f]`.
`\d`	Any digit. Equivalent to `[0-9]`. For example, the regular expression for a US social security number is: `\d{3}-\d{3}-\d{4}`
`\D`	Any character other than a digit. Equivalent to `[^0-9]`.
[*<character_group>*]	Matches any single character in *<character_group>*. By default, the match is case-sensitive.
[^*<character_group>*]	Negation: Matches any single character that is not in *<character_group>*. By default, characters in *<character_group>* are case-sensitive.

3.6.5 Testing Keywords

1. Open the recognition project in Recognition Designer, select **Classification** > **Keyword Classification**. In the **Edit Keyword Classification** window, click the **Edit rules** button.
2. Select the keyword rule in the **Rules** list. The **Keywords** list displays the keywords of the selected rule. Double-click the keyword.
3. In the **Define Keywords** window, click **Test**.

   The **Test** window is different according to the selected category of keyword.

   In the **Search zone**, type in a text that will contain the searched word. This option ensures that the keyword will be correctly recognized during *OCR* testing.
4. Click  to start the test.
5. Click  to reach the best result obtained by the algorithm.

   You can edit a **Constant**, **Regular Expression**, or **Fuzzy regular expression** in the **Test** window and click the **Apply** button to save those changes. For more information about writing regular expressions, see "Regular Expressions" on page 96.
6. (Option) In the **Edit a rule** pane, check the **Keywords nearby** option (if the keyword rule has at least two keywords) and select the maximum number of characters expected to be found between the two keywords.

   Note: If the project contains lots of rules (100 to 150), right-click the list of rules in the **Rules** pane and select **Export the List of Rules** to export the list of rules to a `XLS` or `CSV` file. Having the rules in a spreadsheet facilitates sorting and filtering for example to search for redundant rules.

3.6.6 Testing Keyword Rules

The objective is to ensure that rules are discriminant enough but not too restrictive. This is the main difficulty when testing rules. If rules are too restrictive, they do not classify 100% of the document classes. If they are not discriminant enough, they classify to other document classes, in other words, there are false positives and/or conflicts. Tuning must enable 100% of classification and no false positives. It is a best practice to use the image base of the graphic templates so you also ensure consistency between graphic classification and keyword classification.

- Test the rules against the document classes for which they have been developed. Ensure to reach 100% of classification. Otherwise, understand that rules are too restrictive.
- Test the rules against all the other document classes (that are not meant to be classified with keyword rules) and ensure that 0% of documents are classified. Any document classified here is a false positive. If there are many false positives,

consider that rules are not discriminant enough. To make rules more discriminant, add a new keyword or consider using an anti-keyword (exclusion).

- Fine tune the rules using the evaluation base.

To test a keyword rule:

1. Open the recognition project in Recognition Designer, select **Classification** > **Keyword Classification**.
2. In the **Edit Keyword Classification** window, click **Test**.
3. Select a rule to be tested in the **Rule** list or select **All rules**.
4. Click  to load images for the test or click  to load a tree.
5. Click  to start the test on all the images, or select one image in the list to start the test on that image (the selected image is displayed in the right pane of the window).

   The test results appear in the three following panes : **Results per image**, **Results per rule**, and **Content**.

Note: This test is for keyword rules only. To test keyword classification, run a classification test.

3.7 Testing Classification

Testing classification enables checking classification performance, visualizing classified and unclassified images, checking the pre-classification and decision rates, and exporting test results and classified and unclassified images.

You can also use PDF files as a test base, which also loads their associated OCR data caches.

To run a classification test:

1. Select **Test**> **Classification Test**.
2. Select **File** > **Open** and select the images to be tested.
3. Select **Test** > **Classification Test**. The **Recognition Designer <project name>** window displays and all the project images you selected are loaded by default. Images can be deleted or reloaded using the **File** menu options.
4. Select the **Test** menu. Run classification tests on all templates in the project, or simply on specific types of template. This is useful when running tests on projects that have templates of different types containing a large number of images. Running classification tests on such projects can be a very lengthy process, so limiting the test to a specific template type can considerably reduce the time it takes to carry out the **Classification Test**. If you do not select a template, then all project templates run.

5. Select **Test** > **Run**. The **Classification Test Results** window appears.
6. When the test completes, click **OK**.
7. The status bar displays the classification performance in terms of the number of processed images, the number of images processed per second and the number of images classified. Tested image information displays in the three following tabs:
   - **Classified** displays successfully classified images. **File** indicates the complete path of the image, **Name** and **Code** indicate the name and code of the template to which the image is classified. If the detection of rotation and/or side flipping is enabled, columns appear indicating rotated and/or side-flipped images.
   - **To confirm** displays unclassified images for which the system finds several candidate templates (which are displayed to the operator in the Identification module). The **Conflict** column displays the ID numbers of the candidate templates (the ID number is found in the **Classification View** > **Template properties**). The **Results for current document** pane displays the **Name** and **Code** of each candidate templates.
   - **Not classified** displays the images that cannot be classified. For these images, the system does not supply a list of possible templates and the operator has to select the template in Identification module.
8. Select **Display** > **Advanced Information** to display the pre-classification and decision columns.
9. Select **File** > **Export results** to export the test results (including the pre-classification rates and the decision rates) or export images by right-clicking on one or more images from the **File** list and by selecting **Export the following Images** from the menu that opens from the list.
10. In the **Export results** window, select the export directory and the export format (either *XML* or *CSV*.
11. Select **File** > **Export the following images**. The **Image Export** window displays for selecting images to export from the **Classified** tab the **Not Classified** tab or the **To Confirm** tab.
12. In the **Image Export** window, select the destination using the **Browse** button.
    - If exporting images from the **To Confirm** tab or the **Not Classified** tab, images are exported directly to the selected destination directory.
    - If exporting from the **Classified** tab, select a **Grouping**. Select **Per code** to create subdirectories that take the name of the template codes. Each classified image is copied to the subdirectory that corresponds to its template code. Select **Per template** to create subdirectories that take the name of the template. Each classified image is copied to the subdirectory that corresponds to its template name. Select **Per code** then template to create subdirectories that take the name of the template code and then the template name.

Once the export is completed, a message displays the number of images exported and indicates if any special characters have been replaced. The special characters of template names and codes are replaced by <_>. Duplicate images are renamed. For example, files using the name `image.tif` would be renamed `image(1).tif`, then `image(2).tif`, and so on.

Related Topics:

"Classification Test Results" on page 306

3.7.1 Understanding Pre-Classification and Decision Rates

To speed up classification with the *HPA* and standard methods, a pre-classification phase is carried out automatically during classification. Pre-classification consists of retrieving a subset of best-matching templates through a quick comparison of documents to the templates. Recognition Designer applies default pre-classification thresholds that are suitable for most projects.

When running a classification test, display the pre-classification and decision rates or adjust the pre-classification rates for standard templates and HPA templates. For help, see Testing classification.

Pre-classification and decision rates depend on the type of template:

- The standard template and HPA template pre-classification rate is the result obtained by each image compared to the pre-classification threshold (70% by default). The image is classified if the rate is greater than the threshold, and not classified (displayed in the **Not Classified** tab) if the rate is less than the threshold.
  - All the standard templates of a project have the same pre-classification threshold. The decision rate is calculated only for images that have passed the pre-classification test. Adjust the pre-classification threshold on the **Classification** tab.
  - For the HPA templates, adjustment of the **pre-classification threshold** is defined for each anchor. This threshold is the matching rate that is required between a document and the HPA template for the document to be a potential candidate. A high pre-classification threshold is recommended. By default, it is set to 70%. For checks a pre-classification threshold of 49% to 50% is recommended. See "Editing HPA Templates" on page 72.
- The hand printed template pre-classification rate has no meaning for documents identified as being hand printed. The decision rate is always 100%.
- The text matching template pre-classification rate is 45% (default value). Documents that have a matching rate of above 45% with one of the text matching references are classified. When a document can be classified as two distinct text matching templates, the difference between the matching score for the first and second hypothesis must be greater than the minimum difference between two best candidate results. The default value for this option is 20%.

- The classification templates using keyword rules pre-classification rates are fixed at 100%.

3.8 Assigning Templates to Hand-Printed Documents

Recognition Designer can detect whether a document is hand printed or not by selecting a template to be assigned to all hand printed documents. This is usually a generic template. If no template is assigned to hand printed documents, they are assigned to the default template.

To assign a template to hand printed documents:

1. Select the **Classification View**.
2. Right-click the **Template** list and select the **New Generic Template** menu.
3. Select the image to be used as the reference image, an image using hand printed text.
4. Name the template.
5. Select **File** > **Project Options**, the "Project Options" on page 276 window appears.
6. Select the **Classification tab**.
7. On the **Classification tab**, select the option **Assign hand printed documents to a template**.
8. Select the generic template created for hand printed documents from the **Templates** list box.

3.9 Setting a Default Template

At the end of classification, if any document does not match a template (in other words, if it is not classified), it is associated with the default template, assuming a default template has been specified.

To assign a default template to unclassified documents:

1. Select **File** > **Options**, the **Project Options** window appears.
2. In the "Project Options" on page 276 window, select the **Classification** tab.
3. Select the option **Assign a template by default**.
4. Select the template to assign to all non classified documents from the **Templates** list box.

3.10 Defining OCR Engines

Recognition engines interpret images and return the equivalent text, barcode, or other information depending on the nature of the image data. Selection of the appropriate recognition engine is important to analyze images properly and return the correct data. Recognition Designer provides several Optical Character Recognition (*OCR*) and Intelligent Character Recognition (*ICR*) engines.

For each selected engine, an engine configuration file (`<*>.reco`) is defined and associated with a project. The engine configuration file stores the customizable parameters of the engine.

Recognition Designer includes predefined engine configuration files designed for many common tasks. These predefined files can be edited to suit specific needs, or you can create custom configuration files based on these predefined files.

Recognition engine configuration files are assigned at the time when fields are placed on templates. Recognition engines can be reassigned for fields during testing for help in selecting the most appropriate engine for the template.

In production, the engine configuration files are assigned during recognition, so operators do not select the recognition engines directly at that time. To improve recognition efficiency, configuration also enables assignment of *confidence thresholds*, *filters*, and can be based on *language*.

Note: In Core Capture, only the Advanced OCR/ICR and Western OCR engines are supported.

3.10.1 Selecting the Appropriate Engine

When selecting a recognition engine for a particular recognition case, consider the following:

- **Type of content to be extracted**: Recognition engines are specific to a particular type of content they can recognize. By matching the appropriate recognition engine to the type of characters to find on a document greatly increases recognition success. Find the full list of recognition engines sorted by supported type of content and other characteristics in the "Recognition Types Supported by Recognition Engines" on page 429 section.
- **Language of content to be extracted**: Recognition engines differ in the list of supported languages they can process. Find the full list of recognition engines sorted by supported languages in the "Languages Supported by Recognition Engines" on page 427 section.
- It is strongly recommended that you use one of the Advanced Zonal OCR/ICR engines for all recognition projects where you have a choice as to which recognition engine to use.

3.10.2 Adding Engine Configuration Files to the Project

The engines that can be used for data recognition in production need to be specified in the recognition project. For this purpose, every engine needs configuration files be added to the project.

Recognition Designer allows you to select the required engines from the list of available engines that ship with Intelligent Capture. Every selected engine opens its properties for editing. When you save your preferences, a new engine configuration file is created and associated with the project. Recognition Designer also ships with a collection of standard engine configuration files that can be customized and added to the project.

To add an engine configuration file to the project:

1. Select **Tools** > **OCR/ICR Engine** > **New**.
2. Select the recognition engine from the list of available engines.

   The engine window displays with the default parameters specific to the selected engine.
3. (Optional) Customize the engine parameters.
4. Click **OK** to save it.

The saved configuration file becomes available for editing in the **Local Resources** tab of the **Select Resources** window.

To edit an engine configuration file:

1. Select **Tools** > **OCR/ICR Engine** > **Edit**.
2. In the **Select Resources** window, select the **Local Resources** tab. Select the file and click **Select**.
3. The **<***Engine Name***>** window, customize the engine settings and click **OK**.

To add a customized standard configuration file:

1. In Recognition Designer, select **Tools** > **OCR/ICR Engine** > **Edit**.
2. In the **Select Resources** window, select the **Global Resources** tab. Right-click the file and select **Copy to Local Resources** from the popup menu.
3. Enter the name of the engine configuration file in the **Copy Global Resource Locally** window and click **OK**. This makes a copy of the file on the **Local Resources** tab.

   Note: The following characters are reserved and must not be used in OCR/ICR engine names or engine configuration files: “|”, “\”, “/”, “<”, “>”, “” (space), “:”, “*”, “?”, “,”, “;”, “$”, “_” (underscore)
4. Modify the file from the **Local Resources** tab and click **OK**.

3.10.3 Assigning an Engine Configuration File to a Placed Field

Placed fields use zonal recognition on structured documents where fields are placed in specific locations on templates. During field placement or creation, the engine configuration file is assigned to the field based on the type of data that resides in the location.

Note: You can also set the OCR engine and its confidence threshold value in **File** > **Project Options** > **Recognition** > **OCR engine for anchor text values**.

To assign an engine configuration file to a placed field:

1. Select **Indexing** > **Index View**.
2. Select a placed field.
3. In the **Recognition** tab of the **OCR engine** pane, click the folder icon to open the **Select Resources** window.
4. On the **Global Resources** tab, select one of the standard configuration files. Or, select the **Local Resources** tab to select a locally created custom configuration file.
5. Select a standard engine configuration file from the **File** list.
6. In the **Field parameters**, next to the **OCR Engine** field, specify a confidence threshold value. For details, see "Recognition Engine Confidence Threshold" on page 110.

3.10.4 Recognition Engine Confidence Threshold

Confidence thresholds control character recognition based on how closely an image character matches a recognition engine character. Recognition engines return confidence level values during testing so accuracy can be determined and confidence thresholds can be defined. Confidence threshold is defined as two values in the form X,Y.

- X is the minimum threshold that a character must meet or surpass to be recognized.
- When two characters are recognized as potential matches, Y is the minimum difference in returned threshold values between two returned characters. When two characters are potential matches during recognition, the minimum difference between the confidence threshold of the first and second candidates. For example, consider the characters "v" and "u". If the character to recognize is "v", entering *<80>*, *<20>* as confidence threshold values can return these potential results:
  - "v" is returned if the value 90 is returned for "v" and 54 is returned for "u", because 90 > *<80>* and (90-54) > *<20>*

- "?" is returned if the value 90 is returned for "v" and 85 is returned for "u", because (90 - 85) < *<20>*
- "?" is returned if the value 78 is returned for "v" and 50 is returned for "u", because 78 < *<80>*
- "?" is returned if the value 85 is returned for "v" and 70 is returned for "u", because (85 - 70) < *<20>*

Recognition engines differ on the precision of threshold values. Some return values down to a single digit, while others return a confidence level value rounded to the nearest ten. To review confidence levels, run a unit test and move the cursor over the value returned for a character to display confidence information.

Related Topics

"Assigning an Engine Configuration File to a Placed Field" on page 110

3.10.5 Applying Filters to Improve Recognition

Filters can improve recognition by cleaning up images prior to recognition. Improving recognition can be accomplished by erasing graphic elements, such as checkboxes or lines, or dilating or eroding pixels at the edges of characters. Several filters can be applied to a field, and the order in which the filters are applied can be set to improve cleanup.

To apply a filter to a field:

1. Select **Indexing** > **Index View**.
2. From the template, select a field that is placed on the template.
3. From below the list of fields, select the **Image Clean Up** tab.
4. Click **Add** and select a filter from **Select Resources** the **Global Resources** tab. Selecting a filter populates the **Result** and **Help** panes which provide an example and description of the filter. If selecting several filters, use the red arrows to rearrange filters to improve recognition.
5. Click the **Delete** button to remove a filter from the list.

3.10.6 Using Rubber Band Recognition

Rubber band recognition enables operators to use the mouse to carry out field recognition in Completion and in Identification. Rubber banding can help optimize field validation since operators do not have to type field values. It can be used both on placed and not placed fields in Completion and on pre-indexed placed and not placed fields in Identification. An *OCR* engine must be associated to fields.

To test rubber band recognition, select **Test** > **Template Test** in Recognition Designer. Select a document from the Documents list in the left pane. Right-click and draw a rubber band on the value to read. The read characters appear in the input box. If you have not properly selected the value, use the rubber band tool again. Then press **ENTER** to confirm the value. You cannot test rubber band recognition by running a Unit test.

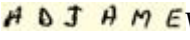
3.10.7 Multi-Engine Voting Recognition

Multi-engine voting improves recognition by allowing combination of several engine configuration files from the same or different engines. A voting mechanism is applied to keep the results with highest confidence level. Voting requires increased processing time. It is good practice to combine engines that are similar in terms of performance. For example, do not combine a slow engine with a fast engine or an accurate engine with an engine that reads less accurately.

There are different voting types that provide various results:

- **Pessimistic vote**: This method compares all of the lowest scores from all configuration files, and selects the best return. For example, if the lowest score is 41% for A is and 47% for B, then B is selected.
- **Optimistic vote**: This method compares all of the highest scores from all configuration files, and selects the best return. For example, if the highest score is 52% for A and 48% for B, the selected character is A.
- **Average score vote**: This method compares the average scores from all configuration files and selects the character with the highest average score. For example, if the average score is 46.5% for A and 47.5% for B, the selected character is B.
- **Global score vote**: This method calculates a global score from all the results for each engine configuration. The global score is the average of all the scores. The result retained is the best global score. This calculation mode applies even if the number of characters differs from one engine to another. In fact, the vote is done at the level of the result and not at the level of the character.

Example 3-3: Example of Multi-Engine Voting results

In this example, the first letter of the field is in question.  When running two engines on this example, the reading results are:

- Engine One reads A (confidence rate 52%) and H (confidence rate 48%).

- Engine Two reads A (confidence rate 41%) and H (confidence rate 47%).

The Multi-Engine Voting results are:

- A pessimistic vote outputs H, the higher of the low scores. The lower score for Engine One is H (48%) and for Engine Two is A (41%).
- An optimistic vote outputs A. the higher of the high scores. The higher score for Engine One is A (52%) and for Engine Two is H (47%).
- The average score vote outputs H with the higher average. For A the average is 46.5% and for H the average is 47.5%.

3.10.7.1 Understanding Segmentation Errors

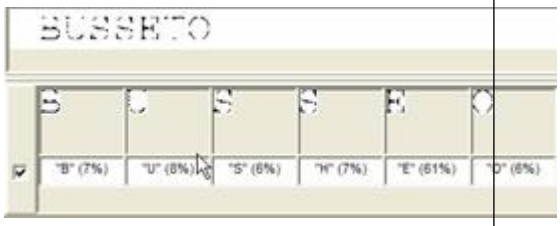
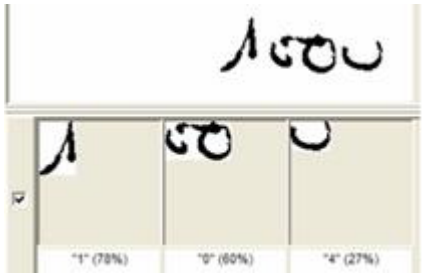
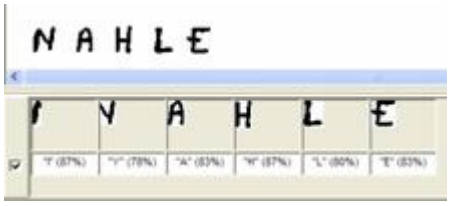
Segmentation errors arise when the recognition engines ignore characters, improperly combine adjacent characters, or separate a single character into two characters. The multi-voting option **Takes into Account Segmentation Errors** recognizes these segmentation errors and returns results from all selected engines. The potential results when using this option are:

- If the segmentation is the same for all engines, the voter algorithm is applied character by character.
- If the segmentation is different (for example, one engine finds two characters and another engine finds only one), the **Takes into Account Segmentation Errors** option finds the best segmentation and keeps it and the characters as the result.

Note: It is good practice to leave the **Takes into Account Segmentation Errors** activated. When off the voter algorithm is applied character by character even if the strings are not the same length based on the results from separate engine. If the strings are not the same length, at some point the compared characters are not the same due to shifting based on differing string lengths.

This table shows the different options available for pessimistic, optimistic, and average score votes, but not for global score votes.

Table 3-10: Example Segmentation Errors

Error	Example
Skipping characters	In this example, the character “T” is not returned as expected. 
Under segmenting	In this example, the two middle characters are not separated as expected. 
Over segmenting	In this example, the character “N” is separated into two characters. 

3.10.8 Defining Multi-Engine Voting Recognition

Multi-engine voting improves recognition by allowing combination of several engine configuration files from the same or different engines. This procedure gives information on defining the options for a typical multi-engine set up.

To create a configuration file for multi-engine voting recognition:

1. Open a project and then select **Tools** > **OCR/ICR Engine** > **New** > **Multi-Engine Voting**.
2. Name and define the settings for the multi-engine voting the engine configuration file.
   a. Select a voting method.

b. Select the option **Takes into Account Segmentation Errors** to recognize and correct for segmentation errors. This option applies to pessimistic, optimistic, and average score votes and not to global score votes.

c. In the **Select engines** pane, click the + button.

d. In the **Define an Engine** window, select an engine.

e. If necessary, specify an **Engine weighting coefficient** (set to *<1>* by default). For example, if using two engines and tests indicate that the first engine returns higher confidence rates than the second engine, weighting coefficients can correct for this difference. If the first engine returns confidence rates 20% higher than the second engine, apply a weighting coefficient of *<0.80>* to the first engine to decrease its confidence rates. Instead, the same results can be obtained by applying a weighting coefficient of *<1.20>* to the second engine to increase its confidence rate.

f. For images with noise or poorly scanned images, select one or more filters to clean up the image before recognition. For help on understanding and using filters, see "Applying Filters to Improve Recognition" on page 111.

g. Click **OK** to preserve the settings and **OK** in the **Mult-Engine Voting** window.

3. Select and place the field to apply multi-engine voting. For help on placing fields, see "Manually Positioning Index and Table Fields on a Template" on page 221.

4. On the **OCR engine** pane in the **Recognition** tab, select the multi-engine voting recognition file.

5. Select **Test** > **Unit Test** to perform a recognition test. If not satisfied with the settings of the engine configuration file, edit the file.

a. Select **Tools** > **OCR/ICR Engine** > **Edit**.

b. From the **Select Resources** window, select the **Local Resources** tab. Click the **Select** button to open the **Engine Edition** window.

c. Make the appropriate changes and click **OK**.

d. Run the test again to verify the results.

3.11 Templates

The topics in this section describe the operation on templates that can be performed in Recognition Designer.

3.11.1 Setting the Template Code

The template code facilitates the work of the operator in Identification. When several templates have the same fields to extract, assign them the same template code and use it to classify templates more rapidly in Identification.

If several templates having the same code potentially match the document (in other words, if several templates are in conflict), the operator enters the document code to classify the document and Recognition Designer automatically determines which template (among the templates in conflict) best matches the document. To create templates based on the same template codes, see "Creating Templates Based on Template Codes" on page 61.

To classify documents only by means of template codes in Identification, enable the **Document-code oriented keying** function.

To set the template code:

1. In the **Classification View**, right-click and select **Template Properties** from the template list. The **Template Properties** appears at the bottom of the main window.
2. Select one or several templates in the **Templates** list.
3. Type a template code in the **Code** field of the **Template Properties**.
4. Press **ENTER** to validate the code name.

Note: To help manage codes in a project, Recognition Designer enables importing new document codes. See the description of the function **Document Code Management** in section "Classification Tab" on page 279.

3.11.2 Copying Template Parameters

When the project contains similar templates, it is possible to copy template parameters (field positions, anchors and their parameters) and apply them to other templates in the project.

To apply the parameters of one template to another:

1. Click the **Index View** button in the main toolbar.
2. Select the template from which you want to copy the parameters.
3. Select **Indexing** > **Copy the Indexation Parameters** from the menu. The parameters are copied and memorized (no confirmation message appears).

   The **Copy the Indexation Parameters** option is dimmed if no template is selected or if the selected template has no associated index family.
4. Select the templates to which you want to paste the saved parameters.

5. Select **Indexing** > **Paste the Indexation Parameters** from the menu. The fields are created automatically and positioned at the same locations as in the template from which they were copied.

   The **Paste the Indexation Parameters** option is dimmed if no template is selected or if the selected template has no associated index family.

   If the template which the parameters are to be pasted to is not associated to an index family, the parameters are copied automatically without asking for confirmation. If the templates which the parameters are to be pasted to is already associated with an index family, the following warning message appears **Select Yes to replace the indexation parameters**.

Note: The parameters are kept in memory until the project is closed. Parameters cannot be copied from one project to another. A template can be imported from another project using the Importing Templates function.

3.11.3 Searching for Templates

To search for a template:

1. Click  from the toolbar or select **Classification** > **Search**. The **Search a Template** window appears.
2. Fill in the search criteria. For more information on searching a template, see section "Search a Template" on page 305.
3. Click **Reset** to clear the text boxes and type other criteria.
4. Click **OK** to start searching. Search results are displayed at the bottom of the main window.

3.11.4 Importing Templates

When several people create different templates at the same time for the same project, the import function enables updating the project with those templates. All the reference images of the imported templates must match the project resolution. Any template type can be imported. At the end of an import, the project is saved automatically.

Note: When using **Import Templates** feature, templates are imported without index families.

To import a template:

1. Select **File** > **Import templates**. The **Template Import Wizard** appears.
2. Select **Start**. The **Import Project** window appears.
3. Select the source project and select **Open**. A list of all the templates available in the source project appears.

- When selecting a source project, the **Display sub-directories content** option is checked by default, and all the project templates are selected for import to the root directory. The root directory is selected in the **Directory** list box.
- The Project path indicates the path and the source project name of the selected project. To select another project, click  .

4. Select the templates to import. Select the project directory in the **Project field**.
   a. To import templates from a single subdirectory, clear the **Display sub-directories content** checkbox then select a subdirectory. By default, all the templates of the subdirectory are selected.
   b. To import templates from different subdirectories select the **Display sub-directories content** option. Browse through the list of templates and clear the selection boxes for those templates you do not wish to import.
   c. To display the templates in a single subdirectory, select it from the **Directory** list box, where it appears represented by  .
5. Select the destination directory in the **Directory** list box. The default destination directory is the directory of the current project. Type a subdirectory name in the text box at the bottom of the window. This subdirectory is created in the current project directory and will contain the imported templates. If the target directory does not already exist, it will be created. The message "Target Directory will be created if it does not exist" appears. This means that if the templates reside in subdirectories in the source project, corresponding subdirectories will be created automatically in the destination subdirectory.
6. Click **Next** to begin the import. After the import is complete, the **Imported files field** displays all the elements that have been imported: the templates, the associated indexation families, the template associated resources and the keyword classification rules.

For help using the **Template Import Wizard**, see "Template Import Wizard" on page 405.

3.11.5 Naming Templates

To name a template:

1. In the **Classification View**, right-click a template and select **Template Properties** from the template list. The **Template Properties** appears at the bottom of the main window.
2. Select one or several templates from the **Template** list.
3. Type a template name in the **Name** box in **Template Properties**. When naming several templates simultaneously, the template names automatically inherit a suffix (for example, <*Invoice1*>, <*Invoice2*>).

3.11.6 Deleting Templates

To delete a template:

1. Click **Classification View** in the main toolbar.
2. Select the templates to be deleted.
3. Right-click the **Templates** list.
4. Select the **Delete Templates** menu.
5. Select **Yes** to confirm deletion of the selected templates.

3.11.7 Arranging Templates into Subdirectories

Project management can be greatly facilitated if templates are organized by category, although it is not mandatory to organize templates into subdirectories. The way templates are organized into directories has no impact on the document's classification.

To move templates to a subdirectory:

1. Open the project in Recognition Designer.
2. Select the **Index View** button on the Recognition Designer toolbar.
3. Right-click the project icon and select **New directory**.
4. Type the name of the directory.
5. Drag and drop the selected templates into the new directory.
6. Save the project.

3.11.8 Assigning Separators to Templates to Enable Folder Assembly

There are two types of separators available to assign to a template:

- **Natural separator**: A natural separator is used to define a document page as being the logical start of a folder. It is identified by  symbol in the templates list. A new folder is created each time a document matches a natural separator.
- **Artificial separator:** An artificial separator is used to define a patch document or a template which has been created specifically to fix the limit of a new folder. It is identified by  symbol in the templates list.

When a template has no separator assigned, then any document that matches the template automatically belongs to the same folder as the previous document in the batch. If no templates are assigned separators, then each batch will be have a single folder containing all documents in the batch.

To enable folder assembly using separators:

1. In Recognition Designer, click either **Classification view** or **Index View**.
2. Select one or more templates from the **Templates** list.
3. Select a separator type from the **Separator type** list.
4. Select **File menu** > **Project Options**. Select the **Folder Management** tab and select **Enable folder creation**.

3.11.9 Rotating Templates

It may be necessary to rotate templates to display them in a natural reading orientation on the screen. Automatic learning creates templates with the same orientation as the orientation of images at scan time, but these templates may not be convenient for reading or for creating templates in the **Index View**. Before rotating templates, ensure the project is not being modified or used by another person and that the template is not linked to an index family.

Note: The automatic detection of image rotation works only on graphical templates (that is, standard and *HPA* templates).

To rotate a template:

1. Click the **Classification View** button in the main toolbar.
2. Right-click the **Templates** list and select **Rotate the template**. Select the rotation direction (left, right or 180). A confirmation message appears indicating that the rotation is an irreversible operation and that the project will be saved automatically.
3. Compile the project as described in section "Compiling a Project" on page 44. Compiling is mandatory after rotating a template.

3.11.10 Exporting the Template List

Recognition Designer provides a list of all the templates in the current project together with the following information: template name, relative path, associated index family, template code, template frequency, total images (the number of images in the image base), size in *MB* of the image base.

To export the list of templates and template information:

1. Select **File** > **Project Options**.
2. In the **Project Options** window, select the **General** tab.
3. Click **Advanced information** in the **Information** panel.

   The list of templates with their parameters displays in a grid control.
4. To export the displayed list of templates, click the **Export template list** button.

5. In the system dialog box, navigate to the folder where you can save the file. Specify the file name and select the type of file: TXT, HTML, XLS, or XML.
6. Click **Save**. The grid with the list of templates is exported to the specified file.

3.11.11 Printing the Template List

Recognition Designer features a list of all the templates in the current project together with the following information: template name, relative path, associated index family, template code, template frequency, total images (that is, the number of images in the image base), size (in *MB*) of the image base.

To print the list of templates and available template information:

1. Select **File** > **Project Options**.
2. In the "Project Options" on page 276 window, select the **General** tab
3. Click **Advanced information** in the **Information** pane
4. In the **Advanced Information** window, click **Print template list**.

3.12 Setting Up Classification

The **Classification** tab of the **Project Options** window enables setting up the behavior of the Classification module in production.

To define Classification settings:

1. In the **Project Options** window, select the **Classification** tab.
2. For graphic classification (standard and *HPA* templates), identify the rotation to apply to inverted or rotated images. Select accordingly from the **Engine options** pane:
   - **Test 180 rotations**: Rotate the scanner output image by 180 if it is upside down.
   - **Test 90 and 270 rotations**: Change the orientation of the scanned images from landscape to portrait, rotating front and back pages by 90 or 270 as required to achieve correct orientation.
   - **Test side flipping**: Check the order of the scanned images and display the two sides in the correct order.

     Note: Classification processing time can increase 10-40% when detecting and correcting image orientation.
3. For standard templates, adjust the pre-classification and decision rates. Click **Advanced engine parameters**. In the **Advanced parameters for classification engine** window, adjust the following settings:
   - Pre-classification: Set the pre-classification threshold for standard templates.

- Decision: Set the decision threshold to be applied to the successful templates.

Note: Keeping to the default parameters is highly recommended.

After adjusting either of these two thresholds, check the resulting pre-classification and decision rates by running a classification test.

4. For HPA templates, set the **HPA Default Search Zone** settings:

   - **H (mm)**: the height of the search zone applied to HPA anchors when placed on an HPA template
   - **W (mm)**: the width of the search zone applied to HPA anchors when placed on an HPA template

   Once the anchor is placed, the search zone can be modified.

5. For text matching classification, select **Enable text matching classification** to enable text matching settings.

6. Select the default template for the project with the option **Assign a template by default (if not recognized)**. Assign a default template to all non-classified documents to avoid sending the document to manual classification. Select the template from the drop down list.

7. For classification of handwritten documents, select the option **Assign handwritten documents to template** to define the template to which all handwritten documents are to be classified with. Select the template from the drop down list.

8. To select the fields to be read during the classification steps, select **Pre-index following fields** under **Pre-index**. These fields are known as pre-indexed fields in Recognition Designer. Table fields cannot be pre-indexed.

   a. Select the **+** button from the **Pre-indexing** pane.
   b. In the **Pre-Indexed Fields** window that displays, select fields from the available fields and click the **Add** button. The available fields are grouped by index family. As soon as a field is added to the list of **Selected fields**, it no longer appears in the list of available fields.

      Note: Fields to be pre-indexed are available as folder binding fields. The folder binding field is the index field which initiates the creation of a folder during classification. To split the document flow into folders, a comparison is made between the values read on the different documents. A folder is created whenever the folder binding field value on a document is different from the previous document, or when two documents have the same folder binding field value but different template codes. This option is most useful to combine multiple-page invoices based upon their invoice number.

   c. To enable folder assembly using a folder binding field, type the index fields to be pre-indexed in the **Pre-index following fields** text box. These fields

must be predefined so they can be selected in when enabling the folder binding field.

9. **Document code management** lists the template codes defined in the project and enables creating and importing new codes. The digit in the **Used** column indicates how templates use a code in the project. Understand how template codes are intended to be used in Recognition Designer in the section "Achieving Business Logic with Template Codes" on page 65.

   - Click **+** to create a code. The template code must not exceed 30 characters.
   - To remove codes, select the codes and click **-**.
   - If one of the codes to be deleted is used in a template (1 in the **Used** column), then a window prompting for confirmation is displayed. Click  to edit a code
   - **Import codes**: Imports a list of codes from a `TXT` file containing one code per line. Imported codes are added to any existing codes.

     For Unicode support, the Project Designer must select the encoding format of the `TXT` file to load when importing data in Recognition Designer or Free Form Designer:

     – Autodetect
     – ANSI
     – UTF-7
     – UTF-8
     – Unicode (UTF-16 Little-Endian)
     – Big-Endian (UTF-16 Big-Endian)

10. The **Compilation cache** panel settings optimize the time it takes to compile a project. Select **Activate** to initiate, and click **Empty** to empty the cache.

11. If you want to use the OCR data cache instead of running full OCR on PDF and PDF/A documents and image files, then select the appropriate options on the **Standard OCR** tab.

Notes

- In Recognition Designer, PDF and PDF/A pages are displayed and processed as images. The resolution of these images is determined by the project resolution. If the project resolution is not defined or the project contains only generic templates, then the resolution for images is 300 DPI.

Property	Description
Enable OCR data cache from Standard OCR	When this option is selected, the following behavior for PDF and PDF/A documents (converted from Microsoft Office documents and original PDF and PDF/A documents only) and images is enabled (except for the specified in **Select OCR engines to use instead of Standard OCR cache**): • Classification When performing textual, keyword, and text matching classification on PDF pages and images, the text (including coordinates and line/word separation) in the OCR data cache is used. In addition, PDF pages are not converted to images, which could result in better performance. **Note: Project Options** > **Recognition** > **Free Form image filters**, **Project Options** > **Text matching image filters**, and **Indexing** > **Index View** > **Image Clean Up** filters are skipped. • Identification and Completion For a PDF page, rubberbanding uses the OCR data cache. **Notes** – Annotations are disabled. – The PDF page cannot be rotated. • Extraction When performing extraction on PDF pages and images, zone and free-form recognition uses the OCR data cache. In addition, if PDF pages are not converted to images, then better performance could result. If the following elements do not exist on the page, then the PDF page is not converted to an image: – Table fields – Graphical anchors However, if the aforementioned elements do exist on the page, you could ignore them as follows:

Property	Description
	– To ignore graphical anchors, enable the **Disable graphical anchors for zonal fields** option. – To ignore table fields, disable assignment of table fields.  **Note: Project Options** > **Recognition** > **Free Form image filters** and **Indexing** > **Index View** > **Image Clean Up** filters are skipped.
Apply text-based classification before graphical classification	With the OCR data cache, text-based classification (Textual, Keyword, and Text-Matching) could be faster than graphical classification. If the aforementioned is true, then select this option to improve performance.
Disable graphical anchors for zonal fields	Select this option to disable graphical search for zonal index or table field anchors, which prevents the loading and rendering of these input pages. Thus, enabling this option could result in improved performance. In addition, if a text value is assigned to the anchor, then the anchor is determined by the OCR data cache.
Select OCR engines to use instead of Standard OCR cache	Specify the OCR engine to use in place of the OCR data cache. The following images on PDF pages can only be recognized by OCR: • Barcode images • US and/or French check images • Checkboxes In addition, if PDF pages are composed of images only and free-form rules are optimized for a particular OCR engine, then the recognition accuracy of the OCR data cache would be reduced.

3.13 Setting up Text Matching

The **Text Matching** tab of the **Project Options** window serves to specify the parameters of text matching classification. This tab is also available from the Text Matching Designer.

To define text matching options:

1. Specify the options in the **Image filters** panel:

   - **Reverse video zones**: Use this setting to detect reverse video text boxes in the image and changes their background from black to white and the characters from white to black so that they can be read by *OCR* engines.
   - **Matrix font**: Use this setting to detect automatically the presence of matrix characters on the image and make them bold so they are better read by OCR engines.
   - **Table lines**: Use this setting to delete all horizontal and vertical lines in the image. This filter is recommended for most projects whenever table characters overlap or touch the table borders.
   - **Shaded areas**: Use this setting to detect all the shaded areas in the image and removes the shaded background (by removing pixels) without altering the other areas of the image. This filter is not recommended when shaded areas contain thin or very thin characters.
   - **Text box reading**: Use this setting to specify the segments the whole page into individual text lines that are then passed individually to the OCR engine.

2. Specify the **OCR engine** options:

   - **OCR engine**: Use the setting to select the OCR engine configuration file used for text matching. Click the button to the right of the text box to open the **Select Resources** window to select engine configuration files (`*.reco`) either from the **Global Resources** tab or from the **Local Resources** tab.
   - **Engine confidence threshold**: Use this setting to define the confidence threshold value for the **OCR engine**. If a character obtains a confidence level value less than the confidence threshold value, the character is not recognized.
   - Browse (**...**): Browse for a directory and select an OCR engine configuration file.

3. Specify the options in the **Parameters** panel:

   - **Matching threshold**: matching rate from which Recognition Designer considers that a given document is classified.
   - **Minimum difference between two best candidate results**: If the difference between the two matching values exceeds the **Minimum difference**

between two best candidate results value, then the first template is selected. If there is not enough difference, the image is set in conflict.

- **As a second attempt, lower the matching threshold by**: If the best match is lower than the matching threshold and both candidates are associated with the same template, a further test takes place to determine if the first candidate template rate is higher than **Matching Threshold** minus **As a second attempt, lower the matching threshold by**. If yes, the first template is selected. Otherwise the image is set as unclassified.
- **Advanced**: A set of parameters used to speed up the processing.
- **Default**: Reverts back to the initial default values for all text matching parameters.

3.14 Setting Up Classification Edit (Deprecated)

The **Classification Edit** tab of the **Project Options** window enables setting up the behavior of the deprecated Classification Edit module in production.

Notes

- This option is always enabled in the Completion module. Although you cannot disable it in the Completion module. You can use another feature in Dispatcher Manager to disable this option for the Completion module.
- For Identification, there are other options that allow the automatic rejection of a folder containing a document rejected by an operator and the definition of a custom hotkeys for rejecting documents.

To define Classification Edit settings:

1. In the **Project Options** window, select the **Classification Edit** tab.
2. To set **Application options**, select from the following options:
   - **Reject folder if one of its document is rejected**: Rejecting a document rejects the whole folder that contains the document.

     Note: The **Reject document** option under **Keyboard shortcut setting** is used to allow the operator manually reject documents.
   - **Display active folder thumbnails**: Displays all the images of the current folder as thumbnails in Classification Edit. Otherwise, only current image is displayed.
   - **Show only Generic template in template list**: Shows only generic templates in the list from which the operator can pick up a template to classify the document.
   - **Document-code oriented keying**: Forces operators to use template codes exclusively to classify documents in Classification Edit. When enabled, an operator can type only template codes in the identification zone of Classification Edit; the list of suggested templates only displays template

codes. Otherwise, operator can type template codes or names in the identification value zone and the list of the suggested templates displays both template codes and template names.

- **Display next document automatically**: Available in conjunction with **Document-code oriented keying**. If selected, as soon as a valid template code has been attributed to a document in Classification Edit, the system automatically goes to the next document to be classified. If cleared, the operator must press **ENTER** to go to the next document to be classified.
- **Go to next field automatically**: Applies to fields for which the option **Character validation** is selected in the **Index Family**. If selected, a field is automatically validated if no characters are in error in Classification Edit. Use the **Enable/disable go to next field automatically** from the **Keyboard shortcut setting** to toggle this setting on and off.
- **Confirm closing session after task completed**: Opens a confirmation window at the end of each task asking the operator to close the session.
- **Default zoom**: Sets a default zoom level to display images in Classification Edit. Sets also a default zoom level for viewing pre-indexed non-placed fields. Each time an non-placed field is selected, the default zoom will be applied.
- **Persistent zoom**: Enables overriding the **Default zoom** setting, by defining a new zoom setting during production. Each time a new zoom setting is specified, it will become the new zoom default for the current document until a new setting is specified or the next document is opened. The first pre-indexed field selected in a document always uses the **Default zoom**.
  - When **Persistent zoom** is activated, the current zoom level is used. In other words, if you reset the zoom level for a field, that becomes the active zoom level while viewing fields in the current document. This new zoom level persists until another zoom level is defined or the next document is opened.
  - When **Persistent zoom** is not activated, each selected field uses the **Default zoom** setting.

3. Define **Keyboard shortcut settings** as required:

- **Help display**: Displays product documentation.
- **Default zoom**: Displays the **Default zoom** which value is specified under **Application options** on the image.
- **Reject document**: Manually rejects documents that cannot be processed in Classification Edit.
- **Rotate to the right** : Rotates documents to the right.
- **Hide field value**: Used to hide values.
- **Rotate to the left**: Rotates documents to the left.

- **Enable/disable go to next field automatically**: Toggles the automatic display of the next field on or off.

4. Set **Color settings** as appropriate to aid visual recognition of folders (even and odd ones) and fields in Classification Edit.

Chapter 4

Recognition and Extraction

Extraction is the process of recognizing and extracting data from documents. It includes a combination of specifying the type of data to extract and selecting the appropriate recognition engine.

To configure extraction in Dispatcher Manager, designers create index families that are associated with classification templates.

An index family is a set of index fields (for example, single types of data such as invoice number or amount) or table fields (for example, data in tables such as a number of ordered items) to be extracted from a specific type of document, such as an invoice. Each field is customized for the type of data to extract and then the index family is assigned to the classification templates in the indexing view. The fields are displayed for positioning on the template image. Extraction requires that index families are associated with project templates that are representative of the documents in a project. Templates are defined during classification setup which creates a separate template for each document type. Once a representative set of templates is defined, index families must be associated with the templates.

Dispatcher Manager can perform both zonal and free form recognition.

- *Zonal recognition* defines specific areas on a document where the same data field always resides. This type of document is considered a structured document. For example, specific tax forms or medical forms are structured documents and always contains the same specific information in the same location. Zones define data retrieval from exact locations on the documents.
- *Free form recognition* is used for semi-structured or unstructured documents where data retrieval is from different locations on different templates. For example, invoices from different companies are often structured differently and would be assigned to different templates. To extract data from these invoices, recognition must be flexible and setup of index families alone is not always sufficient. This situation often requires a definition file created with the module Free Form Designer and assigned to the index family. Free form recognition uses full page recognition, and location of data is determined in relation to keywords and associated words on the form.

4.1 Choosing the Data Extraction Technologies

After documents are sorted into structured, semi-structured, and unstructured groups, choose the data extraction technologies that are appropriate. The Extraction module features two technologies for data extraction:

- **Zonal recognition**: Uses zones and anchors to locate and extract fields from predefined areas of the documents. Zonal recognition is recommended for structured documents. This example illustrates the various features included in Extraction such as reading checkboxes, tables, signatures, or barcodes.
- **Free form recognition**: Uses keywords to locate and extract fields when they are not in predefined areas of the documents. Free form recognition is recommended for semi-structured and unstructured documents.

Document Data Analysis

Data extraction includes retrieving and extracting predefined data items from documents. When designing data extraction in Recognition Designer, analyze the following elements:

- **Fields**: On structured documents, fields are typically always the same and always in predefined areas in all documents. In semi-structured documents, fields may vary between documents and are not always in the same areas. In unstructured documents, fields always vary and are never in the same areas.
- **Document classes**: A document class is a collection of data items to be extracted that is common to several documents, even if they are graphically different or originate from different sources or vendors.

  For example, analyzing the production flow reveals that it contains two document classes: invoices and medical forms. First, define the collection of fields to be extracted from invoices such as invoice number, invoice date, amount, and vendor name. Then, define the collection of fields to be extracted from medical forms such as patient name, clinic name and address, and patient *SSN*.
- **Index families**: A collection of fields (data items) to be extracted for a given document class. The advantage of an index family is that even if there are hundreds of different graphical layouts belonging to the same document class, one index family is used for all the layouts of the same document class. For example, to handle two document classes, such as invoices and medical forms, create two index families: The “invoice” index family contains all the fields to be extracted on all invoices. The “medical form” index family contains all the fields to be extracted from all medical forms.

  Index families are composed of index and table fields for extracting data from documents during production. Index family scripts can further customize the behavior during production. When defining index families, operators create fields and assign properties to fields that specify the extraction data characteristics. Those fields are then associated with and placed on templates before production.

The number of index families needed for a project varies based on the type of documents being processed. When working with structured documents, such as strictly formatted forms, each template is unique and a separate index family is necessary for each template. Index families for structured documents are precise and inflexible, since data always appears in the same location on all documents assigned to a specific template.

Index families for semi-structured or unstructured documents are defined to locate information based on search techniques. The data does not have to appear in the same location on each template. A single index family can be appropriate for several semi-structured or unstructured templates, so fewer index families than templates are needed.

Index families can also use scripts that contain functions and events calling object properties. Dispatcher Manager features a complete integrated development environment. "Creating or Editing an Index Family Script" on page 227 provides information on creating scripts for index families.

Data Extraction Templates

Based on the above analysis, decide which templates are needed for data extraction: graphic templates or free form templates. Most projects require combining both types of templates to achieve business requirements. Find examples of business requirements with their most appropriate template types in the section "Examples: Addressing Business Requirements" on page 33.

When deciding on the best combination of graphic and free form templates, consider the following:

- Creating graphic templates is fast as it is facilitated by the *automatic learning feature*. Creating the index families can also be quickly accomplished because the same index family is used for all the templates (graphic or free form) belonging to the document class. More time-consuming is positioning the fields to be extracted with zonal recognition on the predefined areas where they appear in the graphic templates. The time required to position fields is proportional to the number of graphic templates.
- Creating a free form template is fast as you create one template for each document class. Creating free form rules for each field extracted with free form recognition is more time-consuming. Depending on the complexity, designing the free form rules for one field can take up to one day. This initial investment is compensated by the fact that the same free form rules can process an infinite number of different documents in production, so long as they belong to a document class for which free form rules exist.

Processing Speed

Processing is much faster with graphic templates than with free form templates. This is because free form recognition requires full-page *OCR*, recognizing all the data on the entire page. Full-page OCR takes 1 to 10 seconds per image depending on the

number of characters in the image, the complexity of the image layout, and the *CPU* speed.

4.2 Zonal Recognition

Zonal recognition uses *OCR* to recognize text in predefined zones (placed fields) on structured documents. When creating a recognition zone for a field, set the recognition properties for that field in Recognition Designer when placing the field on the template.

Zonal Recognition and Structured Documents: An Example

Figure 4-1 illustrates a simplified dental claim with data extraction on predefined zones in a structured document.

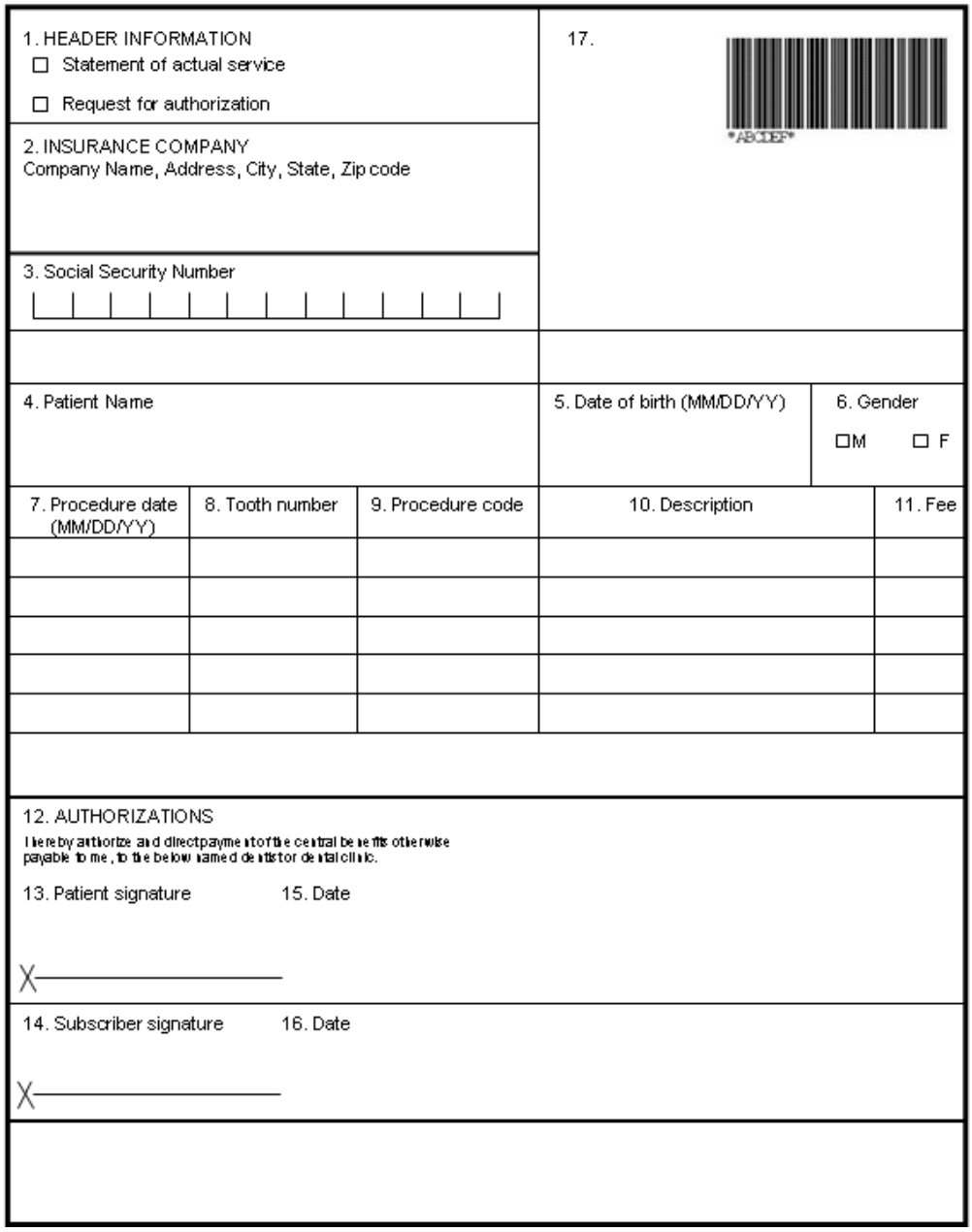


Figure 4-1: Data extraction on predefined zones

Checkboxes - Items 1 and 6 in the Sample Form

Learn on checkboxes detection from the topic "Detecting Marked and Unmarked Checkboxes" on page 141.

Multiple-line Address Block - Item 2 in the Sample Form

The two options are:

- Use zonal recognition with a script to extract the sub-fields from the main field:
  - Create a field called “Address” which you place over the whole address block.
  - Set this field as **Hidden** so that it does not display when performing a template test.
  - Associate this field with a full text OCR engine to recognize multiple lines.
  - Create all the other fields contained in the address block: Company name, Street, City, State, Zip code. Place all these fields on the form to cover the whole address block but do not assign any OCR engine to these fields.
  - Create a script in the index family to retrieve the OCR engine output for the field “Address”: send the first line as input to the field Company name, the second line as input to the field Street, and so on with the other lines. You can detect the zip code (5 digits) and send it to the Zip code line.

  If you want to reuse this script on graphic templates of a different document class (so which have another associated index family), you can reuse the script by means of the include statement `#Uses` (learn more in the topic Using Script Inclusions).

  Note: Another solution consists in keeping the fields unplaced but retrieving the coordinates of the bounding boxes of `DpDocField` objects by means of the `SetBounds` method. For more information on this method and object, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.
- Use a free form rule which you can associate with the field:
  - Create as many fields as there are fields to extract from the address block: Company name, Street, City, State, Zip code.
  - In Free Form Designer, create as many full text fields as there are fields to extract for the address block: Company name, Street, City, State, Zip code.
  - In Free Form Designer, create the free form rules to detect and extract the data. Save the rules to the definition file (.dft).
  - In Recognition Designer, associate each field with the *DFT* file.
  - In Recognition Designer, associate a full text OCR engine to the fields. Or you associate a full text engine to only one of the fields and reuse the OCR output to recognize all the other fields and reduce processing time.

Grids - Item 3 in the Sample Form

This grid is for the 9-digit *US* Social Security Number. To detect the *SSN* follow these steps:

- Create a field and place it to cover the entire grid where the SSN is to be found.
- In the index family, select **Fixed format** and enter the format 9N (meaning nine numerical characters) and clear **Partial value accepted**. If the field value does not comply with this format, the value will display in the Completion module for the operator to fix it.
- Select an OCR engine that can read numerical characters.
- Apply an adaptive filter to the field. The filter deletes the grid and thereby improves character recognition. Learn more on filters in the topic Applying Filters to Improve Recognition.

Machine Printed Alphabetical Characters - Item 14 in the Sample Form

To extract the patient name as in the sample form, follow these steps:

- In the **Index Family Editor**, create a new field. Select **text** as field type. Select an OCR engine that can read alphabetical characters.
- Place the field onto the template to cover the area where the patient name is to be found.

Dates - Items 5, 15 and 16 in the Sample Form

To extract a date, follow these steps:

- Create a field and place it to cover the area where the date is to be found.
- In the index family:
  - Select **date** as field type.
  - Select a date format. In the sample form, the format is DD/MM/YY. If the date happens to have another format in a document, this document displays into the template test window for the operator to change the date format to the selected date format.
  - Select an OCR engine that can read numerical characters.

Table Data - Items 7, 8, 9, 10 and 11 in the Sample Form

To extract table data, follow these steps:

- Create an array field for each column to extract from the table.
- Place each array field to cover entirely each column.
- Select the parameters.

When table data is extracted, the table is reproduced in column and row format matching the original table. Table cells that cannot be retrieved display in the template test window for the operator to complete them. When a table line is not recognized (missing line) the operator can add the line and then enter each cell value manually.

Signature - Items 13 and 14 in the Sample Form

To detect whether the document bears or a signature or not, follow these steps:

- Create a field and place it to cover the entire area where the signature is to be found.
- Select the engine Modification Detection.

Note: Use this engine also to detect modified pre-printed characters. For example, a customer may strike a pre-printed address and handwrite a new address.

Barcode - Item 17 in the Sample Form

Recognition Designer recognizes all types of barcodes whether barcodes appear in predefined areas in the document or in undefined areas in the document.

4.2.1 Machine Printed Zones

Machine printed alphanumeric characters have consistent, predictable shapes and fonts. When the information to extract is in the same location on all documents, fields can be placed on the template. Then select an engine appropriate for machine printed characters and zonal recognition. For information helpful in selecting the most appropriate engine, see "Recognition Types Supported by Recognition Engines" on page 429.

To define recognition for machine printed zones:

1. Place a field on the image.
2. In the **OCR engine** pane in the **Recognition** tab, select an engine recognition file whose type is machine printed (machine printed appears in the column **Type** in the **Select Resources** window) and check the description of the engine that displays in the **Description** column of the **Select Resources** window to select the engine whose characteristics fit the hand printed zone.
3. Perform a recognition test by selecting the menu **Test** > **Unit Test**. If not satisfied with the settings of the engine configuration file, either customize the file or create another one.

Note: Extraction features a voting engine that is able to detect whether characters are hand printed or machine printed to apply the appropriate recognition engine to the field.

4.2.2 Hand Printed Zones

Hand printed recognition, also referred to as Intelligent Character Recognition (*ICR*), recognizes alphanumeric characters that are hand printed or otherwise not precisely consistent in size and shape from one character to the next.

To define recognition for hand printed zones:

1. Place a field on the image.
2. On the **Recognition** tab in the **OCR engine** pane, select an engine recognition file whose type is Handwritten (Handwritten appears in the column **Type** in the **Select Resources** window) and check the description of the engine that displays in the **Description** column of the **Select Resources** window to select the engine whose characteristics fit the hand printed zone.
3. Perform a recognition test by selecting the menu **Test** > **Unit Test**. When the test returns poor results, modify the engine configuration file settings to obtain better results, or create another one.

Note: Recognition Designer features a voting engine that is able to detect whether characters are hand printed or machine printed to apply the appropriate recognition engine to the field.

4.2.3 Modification Detection for Handwritten Notes or Signatures

Modification Detection detects handwritten modifications made to pre-printed forms. This detection type detects the presence of information in an empty field, such as the presence of a signature at the bottom of the document.

To detect handwritten notes:

1. Create a new field. For help creating fields, see "Creating or Modifying Index Fields and Table Fields" on page 210.
2. Place a field on the zone that contains pre-printed characters.
3. Select the menu **Tools** > **OCR/ICR Engine** > **New** > **Modification Detection**.
4. Give a name to the engine configuration file.
5. Specify the size parameters:
   - For handwritten notes, only the **Minimum height**. It should be slightly higher than the size of pre-printed characters in the zone. If the handwritten notes found in the zone are greater in height than the field placed on the pre-printed characters, then the engine considers that handwritten notes have been added onto the pre-printed characters.
   - For signatures, specify the **Minimum height** and **Minimum width**. For **Minimum height**, it is advised to specify a height of 10 mm. If the shape

detected in the zone is of a greater size than the **Minimum height** or the **Minimum width**, then the engine indicates the presence of a signature.

6. Leave the **Minimum connected shapes required** parameter set to *<1>*.

7. It is recommended to apply a dilatation filter to thicken fine or eroded characters and make it easier to detect a signature. For help using filters, see "Applying Filters to Improve Recognition" on page 111.

4.2.4 Detecting Hand Printed or Machine Printed Characters

Extraction features a voting engine that detects whether characters are hand printed or machine printed. Figure 4-2 illustrates the portion of a form containing both hand printed and machine printed characters:

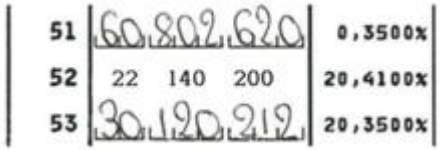


Figure 4-2: Machine and hand printed characters

In this form, line 52 is pre-filled with machine printed characters whereas lines 51 and 53 are filled with hand printed characters. To determine the hand printed versus the machine printed characters, a voting engine can compare the size of the characters in the three lines. When detecting hand printed versus machine printed characters, **Max Type Height** and the **Max Type Width** values are defined that set the maximum size expected for machine printed characters. The assumption is that hand printed characters will be larger than machine printed characters. Since characters in lines 51 and 53 are greater in height and width than the machine printed characters (with appropriate **Max Type Height** and the **Max Type Width** specified values) the voting engine considers that these characters are hand printed characters.

It is possible to have a combination of hand printed and machine printed characters in the same line. If this is likely to occur, set the **% minimum of machine printed characters** value to 100 to allow a mix of hand printed and machine printed characters in the same line.

Note: It can be useful to apply an adaptive filter, such as **AdaptiveFilter1.mask**, to delete pre-filled characters, such as the character delimiters in this example.

To create a configuration file to vote between hand printed and machine printed characters:

1. Select **Tools** > **OCR/ICR Engine** > **New** > **OCR/ICR Voting**.

2. Give a name to the engine configuration file.

3. Select a hand printed engine as “Engine 1” and a machine printed engine as “Engine 2”. Type a confidence threshold in the field to the right.
4. Select the **Max Type Height** and the **Max Type Width**. Characters found to be higher or wider than these values are considered to be hand printed.
5. Set the **% minimum of machine printed characters** (minimum percentage of machine printed characters). This is the percentage of characters which are higher or wider than the specified values (i.e., the values specified for **Max Type Height** and the **Max Type Width**) that a field must contain for the characters to be considered as hand printed characters.
6. In the main interface of Recognition Designer, select the field to which you want to apply this voting engine.
7. Place the field on the index image. For help on placing fields, see “Manually Positioning Index and Table Fields on a Template” on page 221.
8. On the **Recognition** tab of the **OCR engine** pane, select the engine recognition file you created. From the **Select Resources** window, in the **Local Resources** tab, select the engine configuration file that you have created for voting between hand printed and machine printed characters.
9. Apply an adaptive filter to the field to delete pre-filled characters. For help on applying filters, see “Applying Filters to Improve Recognition” on page 111.
10. Perform a recognition test: select **Test** > **Unit Test**. If not satisfied with the settings of the engine configuration file, edit the file and adjust the settings.
    a. Select the menu **Tools** > **OCR/ICR Engine** > **Edit**.
    b. From the **Select Resources** window, select the **Local Resources** tab that contains the multi-engine configuration file which you have created. Click the **Select** button to open the **Engine Edition** window.
    c. Make the appropriate changes and click **OK**.
    d. Run the test again to verify the results.

4.2.5 Detecting Marked and Unmarked Checkboxes

The `Basic_OMR.reco` recognition engine can determine the fill rate (the percentage of black pixels inside a checkbox) of the checkbox. This indicates whether a checkbox is marked (checked) or unmarked (unchecked), or that an ambiguous fill rate is present. In other words, checkbox marking is determined by the percentage of black pixels inside the checkbox boundary. This determination is based on both positive and negative percent values. For example, setting <5> in the **Positive percentage** option means that when more than 5% of the pixels inside the checkbox are black, the checkbox is considered checked (this returns a value of “1”). Setting <*1*> in the **Negative percentage** option means that when less than 1% of the pixels inside the checkbox are black, the checkbox is considered unchecked (this returns a value of “0”). If the percentage value of black pixels is between 1% and 5%, the mark is ambiguous (this returns “?”).

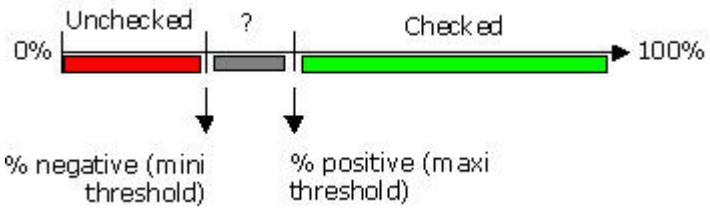


Figure 4-3: Relationship between selected and cleared checkboxes

To create a zone for a single checkbox on the index image:

1. Create a field for each checkbox to be evaluated.
2. Place the field on the checkbox to be read on the index image. You need a field for each checkbox to be evaluated. Place the field at a maximum of 2 millimeters from the checkbox border. Options are:
   - If the form is printed with inactinic ink, frames around checkboxes do not appear on the scanned images. You do NOT need to apply an adaptive filter.
   - If the form is NOT printed with inactinic ink, frames around checkboxes appear on scanned images and prevent correct reading by the OCR engine: recognition returns 1 even if the checkbox is not marked. In this case, you must apply adaptive filters to the fields. In the **Index View**, select the field on which to place the adaptive filter. Then select the **Image Clean Up** tab and one of the two adaptive filters. An essential condition for the adaptive filters to work properly is that the reference image be a blank form, that is a form without nothing filled out in the text boxes. This blank form is used as a mask for the completed forms. If you do not have a blank form, use any image editing software to remove text from the form and create a blank form. Recognition Designer does not include any image editing software. The exception is Imaging software that is available if you are running Recognition Designer with Windows NT4 or Windows 2000. In this case, right-click the template and select **Open Index Image with Imaging**.
3. On the **Recognition** tab of the **OCR engine** pane, select the engine configuration file `Basic_OMR.reco`. This is the engine configuration file supplied by default with Recognition Designer.
4. Perform a recognition test by selecting the menu **Test** > **Unit Test**. If not satisfied with the settings of the engine configuration file, customize the file or create another one. To do so:
   a. Create a new engine configuration file.
   b. From the main interface of Recognition Designer, select the menu **Tools** > **OCR/ICR Engine** > **New** > **Basic OMR**.
   c. Name the engine configuration file.
   d. Specify the **Positive percentage** and **Negative percentage** values used to set the matching threshold of the engine. The engine analyzes the

percentage of black pixels in the checkbox, that is, the fill-rate of the checkbox. The positive value must be greater than the negative value.

e. Leave the **Without skeleton** option checked. This option is preserved for ascending compatibility.

4.3 Table Data

Table data is displayed on images in a table format, composed of rows and columns of data. The data can be of any type, and may or may not have descriptive headers or section identifiers used to separate data based on grouping (by date, for example). Recognition Designer enables extraction of data that preserves the table, and can be set up to customize the way the data is presented in final output, so that groupings are preserved, for example. It is important to understand the Recognition Designer terminology so that creating table fields for extracting data is clearer.

Note: Table fields are supported for Advanced Recognition licensing only.

To extract table data from an image requires definition of table fields as part of an index family. Table fields are created for each column of data to extract from a table. Each table cell encompassed by a table field is called a line. When data is extracted, the table is reproduced in column and row format matching the original table.

Note: In Recognition Designer, the detection of table lines is a process that is separate from the recognition process. In other words, lines are detected even if no recognition engine (that is, an engine configuration file) has been associated with the table fields. If a line is detected but no engine configuration file is specified, a blank line will be returned.

Another important aspect of table data is the use of headers and how to represent and reproduce them as extracted data. In Recognition Designer, groupings of data separated by header lines are called paragraphs. Groups of data, or paragraphs, can be distinguished in the output by extracting and reproducing header information. When setting up table fields for tables grouped using headings, a single table field must be created to detect paragraph headers. Only one paragraph header table field can be created for a single Index Family. Paragraph headers are recognized based on keywords to search for in a table. Unlike column table fields, paragraph table fields are drawn to encompass all header information in the table. This can be the size of the entire table, but that is not always necessary.

Using the simple example given in the table below, consider these points:

- Four table fields would be created to extract the data in the four columns (for the Item Number, Quantity, Price, and Total columns). In the index view of Recognition Designer, each table field would be drawn to encompass a single column of data.
- One paragraph table field would be created to capture the date headers. A keyword of “Date” can be assigned to this table field, so the rows containing the word “Date” can be recognized as the paragraph headers.

- An additional table field can be created that will capture the actual dates, and is drawn to encompass the actual dates in the table. This can also be set up to write the date in a new column in the output.

Table 4-1: Data Table Example

Item Number	Quantity	Price	Total
Date: 21/05/2009			
10112	2	$5.00	$10.00
10114	4	$7.50	$30.00
Date: 28/05/2009			
10112	4	$5.00	$20.00
10114	1	$7.50	$7.50
10116	3	$2.00	$6.00

Using the simple points indicated, the returned values might be organized in this configuration. However, this is only one of several possibilities. For more details in setting up table fields, see "Table Recognition: A Simple Example" on page 144 and "Table Recognition: A Complex Example" on page 146.

Table 4-2: Returned Values from the Data Table Example

21/05/2009	10112	2	$5.00	$10.00
21/05/2009	10114	4	$7.50	$30.00
28/05/2009	10112	4	$5.00	$20.00
28/05/2009	10114	1	$7.50	$7.50
28/05/2009	10116	3	$2.00	$6.00

4.3.1 Table Recognition: A Simple Example

This topic describes the method to process simple tables in structured documents. This procedure also describes how to process tables where some lines (cells) have missing data.

Note: In Recognition Designer, the detection of table lines is a process that is separate from the recognition process. In other words, lines are detected even if no recognition engine has been associated with the table fields. The recognition of table data is performed after the lines have been detected.

To detect table lines:

1. Create the table fields by using Intelligent Capture Designer to create a document type in which you create the table fields and specify their validation properties.

Note: For Dispatcher Manager, use the Index Family Editor to create an index family in which you create the table fields and specify their validation properties.

2. Specify the document type as an index family for your project template.

   See "Manually Positioning Index and Table Fields on a Template" on page 221.

3. In the **Index View**, place the table fields on the table in the template: place one table field on each column to cover the whole column height and width.

4. Perform a template test to check whether all lines are detected or not. For help performing a template test, see "Performing a Template Test" on page 200. If one or several lines have not been detected, improve the line detection for each table field by updating their properties in the **Index View** as follows:

   - Apply the following filters on the **Borders** tab as appropriate:
     - Select **Right/left border detection** and/or **Top/bottom border detection** to improve line detection using the vertical and horizontal borders that separate columns and lines.
     - Select **Line Removal** to filter out detected border lines. If several table lines (cells) have not been detected, check whether they are separated on the document by horizontal borders or not. Horizontal borders may prevent correct line (cell) detection.
   - Apply the following properties on the **Lines** tab as appropriate:
     - For **Min** and **Max** values, enter the minimum and maximum line height required for the detected line to be displayed. By default, **Min** is set to *<15>* and **Max** is set to *<50>*.
   - If the table contains columns with empty lines, choose either to detect only lines having values in each field or the lines including those with missing values by selecting or clearing the **Detect all lines** option in the **Lines** tab.
     - When this option is cleared, only lines having values are detected.
     - When this option is selected (by default this option is selected), all lines including those without a value are detected.

   Note: Even if the **Detect all lines** option is selected, empty rows in a table are still not returned.

5. Save the field properties and close the **Index View** window.

   Note: When you save the modified index fields, the associated document type is updated accordingly.

6. From Recognition Designer, launch a **Template Test** from the toolbar. The **Template Test** window displays.

7. Select the **Launch Test** button. Detected lines display at the bottom of the window.

Here is an example of how the detection of lines works. On the following table there are three table fields, one on each column:

Table 4-3: Example of Line Detection

Field A	Field B	Field C
000	2000	
001		101
002	2002	102

- If fields A, B and C have the option **Detect all lines** cleared, then only the third row of the table is detected since only lines (cells) having values must be detected.
- If fields A, B and C have the option **Detect all lines** selected, then all the rows are detected since all the lines (cells) including those without a value must be detected.
- If fields A and B have the option **Detect all lines** cleared and field C has the option selected, then only the first and third rows are detected since a value must be detected in fields A and B. In the second row, there is no value in field B so this line (cells) is not detected.
- If fields A and B have the option **Detect all lines** selected and field C has the option cleared, then only the second and third rows are detected since a value must be detected in field C. In the first row, there is no value in field C so this line (cell) is not detected.

4.3.2 Table Recognition: A Complex Example

This example of a complex table in structured document shows dates that span several rows, but are only placed in the first row of the group. In this example, the dates are identified during recognition to extract paragraph header lines and the associated lines (cells) to create distinct columns.

The following example is a table with two paragraph header lines: “01/10/08 REF001” and “04/12/08 REF002”.

Table 4-4: Table with Paragraph Header Lines and Their Associated Lines

DATE	REF
01/10/08	REF001
	100.100
	110.120
04/12/08	REF002
	100.120
	120.130

DATE	REF
	140.200
	150.120

You may want to have the header lines in one column and the other lines in another column and also have the date applied to each line.

To obtain the results in this example:

1. Create three table fields named *<DATE>*, *<REF1>* and *<REF2>* by using Intelligent Capture Designer to create a document type in which you create the table fields and specify their validation properties.

   Note: For Dispatcher Manager, use the Index Family Editor to create an index family in which you create the table fields and specify their validation properties.

2. Specify the document type as an index family for your project template.

   See "Manually Positioning Index and Table Fields on a Template" on page 221.

3. In the **Index View**, place the table fields "REF1" and "REF2" on the "REF" column in the template. Superimpose the two table fields to cover the whole column height and width. Place the table field "DATE" over the date column. When running this in production, the table shown below is the result.

4. For each table field, select an engine configuration file in the **OCR engine** field on the **Recognition** tab.

5. For the three table fields, clear the **Detect all lines** field on the **Lines** tab.

6. Select the table field *<REF1>*.

7. Save the field properties and close the **Index View** window.

   Note: When you save the modified index fields, the associated document type is updated accordingly.

8. In the **Index Family Editor**, perform the following actions:

   a. Under **Lines Extraction**, expand **Paragraph Settings** and select **Paragraph** for the **Field Status**. This option enables differentiation of the paragraph header lines from the other lines.
   b. In the **Keywords** field, click the browse button and the **Keywords Editor** window displays.
   c. Click **Add** to create a keyword **Member**. Name the keyword *<REF>*.
   d. Select **Constant** for the **Type** value.
   e. Expand **Type Settings** and type *<REF>* in the **Value** field.
   f. Click **OK** to close the **Keywords Editor** and the keyword is set in the **Keyword** field.

g. Under **Paragraph Settings**, set **Headers Only** to **True**. It enables detection of only the headers of the paragraphs.

h. To have the paragraph header lines values applied to all the lines, select the following settings:

   i. Select the table field *<DATE>*.

   ii. Under **Paragraph Settings**, set **Field Status** to **Linked to Paragraph Field** to duplicate paragraph headers to empty lines, so the value of the first detected field is repeated on each line of the paragraph. For example, in the table below, the paragraph header lines values "01/10/08 REF001" and "04/12/08 REF002" are respectively duplicated to all the lines of each paragraph.

Table 4-5: Table with Dates Applied to All Empty Lines

DATE	REF1	REF2
01/10/08	REF001	REF001
01/10/08	REF001	100.100
01/10/08	REF001	110.120
04/12/08	REF002	REF002
04/12/08	REF002	100.120
04/12/08	REF002	120.130
04/12/08	REF002	140.200
04/12/08	REF002	150.120

4.4 Barcodes

Intelligent Capture supports recognition of Code 39 barcodes as well as of most frequently used 1D and 2D barcode types. The full list of supported barcode types is available in section "Supported Barcode Types" on page 434.

When recognizing barcodes, the index image must have a minimum resolution of 200 DPI. A lower resolution could render the barcode unreadable by the engine. A resolution of 300 DPI is recommended for smaller-sized barcodes.

4.4.1 Code 39 Barcodes

Code 39 is an alphanumeric barcode. A Code 39 barcode recognition file is included with Recognition Designer. This barcode type is used in industrial applications and has both original and extended versions.

- The original version enables to encode 43 characters including digits 0 to 9, letters A to Z, 6 symbols, plus one special character (*) that marks the beginning and end of the barcode. This character is not read during recognition.
- The extended version enables encoding of all *ASCII* table characters (128 characters). The 39 barcode has a variable, bidirectional length. Its name comes from its structure, 3 of 9 and is sometimes called Code 3 of 9 code or USD-3. Each character is encoded by 9 elements (5 bars, 4 spaces), of which 3 are large (1 binary) and 6 straight (0 binary). All characters are separated by a space, which are not counted as characters.

Note: When recognizing barcodes, the index image must have a minimum resolution of <*200*> *DPI*. A lower resolution could render the barcode unreadable by the engine. A resolution of <*300*> DPI is recommended for smaller-sized barcodes.

To create a barcode field for Code 39 barcode type:

1. Create a field for a barcode as follows:
   - For Recognition Designer, use Intelligent Capture Designer to create a document type in which you create the barcode field and then associate the document type with your project as an index family.
   - For Dispatcher Manager, use the Index Family Editor to create an index family in which you create the barcode field.
2. In the **Index View**, place the field on the index image barcode to be read.
3. Open **Project Options**. On the **Recognition** tab, select the engine recognition file `Basic_BarCode39.reco` for the **OCR engine**. `Basic_BarCode39.reco` applies to barcodes with at least nine characters. If the barcode to read has less than nine characters, create a custom engine recognition file as indicated next.
4. Do a recognition test: select the menu **Test** > **Unit Test**. If not satisfied with the settings of the engine configuration file, customize the file or create another one.

To create an engine recognition file for Code 39 barcodes of less than nine characters:

1. Create a new engine recognition file as outlined in "Adding Engine Configuration Files to the Project" on page 109.
2. Select the **Tools** > **OCR/ICR Engine** > **New** > **Barcode 39 Recognition**.
3. Give a name to the engine configuration file.

4. In the **Minimum number of characters** text box, enter the number of characters in output. The default setting is <9>.

4.4.2 1D Barcodes with General-Use OCR

General-Use OCR supports 1D barcodes. To know which barcodes are supported by General-Use OCR, see "Recognition Types Supported by Recognition Engines" on page 429 section of this guide. General-Use OCR can detect several barcodes of different types in a document. Recognition Designer is supplied with configuration files (*.reco) to detect 1D barcodes. Also, customize configuration files to detect specific 1D barcodes.

Note: When recognizing barcodes, the index image must have a minimum resolution of <*200*> *DPI*. A lower resolution could render the barcode unreadable by the engine. A resolution of <*300*> DPI is recommended for smaller-sized barcodes.

To define recognition for compatible 1D barcodes:

1. Create a field for a barcode as follows:
   - For Recognition Designer, use Intelligent Capture Designer to create a document type in which you create the barcode field and then associate the document type with your project as an index family.
   - For Dispatcher Manager, use the Index Family Editor to create an index family in which you create the barcode field.
2. In the **Index View**, create a zone on the image that contains the barcodes. For more information on processing types, see "Recognition Types Supported by Recognition Engines" on page 429.
3. Make sure the following conditions of use are met:
   - The thickness of barcode lines and the distance between them should be at least 3 pixels (that is a minimum line thickness of 0.25 mm (0.01 inch) in the case of 300 DPI resolution).
   - No parallel non-barcode lines of similar dimensions should be located less than 6 mm (0.25 inch) from the barcode.
   - The position of the barcode can be in any direction (except for the Postnet code where the maximum allowable skew of the barcode is 10).
   - Image size must not exceed 8400 x 8400 pixels.

4. In Recognition Designer, open the **Project Options** window (**File** > **Project Options**). In the **Recognition** tab, select the engine configuration file `General_Bar1D.reco` in the **OCR engine** box. This global engine supports five 1D barcodes: Code128, Code39, Codabar, *EAN*8, EAN13, and *ITF* (2 of 5 interleaved). There are several configuration files each of which supports 1D barcodes that are compatible between them. To know which 1D barcodes are supported, see the **Description** column in the "Select Resources" on page 274 window.

5. Perform a recognition test by selecting **Test** > **Unit Test**. If the zone contains several barcodes, `General_Bar1D.reco` will return the concatenated output values of all the barcodes. Write a script to process the output values.

To define recognition for specific 1D barcodes:

1. Select **Tools** > **OCR/ICR Engine** > **New** > **General-Use OCR**.
2. Type in a name to the engine configuration file.
3. Select **Barcode** in the **Recognition mode** list. The **Barcode parameters** pane displays
4. Select the specific type of 1D barcodes from the **Barcodes** list.

   Note: The following incompatibilities are known: Postnet is incompatible with all other barcodes, Code 128 is incompatible with *UCC* Code 128 and with Code 128 with check digit transmit, 2. Code 128 with check digit transmit is incompatible with UCC Code 128, EAN8 or EAN13 is incompatible with *UPC*-A, Code 39 is incompatible with the three other Code 39 types (Full *ASCII* mode, with check digit control and transmit, with start stop char transmit) and Codabar is incompatible with Codabar with start-stop char transmit.

5. Click **OK** and close the **General-Use OCR** window.

4.4.3 1D and 2D Barcodes with Barcode Recognition

Barcode Recognition supports 1D and 2D barcodes. To know which barcodes are supported by Barcode Recognition, see "Recognition Types Supported by Recognition Engines" on page 429 section of this guide. Barcode Recognition can detect several barcodes of different types in a document. Recognition Designer does not supply configuration files (*.reco). You can customize configuration files to detect specific 1D or 2D barcodes.

Note: When recognizing barcodes, the index image must have a minimum resolution of <*200*> *DPI*. A lower resolution could render the barcode unreadable by the engine. A resolution of <*300*> DPI is recommended for smaller-sized barcodes.

To define recognition for specific 1D barcodes:

1. Create a field for a barcode as follows:

- For Recognition Designer, use Intelligent Capture Designer to create a document type in which you create the barcode field and then associate the document type with your project as an index family.
- For Dispatcher Manager, use the Index Family Editor to create an index family in which you create the barcode field.

2. In the **Index View**, create a zone on the image that contains the barcodes. We recommend placing the zone on the whole page otherwise some barcodes (such as Add2 or Patch Code) may not be correctly recognized. For more information on processing types, see "Recognition Types Supported by Recognition Engines" on page 429.
3. Create a zone on the image that contains the barcodes.
4. Make sure the following conditions of use are met:
   - The thickness of barcode lines and the distance between them should be at least 3 pixels (that is a minimum line thickness of 0.25 mm (0.01 inch) in the case of <*300*>DPI resolution).
   - No parallel non-barcode lines of similar dimensions should be located less than 6 mm (0.25 inch) from the barcode.
   - The position of the barcode can be in any direction on the image.
5. Select **Tools** > **OCR/ICR Engine** > **New** > **Barcode Recognition**.
6. Type in a name to the engine configuration file.
7. In the **Barcode type** list, select **1D barcode**.
8. Select the specific type of 1D barcodes from the **Barcodes** list.

   Note: The following incompatibilities are known: Code 39 and Code 39 Extended, Code 128 and *UCC/EAN* 128 and Code 93 and Code 93 Extended. Also, Code 32 is a subset of Code 39. To detect Code 32 barcodes, use the Code 32 barcode otherwise the engine might confuse Code 32 and Code 39 barcodes.
9. Click **OK** and close the **Barcode Recognition** window.

To define recognition for specific 2D barcodes:

1. Select **Tools** > **OCR/ICR Engine** > **New** > **Barcode Recognition**.
2. Type in a name to the engine configuration file.
3. In the **Barcode Type** list, select **2D barcode**.
4. Select the specific type of 2D barcodes from the **Barcodes** list. Multi-selection is not allowed. At least one recognition barcode must be selected, otherwise an error message displays `No recognition barcode selected`.
5. Click **OK** and close the **Barcode Recognition** window.

Note: Because Barcode Recognition is not a full page engine, it is not recommended for use in the Template Wizard, the Table Wizard and Free Form Designer. It can be used with the rubber band tool.

4.5 Checks

This section provides recommendations for capturing data on checks. Recognition Designer enables data extraction on French and US checks with the Check Reading engine. This engine requires a license.

4.5.1 Capturing Data on Checks

To identify checks it is best practices recommendation to use *HPA* templates and place anchors on the *MICR* symbols printed on checks. The major MICR fonts are *E-13B* and *CMC-7*. France uses the *CMC-7* font. *US* checks use the *E-13B* font. Besides decimal digits, MICR fonts contains symbols.

To capture data on French and US checks:

1. Create two HPA templates, one for French checks and one for US checks.
2. On both templates, place three anchors on MICR symbols. Since MICR symbols are different in French and US checks, HPA technology can differentiate French and US checks.
3. Apply the following HPA settings:
   - **Pre-classification threshold** set between 60 and 65%.
   - **Search zone** width between 50 and 80 mm.
   - **Anchoring threshold** superior to 70%.
   - **Minimum hit number** set to the total number of anchors minus 1 (for example, set it to 3 if there are 4 anchors).
4. Run a test of each HPA template. If necessary, lower the **pre-classification threshold** to 55%.
5. Give each HPA template a different template code.
6. Create an index family and create the fields to extract. The following data can be extracted:
   - On French checks, the amount and the *CMC-7* code line.
   - On US checks, the amount, the E-13B code line, the check number, the date and, the payee line; the signature is detected.
7. Create one engine configuration file per field to extract. Learn more on the settings in the section "OCR Settings for Check Recognition" on page 154.
8. On both HPA templates, place the fields to extract. When placing fields, ensure they cover the entire document. This is required as Check Reading performs full

page reading. Since the field is placed over the whole page, it is not necessary to place any field anchor.

9. Run a template test. If a field is not correctly extracted, use the rubber band tool.

4.5.2 OCR Settings for Check Recognition

The Check Reading engine recognizes business and personal checks, deposits slips, cash-in, and cash-out documents whether they are hand printed, handwritten or machine printed documents.

Note: When recognizing barcodes, the index image must have a minimum resolution of *<200> DPI*. A lower resolution could render the barcode unreadable by the engine. A resolution of *<300>* DPI is recommended for smaller-sized barcodes.

To define *OCR* settings for check recognition:

1. Create one field in the index family. It is not necessary to create as many fields as there are **Data type** options to be detected in the check.
2. In the **Index View**, place the field on the entire check and then select the **Data type** options. The engine automatically retrieves the output value of all the selected **Data type** options.

   Note: Except for the **Signature detection** option, Check Reading detects the field area that is automatically zoomed-in in the Completion module and Identification (this is also available with the rubber banding option). The coordinates of the field area are returned through the `Bounds` object property.

3. Select the **Data type** options:

   - Select the menu **Tools** > **OCR/ICR Engine** > **New** > **Check Reading**.
   - Type in a name to the engine configuration file.
   - Select a language, either **French** or **English**. **English** has more options available.
   - Select the **Data type** options you want to read in the check.
   - Click **OK** and close the **Check Reading** window.

     Note: Because Check Reading is not a full page engine, it is not recommended for use in the Template Wizard, the Table Wizard and Free Form Designer. It can be used with the rubber band tool.

4.6 Designing Free Form Rules

Free form rules define which index fields and line item table data will be extracted from the image. Free form rules are built in Free Form Designer that is delivered with Recognition Designer as an auxiliary tool.

Once a free form rule is created, it can be associated with a generic template in Recognition Designer to constitute a free form template. A free form template must contain a unified set of rules that apply to most of the documents to be captured by your process. For example, to process invoices from different vendors, you can create a generic template associated with free form rules to process all invoices. You can do it instead of creating as many graphic templates as there are vendors. In this case, the free form rules must enable data extraction on all the invoices whoever the vendors are.

4.6.1 Recommendations for Designing Free Form Rules

This section offers recommendations to save time and be efficient when designing free form rules. Depending on the complexity of the project and document volumes, consider that up to one day may be necessary to develop, test and fine tune free form rules for one full text field. It is a best practice to focus the design and tuning efforts on the most frequent keywords and target data formats. Always test and fine tune the free form rules field by field first before testing the entire free form template.

4.6.1.1 Preparing Image Bases

Prepare two image bases: a knowledge base and a test base.

Knowledge Base

Collect approximately 500 images. Images must have different layouts and contain different data items, with different formats or patterns. For example, to process invoice dates, ensure to collect several layouts of invoice documents in which dates appear in different formats. A high variability of documents in the knowledge base is essential to create accurate free form rules. Use the knowledge base to design and tune the definition files.

Test Base

Collect approximately 100 images. As for the knowledge base, images must have different layouts and data in different formats. Use the test base to measure the error rates. Do not use the test base to tune the definition files; do not even examine the test results in details on the test base. Learn more recommendations on testing in the section "Testing Free Form Rules for Index Fields" on page 160.

4.6.1.2 Creating a Matrix of Full Text Fields

When designing full text fields, it is best practice to organize work by building a matrix spreadsheet to organize and track the information described here. This is essential for complex projects which can contain for example free form 100 templates with 3,000 full text fields.

Organize Full Text Fields

Review all the images of the knowledge base and list all the full text fields to extract. Organize fields depending on their frequency and on whether they are common to all templates or specific to some templates.

Name Full Text Fields

List all the full text fields and provide a name for the fields. Be sure that the name of the full text field in Free Form Designer is exactly the same as the name of the field in the index family.

List Target Data Formats and Keywords

Understand that you can define target data formats only or keywords only. If no keyword is defined, the algorithm searches the target; if no target data format is defined, the algorithm searches the keyword and the keyword is considered as being the target data format.

List Associated Words

For the keywords of each target, list them all and decide which associated words are necessary in addition to keywords. Associated words are useful to validate or invalidate potential targets.

Organize Field-specific Types

A field-specific type file contains a set of regular expressions to extract a specific target data format, a keyword or an associated word. Field-specific types are recommended to save time when defining regular expressions for data such as amounts, dates, numbers, *VAT* code, phone numbers, and zip codes. It is time saving to be able to define and update the set of regular expressions in one field-specific type file and apply this file to several definition files. To organize field-specific type files, list the data formats that are common to different full-text fields:

- If a full-text field has the same target data format and keyword in several templates, create one field specific type file for both the target data format and the keyword.
- If a full-text field has a target data format that appears in different templates but the keywords are different between templates, create one field specific type file for the target data format and one for the keyword.

Organize Definition Files

Here are recommendations to determine how many definition files are necessary and decide which full text fields to save to which definition files.

- For a project that requires full text relations, be sure to save all the fields involved in the relations to the same definition file so that relations can apply.
- For a project that addresses only one document class or less than five document classes and a small number of fields (less than five), create one definition file for each document class. It enables testing all the fields of a document class at the same time. For example, to address thousands of versions of insurance contracts and claims with five fields to extract on all documents, create two free form templates (one for contracts and one for claims) and two definition files (one for contracts and one for claims).
- For a project that addresses five document classes with ten fields, five of which are common to all document classes, create one definition file for each document class for all the fields that are unique to the document class. Create one definition file for each field that is common to all document classes.
- For a project that addresses tens of document classes with hundreds of full text fields, recommendations are:
  - If a field is common to several templates for both the target format and keywords (for example, a page number), create one definition file for the field.
  - If a field is common to several templates but only for the target format (for example, a date) and the keywords vary between templates, create one definition file for the field for each template (or optionally for each group of templates if you can determine groups of templates that have the same keywords).
  - For a multiple field, for example, an address block, create one definition file for the group of fields that compose the address block and then determine whether the target formats and keywords are common to the templates or not.

Maintain Definition Files

An efficient way to maintain definition files is to version them and add a description of major design updates. File versions and descriptions are useful to compare results of several definition files. Versioning definition files helps going back to a previous version of a file for example if test results evidence that a previous version was giving better results than the updated one. Learn to version files in the section "Versioning Free Form Definition Files" on page 187

Consider Full Text Relations

Understand that full text relations are usually not necessary. Full text fields are sufficient for most projects. Relations come in addition to full text fields to avoid

doubts or improve search results. Determine which full text fields would benefit from relations. Here are some examples:

- Invoice amount with a relation such as “Total Tax inc. = Total Tax exc. + Tax rate”.
- Date to be extracted when found below the patient ID number.
- Fields that are consecutive horizontally on the same row (amount A, amount B, amount C) or vertically in a column. This is frequent on tax forms and bank documents.

Be sure to save all the fields associated with a relation to the same definition file. You cannot set a relation for fields in different definition files and you cannot merge definition files. For these reasons, save time by determining as early as possible the fields to put in relation. To put in relations fields that are in different definition files, re-create the fields and settings in one single definition file.

4.6.1.3 Preparing Regular Expressions

Regular expressions are used for target data formats and keywords. To prepare regular expressions, follow these steps:

- Determine the most frequent target data formats. Treat them in priority.
- Prefer regular expressions to constants for keywords. *OCR* engine errors often prevent correct detection of constants. Working around these detection errors by lowering the **Hit threshold** percentage is not recommended. For example, if the OCR engine reads “inv01ce” instead of “invoice”, you may lower the **Hit threshold** to 70% to keep “inv01ce” but such a low threshold may capture constants other than “invoice”. To avoid typical OCR engine errors such as O and 0 or 1 and I, use regular expressions.
- Determine the target data formats which can be captured by a single regular expression; in other words, that do not require field-specific types.
- Write a regular expression that is representative of the format. For example, use “\d{2,3}\.\d{2}” to search for an amount with formats such as “10.00”, “100.60”, “99.20”. Do this either manually by examining the different existing formats or use the automatic builder of regular expressions. Learn to use this builder in the section “Edit Field-Specific Types” on page 327.
- For keywords and associated keywords, use regular expressions that are specific enough, though not too constrained. For example, use “in.{2,5}ce” to search for “invoice”. If regular expressions are not specific enough, the keyword and the target data may overlap. Learn to resolve this in the section “Resolving Keyword and Target Data Format Overlaps” on page 182.

Learn more and find examples of regular expressions in “Regular Expressions” on page 96.

4.6.1.4 Building Field Value Reference Files

Reference files contain the field values to extract with free form rules. In other words, the reference files contain the correct field values. During the test phase, you can compare the values extracted by free form rules with the correct values contained in the reference files. You then tune the free form rules until the extracted values as close as possible to the reference files. Learn to run this comparison in the section "Tuning Free Form Rules through Comparing to Reference Values" on page 167.

Format of Reference Files

Reference files are text files (TXT). A file contains one field value per line. For example, "<*N*>, IMAGE1.tif,1,22.06.01" where <*N*> is an incremental ID number, IMAGEn.tif is the image file name, 1 is a fixed value (to be left as is; required by the algorithm) and 22.06.01 is the field value. Make sure there are no extra characters or blank characters in the reference files. The quality of the reference files is important.

Recommendations for Creating Reference Files

Create one reference file for the target data formats and one reference file for the keywords. Create these two files from the knowledge base and from the test base; ultimately four files. When creating free form rules for a new recognition project, create the reference files manually. When creating free form rules to optimize a project already in production, collect the target data formats from the XML production data. XML data is not available for keywords.

4.6.2 Generating OCR Output Files for Testing

In Free Form Designer, the **OCR Reading** window is used to generate *OCR* output results. These results are saved and available to perform testing of free form rules in the **Search Keywords** window of Free Form Designer.

Caution

The OCR settings, including the images filters, defined here are only for the purpose of testing in Free Form Designer. These are not the settings that will apply to free form templates in production. For tests of free form rules in Free Form Designer to be representative of the results likely to be obtained in production conditions, it is essential that the settings selected here for generating the OCR output files for tests are the same as the OCR settings defined for the free form templates in production. Defining OCR settings for free form templates in production is explained in the topic "Creating Free Form Templates" on page 187.

To generate OCR output files for testing:

1. Run Recognition Designer and select **Tools** > **Free Form Designer**.
2. Select **OCR Reading** from left pane.

3. Select **OCR** > **Open Test Base** and load images to generate OCR data.
4. In the **OCR options** pane, select a recognition engine that supports full text recognition. For a list of engines that support full text recognition, see "Recognition Types Supported by Recognition Engines" on page 429. Adjust the confidence threshold as required.
5. Select the search zone, that is the zone of the image on which recognition is going to be performed.
6. Select the **Image filters** checkboxes to apply filters to the images before OCR runs.
7. Run recognition on the images.
8. When all images have been recognized, select **OCR** > **Save Results** to save the results as OCR files. There is one OCR file saved per each image. OCR files are saved to the directory where the original images reside. OCR files are readily available for future tests.

4.6.3 Testing Free Form Rules for Index Fields

Free Form Designer provides two types of tests:

- A unit test of a full text field or an element within a full text field. For example, a unit test can run on a single associated word or on all associated words.
- A test of the free form definition file with all the free form rules.

There is not any recommended order to carry out unit tests. For example, you may test the target data formats before testing anchor findings.

It is best practice recommendation to carry out unit tests and ensure the settings of individual elements are correct before running a test of the definition file. For complex full text fields, consider running a unit test of some of the elements within the full text field before running a unit test of the full text field.

4.6.3.1 Running a Unit Test on One Field Element

This topic explains how to run a unit test on one element within an index field. The aim is to ensure the settings of the tested element are correct before running a unit test of the index field.

To run a unit test on one element in a field:

1. Select the element to be tested in the **Settings** pane, right-click and select **Unit test**
2. In the **Search Keywords** window, select **Search** > **Start search** to start the test on the current images. Images are loaded automatically if you have already performed recognition in the **OCR Reading** window. In this case the recognition results display in the **Content** tab. Alternatively load recognition results from *OCR* files saved during previous recognition.

3. In the **Images** pane, check the images on which hypotheses are found (they have a green tick in the fourth column to the right) and the images on which no hypotheses are found (they have a red cross in the fourth column to the right).
4. On an image on which hypotheses have been found select the **Detail of a Field** tab to check all the hypotheses found for the selected element. Select a hypothesis to view in a red box on the image.
5. Select the **List of Hypotheses** tab to check the score obtained by all the hypotheses. Ensure the best hypothesis, that is the one whose score is 0, is the one you consider to be the best hypothesis.

4.6.3.2 Running a Unit Test on an Index Field

This topic explains how to run a unit test on an index field. The aim is to ensure the settings of all the elements in the full text field are correct before running a test of the definition file.

To run a unit test on an index field:

1. Select the full text field to be tested in the **Settings** pane, right-click and select **Unit test**
2. In the **Search Keywords** window, select **Search** > **Start search** to start the test on the current images. Images are loaded automatically if you have already performed recognition in the **OCR Reading** window. In this case the recognition results display in the **Content** tab. Alternatively load recognition results from *OCR* files saved during previous recognition.
3. In the **Images** pane, check the images on which hypotheses are found (they have a green tick in the fourth column to the right) and the images on which no hypotheses are found (they have a red cross in the fourth column to the right).
4. On an image on which hypotheses have been found select the **Detail of a Field** tab to check all the hypotheses found for the selected full text field. Select the full text field (it has an icon in the form of a tree structure) to view on the image the keyword (in a red box) and the target data format (in a green box) that are linked together by a blue line to indicate that this is the best hypothesis so the one that is retained.
5. Select the **List of Hypotheses** tab to check the score obtained by all the hypotheses. Ensure the best hypothesis, that is the one whose score is 0, is the one you consider to be the best hypothesis.

4.6.3.3 Running a Test of the Definition File

This topic explains how to test a definition file with all the full text rules. The aim is to ensure the settings of the full text rules are correct before associating free form rules with free form templates in Recognition Designer.

To run a test of the definition file:

1. Select **Search Keywords** from the right pane.
2. In the **Search Keywords** window, select **Search** > **Start search** to start the test on the current images. Images are loaded automatically if you have already performed recognition in the **OCR Reading** window. In this case the recognition results display in the **Content** tab. Alternatively load recognition results from *OCR* files saved during previous recognition.
3. In the **Images** pane, check the images on which hypotheses are found (they have a green tick in the fourth column to the right) and the images on which no hypotheses are found (they have a red cross in the fourth column to the right).
4. On an image on which hypotheses have been found select the **Detail of a Field** tab to check all the hypotheses found for all the elements within all full text fields. Select a full text field (it has an icon in the form of a tree structure) to view on the image the keyword (in a red box) and the target data format (in a green box) that are linked together by a blue line to indicate that this is the best hypothesis so the one that is retained for the full text field.

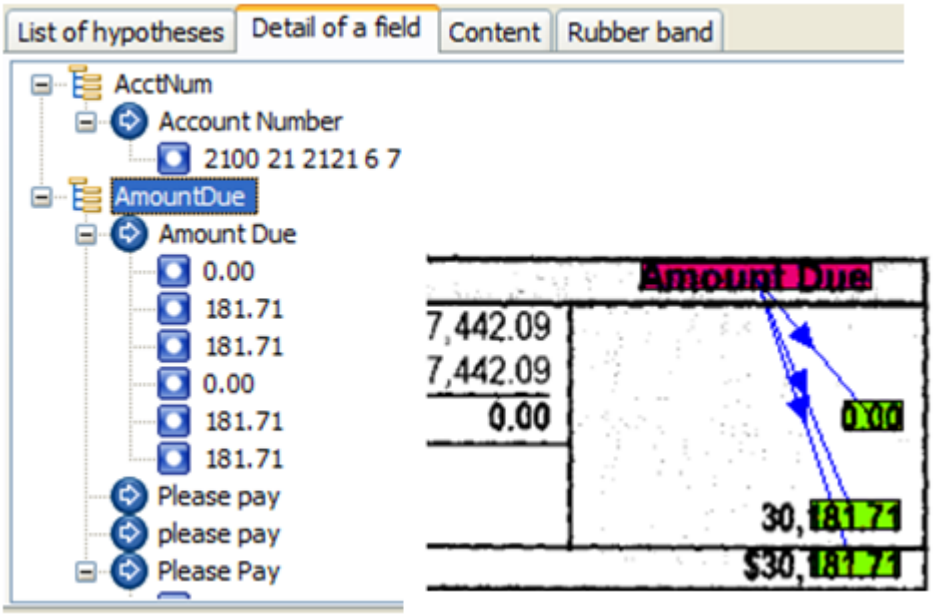


Figure 4-4: Detail of field

5. Select the **List of Hypotheses** tab to check the score obtained by all the hypotheses. Ensure the best hypothesis, that is the one whose score is 0, is the one you consider to be the best hypothesis. Learn on hypotheses in the section "Understanding Hypotheses Returned by the Algorithm" on page 196.

4.6.4 Creating and Testing Free Form Rules for Line Item Extraction

To avoid common pitfalls, follow these steps to develop the free form rules for line-item extraction. Define target data formats and associated relations first. Then, if necessary, create anchor findings to improve results. Learn the fundamentals and recommendations on line-item extraction in the section "Using Free Form Rules for Line Items Extraction" on page 194.

To define free form rules for line-item extraction:

1. Select **Tools** > **Free Form Designer**. From the **Settings** pane of Free Form Designer, create a **Full text table** node by selecting the **+** button in the toolbar.
2. Select the **Full text table** node.
3. In the **Full text table** node, select **+** in the toolbar to create as many columns as there are table fields in the index family.
4. Give the columns the same name as the table fields in the index family. For example, if the index family has three table fields "Field1", "Field2" and "Field3" then the columns should be named "Field1", "Field2" and "Field3".

   For help naming columns exactly as table fields, select **File** > **Link to an Index Family** and select the index family. Right-click a column and select the **Rename** menu to display the list of table fields in the index family and pick up the table field to rename the current column accordingly.
5. For each column, define the target data formats as explained in "Defining Target Data Formats" on page 170.
6. Perform a unit test of the target data formats as explained in "Running a Unit Test on One Field Element" on page 160.
7. Define the primary rows, in other words, define the combination of table columns to be found for a row to be processed as a primary row:

   a. Select the **Full text table** node to display the settings in the right pane.

   b. Select the columns from the **Available columns** list and add them to the **Primary rows** list. For example if there are three columns "Field1", "Field2" and "Field3", add those columns to the **Primary rows** list so that a row is processed as a primary row on condition that it contains the three columns "Field1", "Field2" and "Field3". Any row that does not contain these three fields is processed as a secondary line.
8. Test that primary rows are correctly detected:

a. Generate the OCR data for the test as explained in the topic "Generating OCR Output Files for Testing" on page 159.

b. Select the **Full text table** node, right-click and select **Unit test**.

c. In the **Search Keywords** window, check the test results in the **Primary Row** tab. This tab shows the rows found for each primary row definition. The **Primary row definition** pane indicates the different primary row definitions with for each definition, the column names involved in the primary row definition. Those column names also appear in the first column of the **Primary Rows** tab. The second and next columns are for the text boxes found in the line (one column per text box found). The **Lines** pane lists all the rows found for the primary row definition. The rows with a tick symbol are the rows that match the primary row definition.

9. Define order and/or script relations between columns as explained in "Defining Relations Between Columns" on page 166. Understand that some projects may not require order or script relations if the header relations are sufficient. Learn on header relations in the section "Understanding the Free Form Algorithm for the Full Text Table Field" on page 197.

10. Test the relations between columns:

a. Select the **Full text table** node, right-click and select **Unit test**.

b. In the **Search Keywords** window, select the menu **Search** > **Start search** to start the test.

c. Check the test results on the image: the items found for columns in primary row appear in green frames on the image. The items found for columns in rows that are not primary rows appear in yellow frames.

d. Check the test results in the tabs **Per Row And Per Relation**, **Per Row**, **Whole Table** and **Final Results**. These tabs are described in "Full Text Table Tabs" on page 351.

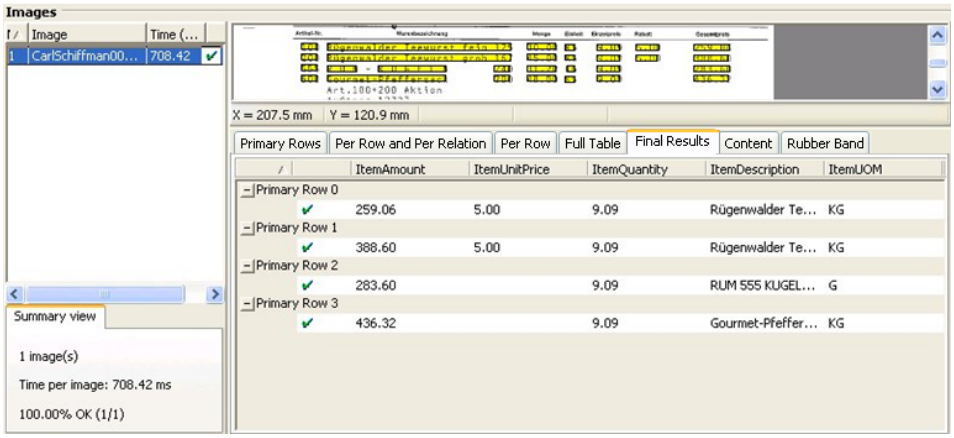


Figure 4-5: Final Results tab

11. If the detection of primary row is not accurate enough, define **Anchor findings** as explained in the topic "Creating Anchor Findings" on page 177.

12. To merge the secondary rows with the primary rows (if any secondary rows in the column), scroll the list of columns in the **Grouping column** pane and select the column that contains secondary rows to be merged with the primary row. It is possible to select only one column per table.

    Secondary rows are merged with a primary row in the following order:

    - All the secondary rows appearing above the primary row.
    - Data in the columns to the left and to the right of the grouping column that are not defined in the full text table. For example, there are three columns ("column 1", "column 2" and "column 3") to the right of the grouping column. "Column 3" is defined in the full text table. "Column 1" and "Column 2" are not defined in the full text table. The data from "Column 1" and "Column 2" is merged with that of the grouping column but the data of "Column 3" is not merged.
    - All the secondary rows appearing below the primary row.

    Merged data is separated by spaces. In Figure 4-6, merging of the grouping column and secondary rows returns: "005 GENERALPATIO GENERAL (36 X 77 7/8) 121S, WHITE, CHRF, SINGLE, 221S, 7/8 BUG, 824 LINE 5".

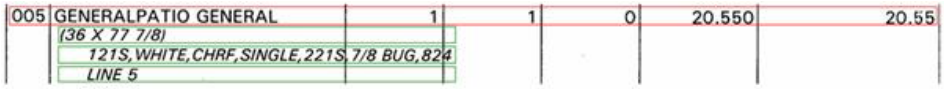


Figure 4-6: Merged data separated by spaces

13. Select **File** > **Save** to save the free form rules. Free form rules are saved to a definition file (*DFT*). All definition files must be saved to `<Project directory>\Resources\OCR\`. This is the default path when you open Free Form Designer from Recognition Designer. Files saved to another path will not be available in Recognition Designer.

14. Associate the free form rules with the table field as explained in the topic "Creating Free Form Templates" on page 187.

4.6.4.1 Defining Relations Between Columns

Learn the purpose of relations for line-item extraction and find some recommendations in the section "Using Free Form Rules for Line Items Extraction" on page 194.

To define relations between columns for line-item extraction:

1. To define an order relation:
   a. Select **+** in the **Order relations** pane.
   b. In the **Order Relation Definition** window, define an order relation. For example, the relation ."? Col1 Col2 Col3" means that the primary row can contain any column or no column, followed by three columns, followed by any column. For help using the **Order Relation Definition** window, see "Order Relation Definition" on page 336.
2. To define a script relation:
   a. Select **+** in the **Script relations** pane.
   b. In the **Script relation definition** pane, add the columns to the relation and click the **Edit Relation Script** button to write the script. Use `DpLineItems`, a collection of string. At execution, it contains a combination of values of the script relation columns in the order defined in the script relation. The function result must be set to True if the relation matches the combination of column values. Otherwise the function result should be set to False. Each collection item value can be accessed either by the column index or by the column name. For more information on `DpLineItems`, see the *Scripting Guide*.
   c. In the script editor pane, select **File**>**Save**. The script file takes automatically the name of the associated free form definition file (*DFT*) and is saved by default to the following path : `[Dispatcher project]/Resources/OCR`. This is the path where the free form definition file (DFT) is saved. If you rename the free form definition file, also rename the associated script file. Otherwise, the latter is not taken into account by the definition file. There is one script file per definition file. You can include other scripts in the relation script (for example with the `#uses` command).
3. After testing the detection of table columns based on the defined **Target data formats** and their associated relations (order relations and script relations), it may be necessary to further improve the detection of table columns. To do so, define **Anchor findings**. For help creating **Anchor findings**, see "Creating Anchor Findings" on page 177.
4. If a column contains secondary lines, have these secondary lines read and merged with the primary row. To do so, scroll the list of columns in the **Grouping column** pane and select the column whose secondary lines you want to merge with the primary row. You can select only one column per table.

   Merging of a primary row with its secondary lines is performed in the following order:

- All the secondary lines above the primary row
- Data in the columns to the left and to the right of the grouping column that are not defined in the full text table. For example, there are three columns (column 1, column 2 and column 3) to the right of the grouping column. Column 3 is defined in the full text table. Columns 1 and 2 are not defined in the full text table. The data from columns 1 and 2 is merged with that of the grouping column but the data of column 3 is not merged.
- All the secondary rows below the primary row

Merged data is separated by spaces. In Figure 4-7, merging of the grouping column and secondary lines returns: “005 GENERALPATIO GENERAL (36 X 77 7/8) 121S, WHITE, CHRF, SINGLE, 221S, 7/8 BUG, 824 LINE 5”

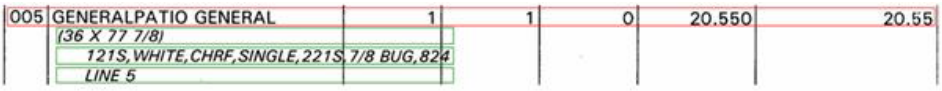


Figure 4-7: Merged data separated by spaces

4.6.5 Tuning Free Form Rules through Comparing to Reference Values

This section explains how to tune free form by comparing test results to reference values. Reference values can be created manually in the case of a new project or automatically by exporting the text results as explained next in this section. To facilitate fine tuning, ideally create several versions of the definition files. This enables comparing the results obtained with the latest version of the definition file and decide to continue fine tuning the definition file or go back to the previous definition file if it is giving better results. When creating several versions of the definition file, it is best practice recommendation to enter comments to document the changes to free form rules. Learn more in “Versioning Free Form Definition Files” on page 187. This section applies to both full text index fields and the full text table field.

To tune free form rules by comparing between definition files or to reference data:

1. Generate reference values either manually or automatically. Creating reference values manually is explained in the section “Building Field Value Reference Files” on page 159. To create reference values automatically, carry out a test of the current definition file using the **Search Keywords** window and select **Export results to a text file**.

2. Compare the current test results to reference values: From the **Tools** menu of the **Search Keywords** window, select **Compare to a reference**. Statistical values are provided in the **Summary** pane. They are different for full text fields and for the full text table: See next.

Statistical Values for Full Text Fields

Statistical values for the full text fields are:

- **Correct values**: Number of images out of the total image number on which field values are identical in both search results and reference data. If the number of identical items is lower than expected, make sure that the images are sorted in the same order (sort by the <*N*> column) in both current project and reference data. When exporting the results (to the TXT file), data is exported in the order in which it is displayed.
- **Including non empty**: Subset of preceding quantity, including the number of images on which field values are not empty and which are identical in both search results and reference data.
- **Wrong and non empty values**: Number of images on which field values are found (not empty), but are different from the reference data.
- **Empty references**: Number of images on which field values are empty (blank fields) in the reference data.
- **Unfound values**: Number of images on which field values are empty in the search result (blank fields) and should be not empty.
- **Wrongly found values**: Number of images for which the system has found some field values whereas they are not present in the reference data.
- **Correct rate**: Percentage of correct fields (including blank fields).
- **Wrong rate**: Percentage of incorrect and non blank fields.
- **Unfound rate**: Percentage of missed fields.
- **Wrongly found rate**: Percentage of wrongly found fields.

Statistical Values for Full Text Table

Statistical values for the full text table are:

- **Tested image**: Number of tested images
- **Reference images**: Number of images in the reference data file
- **Unfound images**: Number of tested images not found in the reference data file
- **Referenced values**: Number of items of the column in the reference data
- **Found values**: Number of items of the column returned by the *LIFFE*
- **Correct values**: Number of correct values in the column returned by the LIFFE
- **Wrong values**: Number of wrong values in the column returned by the LIFFE
- **Unfound values**: Number of unfound values in the column returned by the LIFFE
- **Wrongly found values**: Number of wrongly found values in the column returned by the LIFFE

- **Correct rate**: Percentage of correct values
- **Wrong rate**: Percentage of wrong values
- **Unfound rate**: Percentage of unfound values
- **Wrongly found rate**: Percentage of wrongly found values
- **Correct mean rate**: Mean rate of correct values on all the images
- **Wrong mean rate**: Mean rate of wrong values on all the images
- **Unfound mean rate**: Mean rate of unfound values on all the images
- **Wrongly found mean rate**: Mean rate of wrongly found values on all the images
- **Tested image**: Number of tested images
- **Reference images**: Number of images in the reference data file
- **Unfound images**: Number of tested images not found in the reference data file
- **Referenced values**: Number of items of the full text table in the reference data
- **Found values**: Total number of items on all the columns returned by the LIFFE
- **Correct values**: Total number of correct values on all the columns returned by the LIFFE
- **Wrong values**: Total number of wrong values on all the columns defined in the definition file
- **Unfound values**: Total number of unfound values on all the columns defined in the definition file
- **Wrongly found values**: Total number of wrongly found values on all the columns defined in the definition file
- **Correct rate**: Percentage of correct values
- **Wrong rate**: Percentage of wrong values
- **Unfound rate**: Percentage of unfound values
- **Wrongly found rate**: Percentage of wrongly found values
- **Correct mean rate**: Mean rate of correct values on all the images
- **Wrong mean rate**: Mean rate of wrong values on all the images
- **Unfound mean rate**: Mean rate of unfound values on all the images
- **Wrongly found mean rate**: Mean rate of wrongly found values on all the images

4.7 Creating Free Form Rules

This section contains the procedures to create free form rules by using all the settings available in Free Form Designer. Definitions of all the elements described in this section such as **Target data format**, **Anchors findings** or **Full-text relations** are defined in the section "Understanding Free Form Rules" on page 190.

4.7.1 Defining Target Data Formats

A target data format identifies elements in a document by type. The target format can be a date, such as a delivery date or an invoice date, a number, such as an invoice, file, or customer number, or a piece of text, such as a company or customer name. A full text field may have several Target data formats with unique parameters defined by type.

To define a target data format:

1. Run Recognition Designer and select **Tools** > **Free Form Designer**.
2. Select the **Full-text fields** node and select **+** from the toolbar to create a full text field. Create as many full text fields as there are index fields to process with free form settings.
3. Give each full text field the same name as that of the index field:
   a. Select **File** > **Link to an index family** and select the index family that contains the index fields.
   b. Right-click the full text field, select **Rename**. The list of fields in the associated index family displays.
   c. Select a field name from the list of fields. The full text field takes the name of the selected field.
4. Create the **Target data format**. This is the information to find in the document, such as a delivery date, invoice date, invoice number, file number or customer number. Target data is defined by one of the following types:
   - **Regular expression**: Search different patterns of dates or amounts. A regular expression is a string used to describe or match a set of characters, according to certain syntax rules. During recognition or validation steps, regular expressions can, for example, search for specific characters, the position of characters in a string, or specific grouping of characters. For more information on using and creating regular expressions, see "Regular Expressions" on page 96.
   - **Fuzzy regular expression**: Unlike the **Regular expression** option, fuzzy regular expressions take advantage of the **Hit threshold** option, which enables a single regular expression to have a wider range of text matches. For more information, see "Fuzzy Regular Expressions" on page 100.
   - **Constant**: Search a term, such as "invoice", a static amount or a specific date.

- **Field-specific type**: Combines several regular expressions or constants for greater flexibility. For example, you might build a list of regular expressions or constants automatically from a text file containing a list of formats or keywords. Define the output format so that the extracted value always has the same pattern, such as `DD\MM\YYYY` for a date. The pattern must match the pattern that occurs in the document.

  To define field-specific types, select the text file containing the list of formats. The formats are created and the new regular expressions display in the **List of expressions** pane in the **Edit Field-Specific Types** window. If editing a definition file that already contains regular expressions, the new regular expression list has a priority greater (+1) than previous expressions.

Note: Constants should only be used when the target data format is very precise. Otherwise, false positives are likely. When using numerous regular expressions, consider creating a field-specific type to define a set of regular expressions. It is easier to add a new regular expression to a field-specific type than it is to create a target data format.

5. Select **File** > **Save** to save the free form settings to a definition file (*DFT*). Definition files must be saved to *<Project directory>*`\Resources\OCR\`. Files saved elsewhere will not be available in Recognition Designer. The application will use this default path when Free Form Designer is opened from Recognition Designer. Learn recommendations to organize definition files in the section "Creating a Matrix of Full Text Fields" on page 156.

4.7.2 Understanding Field-Specific Types

Use the **Edit Field-Specific Types** window in Free Form Designer to control how information that appears in multiple formats throughout a document is recognized. By defining field-specific types, information formatted differently across documents can be converted to a consistent format during recognition. An excellent and simple example of this functionality is the handling of dates. A date can appear in multiple formats in different documents.

For example, a single date might take any of the following formats:

- 07/21/10
- 21/07/10
- 07/21/2010
- 21/07/2010
- July 21, 2010
- 21 July 2010
- 07.21.2010
- 21.10.2010
- 21,10,2010

If properly configured in the **Edit Field-Specific Types** window, each of these variations can be recognized and converted to a single date format during recognition. A complete example of implementing this functionality using regular expressions is provided in the topic "TFT Scripting Samples" on page 174. This topic also contains additional examples of modifying currency formats for consistent recognition.

VBA scripting is used to create scripts that control processing of field-specific types. Easy Basic scripts created with previous versions of Recognition Designer are automatically migrated to VBA when they are opened in the **Scripting** tab of the **Edit Field-Specific Types** window. Only *VB* scripting is supported, and all new scripts must be written using VBA. This editor is provided using WinWrap functionality. For specific information about using the editor, see the *WinWrap Basic Language* section of the *WinWrap Editor Help* files available from the **Help** menu.

When the **Scripting** tab in the **Field-Specific Types** window is selected, a script relation is created in a Free Form Designer project . **Edit Field Specific Types** scripting provides a function which contains field-specific type format data. This `FieldSpecificTypeFormat` function is displayed and takes focus:

```
Private Sub FieldSpecificTypeFormat(var Value as String)  End Function
```

The function has values that enable control of data formats:

- *`Value`* (Type = String): This is the field value returned by *OCR* engine. Modify *`Value`* to change the data format.
- `var` means that *`Value`* is a read/write parameter. It is initialized when function starts and can be customized by modifying the code.
- This function does not return a value.

The `FieldSpecificTypeFormat` function cannot be renamed or the script does not work as expected. Instead, the `FieldSpecificTypeFormat` converts to a common function and is not used by the field-specific type script. The next time the **Field-Specific Type** window **Scripting** tab is selected, Free Form Designer will create a `FieldSpecificTypeFormat` function because no `FieldSpecificTypeFormat` function exists.

Although field-specific type scripting does not have access to the Recognition Designer object model, other `BAS` files can be referenced by adding the `#Uses` command at the beginning of the script. When referencing other scripts using this type of include statement, save included scripts in the same path as the *TFT* file.

For Unicode support, select the encoding format of the `TXT` file to load when importing data in Recognition Designer or Free Form Designer:

- Autodetect
- ANSI
- UTF-7
- UTF-8

- Unicode (UTF-16 Little-Endian)
- Big-Endian (UTF-16 Big-Endian)

4.7.2.1 Defining a Field-Specific Type

The **Definition of Field-Specific Types** window displays a full featured VBA editor and expressions builder. From this window, field-specific types are defined and saved.

Write and edit scripts from the **Scripting** tab. Regular expressions are defined on the **Expressions** tab. For help using the script editor, see the *WinWrap Basic Language* section of the *WinWrap Editor Help* file available from **Help** menu. For help working with regular expressions, see "Regular Expressions" on page 96.

To create a field specific type script:

1. On the **Settings** tab of Free Form Designer , create a full text field.
2. Select the **Target data format** node under the full text field, and click **+** to add a new target data format.
3. Select **Field specific type** from the **Type** list box on the **Parameters** pane.
4. Select **Edit** > **Field-Specific Types** to display the **Edit Field Specific Types** scripting window.
5. Create the script using the menu and toolbar options:
   a. Enter regular expressions on the **Expressions** tab.
   b. Type VB scripting functions on the **Scripting** tab.
   c. Select an object from the **Object** list box to add an event for the object.
   d. Type the script text, or select an available event from the **Proc** listbox to insert the event script automatically. Available events differ based on the object selected.
      - When an object is selected, all the events available to that object are listed in the **Proc** listbox. Events present in the script are displayed in bold text.
      - Selecting a function from the **Proc** listbox inserts the event at the end of the current script. If the event is already present in the script, selecting it from the listbox places the cursor inside the correct subroutine.
      - When the **(General)** object is selected from the **Object** listbox, all implemented events plus all user subroutines display in the **Proc** listbox.
      - Although the *DOM* is not accessible from the **Definition of Field-Specific Types** window, other `BAS` files can be referenced by adding the `#Uses` command at the beginning of the script. When referencing other scripts using this type of include statement, save the included scripts in the same path as the TFT file.

6. Save the script. If the scripting editor is closed without saving, a prompt displays. If the script has been modified but not saved, and the **Edit Field-Specific Type** window closed without saving, prompts display to save both the BAS and TFT files.

7. Test the script recognition results using **Unit Test** and **Template Test** tools in Recognition Designer.

Related Topics

"Performing a Template Test" on page 200

4.7.2.2 TFT Scripting Samples

The following examples provide help understanding field-specific type scripting. These examples demonstrate two common functions: 1) Standardizing date formats; and 2) Standardizing currency formats.

Standardize Amount Format from US to AUD Currency

```
Private Sub FieldSpecificTypeFormat(ByRef Value as String)
    Dim tmpStr As String

    tmpstr=Value

    tmpstr= DFLReplaceString(tmpstr,"dollar","AUD")
    tmpstr= DFLReplaceString(tmpstr,"$","AUD")

    Value=tmpStr
End Sub
```

Standardize Amount Format from AUD to UK Currency

```
Private Sub FieldSpecificTypeFormat(ByRef Value as String)
    Dim tmpStr As String

    tmpstr=Value

    tmpstr= DFLReplaceString(tmpstr,"","GBP")
    tmpstr= DFLReplaceString(tmpstr,"pence","GBP")
    tmpstr= DFLReplaceString(tmpstr,"pounds sterling","GBP")
    tmpstr= DFLReplaceString(tmpstr,"pound sterling","GBP")
    tmpstr= DFLReplaceString(tmpstr,"pound","GBP")
    tmpstr= DFLReplaceString(tmpstr,"sterling","GBP")
    Value=tmpStr
End Sub
```

Standardize Date Formats using Regular Expressions

This example is a more complex implementation of field specific type scripting making extensive use of regular expressions.

```
Private Function CompYear(inputyy as integer) as Integer
'-------------------------------------------------------------//
  if inputyy<1900 then
    if inputyy>80 then
        CompYear=1900+inputyy
    else
      CompYear=2000+inputyy
    end if
  else
```

```
 10     CompYear=inputyy
 11   end if
 12 end function
 13
 14 Private Function formatDateAlpha(InputlocalValue as string) as string
 15 '-----------------------------------------------------------//
 16 Dim nbMax As Long, i As Long, tabpos(2) As Long, tablen(2) As Long
 17 Dim errMsg As String, tmpstr(3) As String, mstr As String
 18 Dim d As Long, yy As Long, m As Long
 19
 20   formatDateAlpha=""
 21
 22   nbMax=1 'looking for the month
 23   DFLRegExp("(?i)(jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec)[a-z]
{0,10}",InputlocalValue,nbmax,tabPos,tablen,errMsg)
 24   if nbMax=1 then
 25     mstr=DFLStrCopy(InputlocalValue,tabpos(0),3)
 26     if DFLStrLower(mstr)="jan" then
 27       m=1
 28     elseif DFLStrLower(mstr)="feb" Then
 29       m=2
 30     elseif DFLStrLower(mstr)="mar" Then
 31       m=3
 32     elseif DFLStrLower(mstr)="apr" Then
 33       m=4
 34     elseif DFLStrLower(mstr)="may" Then
 35       m=5
 36     elseif DFLStrLower(mstr)="jun" Then
 37       m=6
 38     elseif DFLStrLower(mstr)="jul" Then
 39       m=7
 40     elseif DFLStrLower(mstr)="aug" Then
 41       m=8
 42     elseif DFLStrLower(mstr)="sep" Then
 43       m=9
 44     elseif DFLStrLower(mstr)="oct" Then
 45       m=10
 46     elseif DFLStrLower(mstr)="nov" Then
 47       m=11
 48     elseif DFLStrLower(mstr)="dec" Then
 49       m=12
 50     end if
 51   else
 52     Goto gotoFin
 53   end if
 54
 55   nbMax=2 'looking for numeric items
 56   DFLRegExp("\d+",InputlocalValue,nbmax,tabPos,tabLen,errMsg)
 57   if nbMax=2 then
 58     d=DFLStrToInt(DFLStrCopy(InputlocalValue,tabpos(0),tablen(0)))
 59     yy=DFLStrToInt(DFLStrCopy(InputlocalValue,tabpos(1),tablen(1)))
 60     yy=CompYear(yy)
 61   else
 62     Goto gotoFin
 63   end if
 64
 65
 66
 67   'format output result
 68   formatDateAlpha = DFLFormatVal("00",d)+"/"+DFLFormatVal("00",m)
+"/"+DFLFormatVal("0000",yy)
 69
 70   gotoFin:
 71 end function
 72
 73 Private Function formatDateNum(InputlocalValue as string) as string
 74 ' supported formats
 75 ' \d{1,2} ?/ ?\d{1,2} ?/[""]? ?(\d{4}|\d{2}) =>slash
 76 ' \d{1,2} ?\. ?\d{1,2} ?\. ?(\d{4}|\d{2})=> dot
 77 ' \d{1,2} ?- ?\d{1,2} ?- ?(\d{4}|\d{2})=> dash
 78 ' \d{1,2} \d{1,2} (\d{4}|\d{2})=> space
```

```
 79 'extended format
 80 '[\dO]{1,2} ?[1$/] ?[\dO]{1,2} ?[1$/][""]?([\dO]{4}|[\dO]{2})
 81 '-------------------------------------------------------------//
 82 Dim nbMax As Long, i As Long, tabpos(3) As Long, tablen(3) As Long
 83 Dim d As Long, yy As Long, m As Long
 84 Dim errMsg As String
 85 Dim simpleRegExp As String, extRegExp As String
 86
 87   FormatDateNum=""
 88
 89   InputlocalValue=DFLReplaceChar(InputlocalValue,Chr(39),"")'replace '
 90   InputlocalValue=DFLReplaceChar(InputlocalValue,Chr(34),"")'replace "
 91   InputlocalValue=DFLReplaceChar(InputlocalValue," ",")'replace space
 92
 93   simpleRegExp="\d{1,2} ?[ /\.-] ?\d{1,2} ?[ /\.-] ?(\d{4}|\d{2})" 'combination of
the 4th simple formats"/>
 94   nbMax=1
 95   DFLRegExp(simpleRegExp,InputlocalValue,nbmax,tabPos,tabLen,errMsg)
 96
 97   if nbMax=1 then
 98     nbMax=3
 99     DFLRegExp("\d+",InputlocalValue,nbmax,tabPos,tabLen,errMsg)
100     if nbMax=3 then
101       d=DFLStrToInt(DFLStrCopy(InputlocalValue,tabpos(0),tablen(0)))
102       m=DFLStrToInt(DFLStrCopy(InputlocalValue,tabpos(1),tablen(1)))
103       yy=DFLStrToInt(DFLStrCopy(InputlocalValue,tabpos(2),tablen(2)))
104       yy=CompYear(yy)
105     end if
106     formatDateNum = DFLFormatVal("OO",d)+"/"+DFLFormatVal("OO",m)
+"/"+DFLFormatVal("OOOO",yy)
107     Goto gotoFin
108   end if
109
110   extRegExp="(?i)[\dO]{1,2} ?[1$/] ?[\dO]{1,2} ?[1$/]([\dO]{4}|[\dO]{2})"
111   nbMax=1
112   DFLRegExp(extRegExp,InputlocalValue,nbmax,tabPos,tabLen,errMsg)
113   if nbMax=1 then
114     InputlocalValue= DFLReplaceChar(InputlocalValue,"O","0")
115     InputlocalValue= DFLReplaceChar(InputlocalValue,"o","0")
116     formatDateNum=inputlocalValue+"???"
117     Goto gotoFin
118   end if
119
120   formatDateNum=InputlocalValue+"???"
121
122   gotoFin:
123
124 end function
125
126 Private Sub FieldSpecificTypeFormat(ByRef Value as String)
127 '-------------------------------------//
128 ' format  dates                       //
129 '-------------------------------------//
130
131 '-------------------------------------------------------------//
132 '-------------------------------------------------------------//
133
134 '-------------------------------------------------------------//
135
136
137 '-------------------------------------------------//
138 'Point d'entre principal
139 '-------------------------------------------------//
140 Dim tmpStr As String, tmpStr1 As String 'pour test
141
142   tmpStr=Value
143
144   tmpStr1=formatDateAlpha(tmpStr)
145   if tmpStr1="" then
146     tmpStr1=formatDateNum(tmpStr)
147   end if
```

```
148
149   Value=tmpStr1
150
151
152 '------------------------------------------------//
153 ' ---Fin---
154 '------------------------------------------------//
155 End Sub
156
157
```

4.7.3 Creating Anchor Findings

Anchor findings include keywords and associated words that provide information about words associated with the target data. After *OCR*, these words help the application to accurately locate and extract target data. For example, to locate the purchase order number on an invoice where the word “invoice” always appears, use “invoice” as a keyword and specify the position where the word appears relative to the target data (the purchase order number).

To further refine the location of the purchase order number, add associated words that validate or invalidate a potential keyword match. The associated words algorithm keeps only keywords for which all validating associated words are found and no invalidating words are found. For example, on some invoices a purchase order number might be represented by an abbreviation, such as “P.O.”, or “P.O. Number”. If only the keyword “P.O.” is used to identify the target data, then other occurrences of “P.O.”, such as “P.O. Date” or “P.O. Box”, would be included in the results. Using associated words, such as “Date” or “Box” to invalidate the keyword, or “Number” to validate it, helps improve accuracy during recognition.

4.7.3.1 Defining Keywords

Keywords are terms used to locate and correctly identify target data during recognition. When defining keywords, select terms displayed in a consistent location relative to the target data in all the documents to be processed. Keywords cannot be defined before the associated Full text fields are created.

To define a keyword:

1. Run Recognition Designer and select **Tools** > **Free Form Designer**.
2. Select **Settings** from the pane on the left to display the Free Form Designer settings pane.
3. Select an existing full text field or add a new field by selecting **Full-text fields** and clicking **+** on the toolbar. It is preferable to define the **Target data formats** before creating the **Anchor findings**.
4. Select the **Anchor findings** node below the full text field for which **Keywords** will be defined. Click **+** on the toolbar to create an **Anchor finding**, including the **Keywords** and **Associated words** nodes for the full text field.
5. Specify a **Search zone** where the application should look for the **Keywords** and **Associated words**. Select **Full page**, **Upper third**, **Middle third**, **Lower third**, or

a **Custom size (mm)** for the **Search zone**. If required, type a description of the **Anchor finding** in the **Description** text box.

6. Select the **Keywords** node for the **Anchor finding** and click **+** on the toolbar to create a keyword. The keyword **Parameters** pane displays options for defining the keyword. Select from the following options:

   - **Type**: Select **Constant**, **Regular expression**, **Fuzzy regular expression**, or **Field-specific type**.
   - **Value**: Type the keyword or an expression representing the keyword in this field.
   - **Output format**: A regular expression that specifies the format of the output to the operator. You can also reorder the results using groups; the first group is identified with the number 1. For example, you could reorder a date, `MM/DD/YYYY` to `DD/MM/YYYY`. Unrecognized characters (identified by question marks) are included in the appropriate group. You can use this option to replace existing scripting that produces the same output.

     This option is available only with the **Fuzzy regular expression** keyword type.

     For example, you could reorder a date, `MM/DD/YYYY` to `DD/MM/YYYY`:

Field	Regular Expression	Sample
Fuzzy Regular Expression Value	`(\d{1,2})/(\d{1,2})/(\d{2,4})`	`01/31/2015`
Output Format	`{2}/{1}/{3}`	`31/01/2015`

   - **Isolated word**: When this box is selected, the application will expect to find the word or expression both preceded and followed by at least one space.
   - **Case sensitive**: When this box is selected, the application will treat upper and lower case letters as distinct characters.
   - **Automatically fix value**: Enable OCR engines to automatically substitute low-confidence scanned characters with an appropriate alternate character. However, if an appropriate alternate is not available, then a question mark is substituted.

     For example:

     – If a fuzzy regular expression specifies digits only for a zip code (for example, `\d{5}`) and the following conditions are also true:

       ○ A zero is recognized as the letter `O`.

       ○ A zero exists as an alternate character.

       Then, a zero is substituted.

Note: The alternate characters available for substitution are provided by each particular OCR engine; that is, the available alternate characters can vary between OCR engines.

- **Hit threshold (%)**: Specify an accuracy threshold to determine the accuracy required for the keyword to be considered a match.
- **Description**: Type an optional description for the keyword in this text box.
- **Position of target data relative to keyword**: Using the toolbar at the bottom of this pane, add information about the position of the keyword. Click **+** to display the **Position in Relation to Keyword** window and select options that refine the **Orientation** and **Distance** of between the keyword and target data. The toolbar includes options to add, delete, edit, or reorder parameters that define the position of the keyword relative to the target data. Learn more in the section "Selecting a Search Orientation" on page 181.
- **Test**: When the keyword is fully defined, click **Test** to display the **Search Word in Content** window and test the accuracy of the keyword definition.

7. Add additional **Keywords** as necessary to refine the degree of accuracy with which the application identifies and extracts target data.

4.7.3.2 Defining Associated Words

Associated words are used to validate or invalidate keywords identified during *OCR* by evaluating the relationship between the keyword and other terms in the document. Associated words cannot be defined before the associated full text fields and anchor findings are created.

To define associated words:

1. Run Recognition Designer and select **Tools** > **Free Form Designer**.
2. Select **Settings** from the pane on the left to display the Free Form Designer settings pane.
3. Select the **Associated words** node below the full text field being defined. If no **Anchor findings** display, select the full text field and click **+** on the toolbar to create an **Anchor finding**, define the search zone, and create a minimum of one **Keyword** before defining**Associated words**. For help specifying a search zone and defining keywords, see Defining Keywords.
4. Click **+** on the toolbar to create a new associated word. The **Associated words Parameters** pane displays options for defining the associated word. Select from the following options:
   - **Type**: Select **Constant**, **Regular expression**, **Fuzzy regular expression**, or **Field-specific type**.
   - **Value**: Type the keyword or an expression representing the keyword in this field.
   - **Output format**: A regular expression that specifies the format of the output to the operator. You can also reorder the results using groups; the first group

is identified with the number 1. Unrecognized characters (identified by question marks) are included in the appropriate group. You can use this option to replace existing scripting that produces the same output.

This option is available only with the **Fuzzy regular expression** keyword type.

For example, you could reorder a date, `MM/DD/YYYY` to `DD/MM/YYYY`:

Field	Regular Expression	Sample
Fuzzy Regular Expression Value	`(\d{1,2})/(\d{1,2})/(\d{2,4})`	`01/31/2015`
Output Format	`{2}/{1}/{3}`	`31/01/2015`

- **Isolated word**: When this box is selected, the application will expect to find the word or expression both preceded and followed by at least one space.
- **Case sensitive**: When this box is selected, the application will treat upper and lower case letters as distinct characters.
- **Automatically fix value**: Enable OCR engines to automatically substitute low-confidence scanned characters with an appropriate alternate character. However, if an appropriate alternate is not available, then a question mark is substituted.

  For example:

  – If a fuzzy regular expression specifies digits only for a zip code (for example, `\d{5}`) and the following conditions are also true:

    ○ A zero is recognized as the letter `O`.

    ○ A zero exists as an alternate character.

    Then, a zero is substituted.

  Note: The alternate characters available for substitution are provided by each particular OCR engine; that is, the available alternate characters can vary between OCR engines.

- **Hit threshold (%)**: Specify an accuracy threshold to determine the accuracy required for the Keyword to be considered a match.
- **Description**: Type an optional description for the keyword in this text box.
- **Position of target data relative to keyword**: Using the toolbar at the bottom of this pane, add information about the position of the keyword. Click **+** to display the **Position in Relation to Keyword** window and select options that refine the **Orientation** and **Distance** of between the keyword and target data. The toolbar includes options to add, delete, edit, or reorder parameters that define the position of the keyword relative to the target data. Learn more in the section "Selecting a Search Orientation" on page 181.

- **Action of the associated word when it is present**: Select the appropriate option to specify if the associated word validates the keyword, or invalidates the keyword.
  - When **It validates the keyword** is selected, the presence of the associated word indicates that the keyword identified is a valid occurrence of the keyword.
  - When **It invalidates the keyword** is selected, the presence of the associated word indicates that the keyword identified is not a valid occurrence and should be discarded.
- **Test**: When the keyword is fully defined, click **Test** to display the **Search Word in Content** window and test the accuracy of the keyword definition.

5. Add additional **Associated words** as necessary to refine the degree of accuracy with which the application identifies and extracts target data.

4.7.4 Recommendations for Tuning Anchor Findings Settings

This section contains recommendations to use the orientation and distance features to define the position of the target data relative to the keyword and the position of the associated word relative to keyword. By using these parameters appropriately, you can improve detection of the target, solve overlap between keyword and target data format and prioritize between multiple targets as a function of their distance to the keyword. To display the pane with the orientation and distance settings when setting up keywords and associated words, select the **+** button down the **Parameters** pane.

Selecting a Search Orientation

Here are recommendations to define the orientation settings when selecting the search orientation:

- **Full page** is recommended when both the position and the distance of the target with respect to the keyword are unknown so the target needs to be searched in the whole document. This slows down processing. In this case, the **Minimum** and **Maximum** distances are not taken into account.
- **All directions** is recommended when the target is near the keyword but in no specific direction. In this case, the target is searched in all directions with respect to the keyword and within the specified **Minimum** and **Maximum** distances. To reduce the number of candidates found, unselect the option **Accept characters between key and target**.
- **Specific direction** is recommended when both the direction and distance of the target location are known. Select one of the directional arrows. When the selected direction is right or left (so horizontal), the target is found even if it is not perfectly aligned horizontally with the keyword. This enables correct detection even on skewed documents.

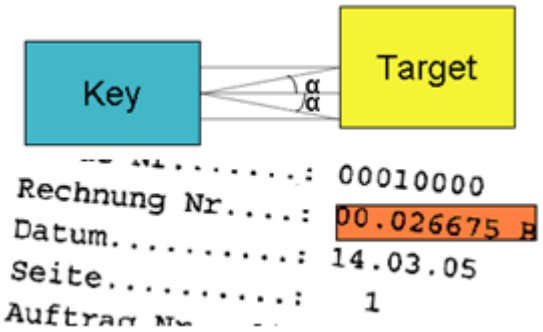


Figure 4-8: Tolerance angle between key and target on a skewed document

- **Accept characters between key and target** is available if you have selected a **Specific direction** that is left, right, up or down. This option is selected by default. This option is recommended to detect the target when the distance from the keyword can be large. In this case it is recommended to set a distance from the keyword that is long enough for the target to be found. If this option is cleared, targets are ignored if there is no blank space between the keyword and the target.

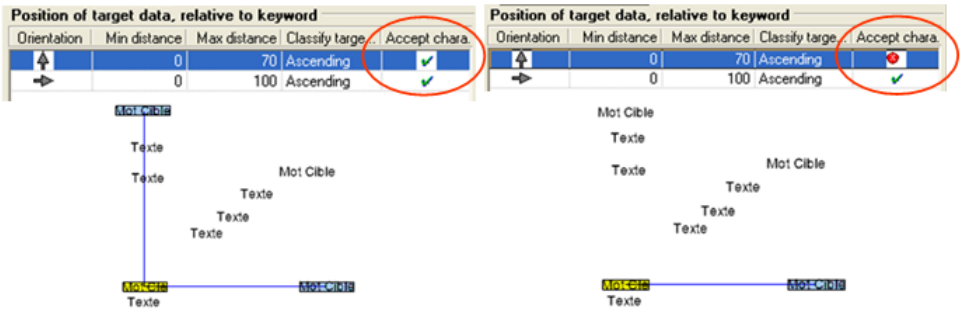


Figure 4-9: Accept characters

Resolving Keyword and Target Data Format Overlaps

This topic explains how to use the distance and orientation parameters to solve possible overlaps between keyword and target data format. When setting up keywords and associated words, select the **+** button down the **Parameters** pane to display a pane with orientation and distance settings.

Authorized Overlap

The maximum and minimum distances (in mm) are calculated from the nearest edge of the keyword text box to the nearest edge of the target data text box. For example, when the target data is located at the right of the keyword in the reference image, the distance is calculated between the left edge of the target data text box and the right edge of the keyword text box. Default values are **Minimum** = 0 and **Maximum** = 50. In the horizontal direction, a small overlap is authorized between the keyword

text box and the target text box enabling the target to be retrieved even if it slightly overlaps the keyword text box.

Solving Overlap

The keyword and the target data may overlap because they are searched using regular expressions that are not specific enough. In this example, the regular expression `(?i)\t?[^\r\n\t]{4,250}` is not specific enough and the keyword "N/REF" (in green) is found as part of the target data (in blue).

When such an overlapping cannot be avoided by setting a more specific regular expression, a solution to retrieve the target data is to set the orientation of the keyword to **All directions** and the **Maximum** distance to 1; this will ensure that the keyword is fully included in the overall target data.

Note: Another solution in this case is to write a script that removes that part of the target data that corresponds to the keyword ("N/REF" in this example).

Prioritizing Multiple Targets

When multiple targets can be found within the minimum and maximum distance interval, it is recommended to prioritize between the multiple targets by setting the distance as recommended in this topic. This option is available to define the position of the target data relative to the keyword. In some cases, the priority is given to the target that is closest to the keyword and in other cases the priority is given to the target that is the farthest from the keyword. In most projects, it is the target that is closest and in some very specific cases it is the one that is farthest.

- Select **Ascending** to keep the target that is closest to the keyword.
- Select **Descending** to keep the target that is farthest from the keyword.

In the following example, there are three target candidates (in the blue frames) found for the keyword (in the yellow frame). The three candidates are 0.01, 817 and 813.28. If **Ascending** is selected, free form will prioritize 0.01 before 817 and before 813.28 to build the hypotheses and so 0.01 has the highest score. If **Descending** is selected, the candidates are kept in the order 813.28, 817 and 0.01 to build the list of hypotheses and so 813.28 has the highest score. The list of hypotheses and their scores display in the **Search Keywords** pane, in the **List of hypotheses** tab.

4.7.5 Defining Full Text Relations

For most projects, carefully defined target data formats and anchor findings are sufficiently precise to produce effective free form recognition. However, if the required results are not obtained using these definitions, free form settings can be further refined using full text relations. Full text relations are based either on field alignment or on rules defined with Visual Basic scripts. The target data format identifies the element found in the document by type. The format can be a date, such as a delivery date or an invoice date, a number, such as an invoice, file, or customer number, or a piece of text, such as a company or customer name. A full text field may have several **Target data formats** with unique parameters defined by type.

- Alignment relations define the relationship between multiple fields based on the relative position of the fields in a given document structure. For example, field "FTfield1" is located to the left of "FTfield2" or field "FTfield4" is located below "FTfield5". Full text relations allow field relationships to be defined with greater precision than the simple orientations defined when creating anchor findings.
- Script relations are used to define a rule to apply to various combinations of field values. A script relation is processed for all the possible combinations of fields in the relation. A score is calculated for each combination; and the sum of scores for all relations gives the score of the combination. After all relations have been processed, the combination that has the highest score is used as the output value. The default score value is 100, but this default can be changed if necessary.

4.7.5.1 Defining a Full Text Relation Based on Alignment

Alignment relations define the relationship between multiple fields based on the relative position of the fields in a given document structure. Learn the fundamentals of full text relation and find recommendations in the section "Using Free Form Rules for Field Extraction" on page 192.

To define a full text relation based on field alignment:

1. Run Recognition Designer and select **Tools** > **Free Form Designer**.
2. In the navigation panel, select **Settings**.
3. Select the **Full text relations** node and click **+** to add a new relation. The full text relations **Parameters** pane appears.

   Note: Spaces and underscores are not allowed in relation names. In addition, the first character cannot be a numeric character.

4. Select **Alignment** as **Relation type**.
5. Leave the **Score** value set to the default value *<100>*, unless there is a compelling reason to change this value.
6. Select **Stop once a relation is confirmed** to stop processing once a combination that matches the relation is found. This combination is preserved as the output

value. Clear this option to process relations for all combinations, regardless of whether or not a matching relation is found.

7. Select **Limit processing time for the relation (ms)** to set a limit on the time allowed to process the relation. The default value is *<1000>* milliseconds (1 second). Valid values are 1 to 99999 milliseconds. Clear this option to allow the application as much time as necessary to process the relation. Users commonly limit the processing time for each image if the definition file results in a large number of hypotheses that take a long time to process.

8. In the **Definition** pane, select an index field from the **List of available fields**, then drag and drop it on the shaded area in the center of the grid. Based on the location of the central field, drag and drop a second field on the grid in the expected position relative to the central field.

9. Set the **Maximum distance (mm)** and **Minimum distance (mm)** in millimeters between the edges of each field. To evaluate the distance between a keyword and its target:

   a. Click **OCR Reading** or **Search Keywords** in the left pane.

   b. Hold down the left mouse button and drag the mouse to draw a frame between the keyword and its target on the image.

   c. Continue to hold down the left button of the mouse and read the values `L` and `H` that appear in the status bar.

10. Create additional alignment relations as needed to obtain recognition accuracy. Do not create more relations than necessary as the time it takes to process many relationships can have a negative impact on performance.

4.7.5.2 Defining a Full Text Relation Based on a Script

Script relations are used to define a rule to apply to various combinations of field values. The combination that has the highest score is used as the output value.

To define a full text relation based on a script:

1. Run Recognition Designer and select **Tools** > **Free Form Designer**.

2. In the navigation panel, select **Settings**.

3. Select the **Full text relations** node and click **+** to add a new relation. The full text relations **Parameters** pane appears.

   Note: Spaces and underscores are not allowed in relation names. In addition, the first character cannot be a numeric character.

   If a relation is renamed in the tree view, only the function declaration is renamed automatically in the script. You need to rename manually any call to this relation in the script.

4. Select **Script** as **Relation type**. The script editor appears.

The following full text relation function displays in the script editor:

```
Private Function Relation1 () As Boolean  End Function
```

5. In the **Fields** pane, click **+** to display and select the fields to put in relation.

   Note: As all the functions are visible in the same script file (BAS), before editing a relation script make sure to select the relation to be edited in the tree structure. Otherwise the **Fields** pane does not update the fields associated to this relation.

6. Write the script defining the relation between fields. Easy Basic scripts created with previous versions of Recognition Designer are automatically migrated to *VBA* when the free form definition file is opened in Free Form Designer. Only *VB* scripting is supported, and all new scripts must be written using VBA. This editor is provided using WinWrap functionality. For specific information about using the editor, the *WinWrap Basic Language* section of the *WinWrap Editor Help* file available from the **Help** menu.

   The type of a relation script is always Boolean. To call a full text field in a relation script, use the object `DpFreeFormFields`. It is the collection of the free form fields associated to the current relation. `DpFreeFormFields` has two properties:

   - *`Item`* (Type = String or Integer): This is a `DpFreeFormField` called by its name or its rank in the **Fields** pane (starting from 0). It is a read-only property.
   - *`Count`* (Type = Integer): This is the number of fields in the collection. It is a read-only property.

   `DpFreeFormField` is one item of `DpFreeFormFields` and has two properties:

   - *`Value`* (Type = String): This is the field candidate value for the relation. It is a read/write property.
   - *`Name`* (Type = String): This is the field name. It is a read-only property.

7. In the **Test Field Value** pane, enter values in the **Input field values** column and click **Run test**. The **Test Result** pane indicates if the relation is successful (green box) or not (red box). If the relation script changes the output data format (e.g. a date format), check the result in the **Output field values** column. Then run the **File definition test** by clicking **Settings** > in the Free Form Designer toolbar, to test all the *DFT* settings, including full text relations.

8. Click **Save** in the Free Form Designer toolbar. The script file *`<DFTFileName>`*`_FFE.bas` is created and takes automatically the name of the associated free form definition file (DFT). It is saved by default to the following path : `/`*`<project directory>`*`/Resources/OCR` where the free form definition file (DFT) is saved. If you rename the free form definition file, also rename the associated script file. Otherwise, the script file is no longer associated to the definition file.

You can include other scripts in the relation script. Although full text relation scripting does not have access to the Recognition Designer Object Model, other BAS files can be referenced by adding the `#Uses` command at the beginning of the script. When referencing other scripts using this type of include statement, included scripts should be saved in the same path as the DFT file.

If the script is modified, and Free Form Designer closed without saving, a prompt appears to save both the `BAS` and `DFT` files. The same goes if another `DFT` file is opened.

If you delete a full text relation in the tree structure or if you change the **Relation type** to **Alignment**, the relation script remains in the BAS file, but is not applied when running the script. You can reactivate the relation script by adding a new relation in the tree structure and renaming it as in the script file.

4.7.6 Versioning Free Form Definition Files

We recommend versioning free form definition files and keeping tracks of changes by entering a description of the definition files. This is useful when testing and comparing the results of several definition files and this is highly recommended when several persons can work on the same free form project. Another advantage is to be able to return to a previous version if necessary, for example, if the test results obtained with a previous version were better than with the current one.

To version a free form definition file:

1. Select **File** > **Options**.
2. Select a number in the **Major** and **Minor** lists.
3. Select the **Summary View** tab and enter a description of the current settings.

4.8 Creating Free Form Templates

This topic explains how to associate the free form rules to the fields created in Recognition Designer and how to define *OCR* settings for the fields.

To create a free form template:

1. Run Recognition Designer.
2. Create a generic template.
3. Select **Index View** and associate an index family with the template by selecting the correct family from the **Index family** list box in the **Template properties** panel.
4. Place the index fields and table fields to be associated with free form rules on the generic template.
5. For each field, populate the **Free Form settings** field with the name of the *DFT* file that contains the free form rules to associate to the field.

Note: When selecting the **Free Form** tab, if the list displays no DFT files or if some DFT files are dimmed, see "Index View" on page 251 for help.

6. Select either an OCR engine for each field associated with DFT files or use the option **Select OCR data from field**. The second option is available only for index fields (not for table fields). In the first case, OCR is run for each field. With the second option, OCR is run on only one field of the free form template and the OCR results are used for all the other fields of the same template. The second option is recommended to speed up processing.

   a. To select an OCR engine for each field:

      - Associate an OCR engine with each field associated with free form rules on the free form template.
      - Place each field on the template so they all cover five times the size of the template image. Full-page recognition will be performed for each field.

   b. To use the option **Import OCR data from field**:

      - Pick up one field among all the fields of the free form templates. This field will be the one on which OCR is run.
      - Associate this field with the DFT file.
      - Place this field on the template so it covers five times the image size. This is to ensure the whole page will be recognized whatever the size of the image in production. If you place the field so as to cover exactly the image size in development, there is a risk that the image in production is not fully recognized if it happens to be larger than the image in development or if images in production come in both portrait and landscape formats. To place the field so it covers five times the image, zoom out the image in the index view until it is small enough so you can place the field and cover five times the image size.
      - Select for this field an **OCR engine** on the **Recognition** tab. Select an engine that supports full-text recognition.
      - Place the remaining index fields on the free form template (that is, on the template image). These fields can be placed anywhere on the image as OCR will not be run for these fields. Associate these fields with a DFT file.

        Note: It is possible to associate a DFT file to the field selected to run OCR that is different from the DFT files associated to the fields that use its OCR results. The only condition to use the option **Import OCR data from field** is that the field selected to run OCR and the fields selected to use its OCR results must belong to the same free form template.

      - For each of the remaining fields of the free form template, select the **Free Form** tab and in the **Import OCR data from field** list and select the field selected to run OCR.

Note: You must not use pre-index fields for importing OCR data. Pre-index fields are populated at the Classification step and not during the Extraction step.

7. Select the free form image filters. This is recommended for most free form projects to improve the accuracy of recognition by OCR engines. Processing time does not increase significantly when images filters are selected.

   a. From Recognition Designer, select **File** > **Project Options**.

   b. In the **Project Options** window, select the **Recognition** tab.

   c. Select one or several filters from the **Free Form image filters** list. A description of the action of each filter is provided in the topic "Recognition Tab" on page 287.

4.9 Understanding Free Form Data Extraction

Aim of Free Form Data Extraction

Free form data extraction enables retrieving a given set of data from wherever the data is located on the document and whatever the data format. For example, data extraction can be used to retrieve an invoice date without knowing the date format or where the date is located on the invoice. The invoice date may be "Dec 12" or "12/12". The date may appear to the left or to the right, in the upper third, lower third, or in the middle of the document. When the position and format of data is unknown, the entire document is considered to be "unknown". Free form data extraction is aimed at capturing data for unknown documents.

Projects Suitable for Free Form Data Extraction

Topics in this section describe the main elements used by Recognition Designer to set up free form data extraction.

Typical projects suitable for free form data extraction are:

- Projects addressing documents with unknown layouts: the difficulty with these documents is that you know which data to extract but do not know where data is positioned. These documents cannot be addressed by graphic templates and zonal recognition. Other projects suitable for data extraction are projects with different document layouts that make it impossible to create as many graphic templates as there are document layouts. For example, if the production flow mostly contains invoices from lots of different vendors, use a free form template (which is a generic template associated with free form rules) to process all invoices instead of creating as many graphic templates as there are vendors.
- Projects addressing only one document type with both known and unknown layouts. For example, if the production flow contains only invoices, create graphic templates for known invoices and use a free form template for unknown invoices. In this case, it is possible to route all unknown invoices to a generic template to which free form rules are associated and which is also selected to be the default template.

- Projects addressing several document types with both known and unknown layouts. Create as many index families as there are document types. For example, if the production flow contains invoices from different vendors but also statements of bank details from different banks, create one index family for invoices and one for statements of bank details. Create one free form template for each document type, so for each index family. To classify the documents, create graphic templates (standard and HPA templates) to classify known layouts and use keyword classification to classify unknown layouts with the free form templates (there is one free form template for each document type).

4.9.1 Understanding Free Form Rules

This section defines the settings that are available in Free Form Designer. Learn more in the section "Recommendations for Designing Free Form Rules" on page 155. Some settings in the interface of Free Form Designer are common to both index fields and table fields, but the overall process is different which is why settings for index fields and table fields are presented in different topics next. There are also two different algorithms for index fields and table fields which are briefly introduced to help you understand the test results and facilitate tuning.

4.9.1.1 Setting Up Free Form Data Extraction: An Example

This section uses an example to illustrate the main steps of data extraction with free form rules. Use free form rules to extract data from an entire page, typically in semi-structured, and unstructured documents in which there cannot be predefined zones. Free form rules apply exclusively to machine printed documents. Extracting handwritten information requires zonal recognition and the creation of graphic templates.

The two samples in this example are bills of lading. A bill of lading is a document covering a shipment. The two samples come in two different layouts as they come from two different consignors. A project typically contains as many layouts as there are consignors. For this reason, there could be up to several hundreds of consignors in a project. In the current example, whatever the number of different layouts or consignors, the data to be extracted is the same for all consignors. The advantage of creating free form rules is that some rules apply to several layouts and are reused many times to extract data. To summarize, you always create fewer free form rules than there are layouts. The more different layouts in the project, the more you can reuse the same free form rules and the more time you save by creating free form rules instead of creating graphic templates. In addition, free form rules can address new layouts entering the document flow.

This section describes the main steps to extract data items common to both samples. In Figure 4-10, data to extract is the "bill of lading number" and the "carrier name".

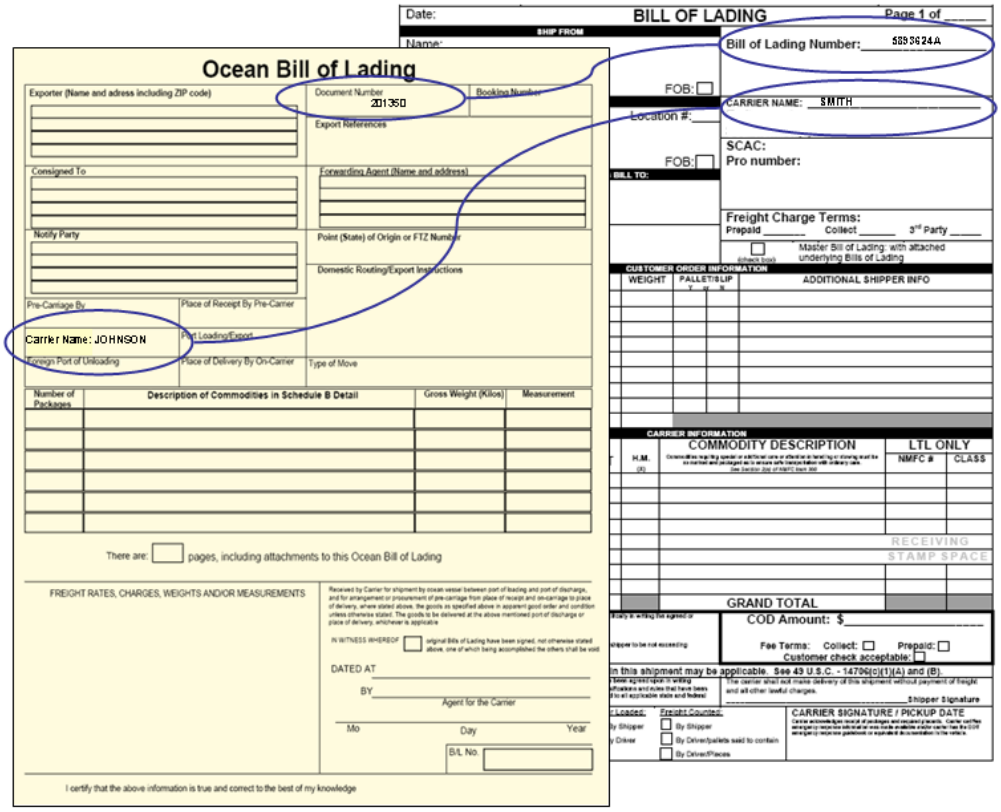


Figure 4-10: Data extraction example

To set up free form rules for both sample bills of lading:

1. Determine which data to extract. In this example, the data to extract is: “carrier name” and “bill of lading number”.

2. Analyze the fields positions, keywords and data formats in the different document layouts and build a keyword list and data format list for each field. In these two samples, the lists of keywords are:

   - Carrier name.
   - Document number, Bill of lading number.

3. In the index family, create one field for each data item to extract. For example: carrier name, bill of lading number.

4. Create one free form template. It is a generic template to which you will associate the free form rules. This template is used to process all bill of ladings whatever the graphical layout. The free form rules are used to detect the data items, whatever keyword is used, and wherever the item is found on the document.

5. In Free Form Designer, in the **Settings** pane, create one full text field for the carrier name and one full text field for the bill of lading number.

6. Develop the free form rules for each field based on the list of keywords.

   - The carrier name can be addressed by the same free form rule in both samples because the keyword is the same ("carrier name") and the target data format can be retrieved by the same regular expression such as [A-Z]{1,20}. This regular expression is a basic example; understand that it is important to consider all possible target data formats and write a regular expression that can cover all the possible formats of the target. Also in this example, for the target data format, set it as an **isolated word**.
   - The bill of lading number requires two free form rules because there are two keywords: "Bill of Lading Number" and "Document Number". The target data format can be retrieved with the same regular expression such as [0–9]{5,9}[A-Z]? which means that the number has always five to nine digits followed sometimes by an uppercase letter. In the current examples, the two formats are "201350" and "5893624A".

7. Save the free form project to a definition (*DFT*) file. Recommendations to organize definition files are provided in the section "Creating a Matrix of Full Text Fields" on page 156.

8. Generate the test *OCR* data as explained in the section "Generating OCR Output Files for Testing" on page 159.

9. Test the free form rules in Free Form Designer, in the **Search Keywords** pane.

4.9.1.2 Using Free Form Rules for Field Extraction

The following settings are used to build free form rules to extract data from fields.

Full Text Field

Build a full text field for each index family field that needs to be recognized with free form rules. Give the full text field the exact name of the index family field. This is required to associate the free form rules to the index family field. A full-text field is composed of two elements:

- *A list of Target data formats*. A target data format answers the question: "What does the information I want to capture look like?". The information is mostly searched by means of regular expressions. Example of regular expressions for two different formats of dates: 10/2/05 and 10 . 02 . 2005: \d{1,2} ?\. ?\d{1,2} ?\. ?(\d{4}|\d{2}).
- *A list of Anchor findings*. An anchor finding answers the question "How to confirm the target data found above is the one I want to capture?". Anchors comprise keywords and associated words. For example, to find an invoice date, the keywords can be: date, invoice date, and billing date. The associated words are optional. For each associated word, you specify if its presence close to the keyword validates or invalidates the keywords. Examples of associated words

for the above keywords can be: due, order, ship and delivery. To detect the location of the target data formats, you can indicate its position relative to the keyword (for example, the target is expected to be to the right of the keyword so the system searches the target to the left of the keyword). You can also indicate the position of the associated words relatively to the keyword.

A full-text field can comprise several target data formats and several anchor findings. For example, the target data format can be a constant or a regular expression. It can also be one or another regular expression. You create several anchor findings if you need several keywords each having a different set of associated words.

Note: If the target data format is complex and requires a long list of regular expressions, it is recommended to use a field-specific type file.

Field-specific Type File

- Field-specific types can be used for target data formats, keywords and associated words. They can be used to combine several regular expressions and constants. The file extension is *TFT*. It is best practice to create a TFT file for each target data format that exists in different full text fields. The same TFT file is reusable for as many fields and as many *DFT* files that contain the same target data format.
- You can use the **Output format** option to specify a regular expression that specifies the format of the output to the operator.

Note: With the TFT file it is possible to modify the output format of the target data formats by means of a script. This is very useful for date fields which usually present variable formats on different documents (for example, Dec, 2, Dec 02 or 12/02). In this case, the TFT file enables, via a script, to standardize the date output value so it is similar on all documents (i.e., same format of output value whatever the original formats on the documents).

Full Text Relations

Relations are not mandatory or necessary in most projects as full text fields are usually sufficient for correct data extraction. Relations are used to avoid doubts or improve accuracy. Relations are of two types: alignment between two fields (A is to the right of B) or a script that uses the values of some fields (B+C=D). Hypotheses are generated from full text relations and are scored. The hypothesis with the highest score is retained as the final hypothesis. A relation requires that all the full text fields to which it applies are saved to the same definition file.

Definition File

A definition file stores the settings of full text fields and full text tables defined in Free Form Designer. It is an *XML* format file, and its extension is DFT. Associate this file with the fields of the template to be recognized with free form rules.

4.9.1.3 Using Free Form Rules for Line Items Extraction

The following settings are available to build free form rules to extract line items from a table.

Full Text Table

Build a full text table for the table that must be captured with free form rules. It is possible to process one table per image. It is not possible to process tables on multiple-page documents. In Recognition Designer, place the table fields on a generic template and associate the free form rules built for the full text table. In the full text table, build as many columns as there are columns to be captured in the table. For example for a dental claim form, create the columns “date”, “procedure code”, “description” and “fee”. In the index family, create as many table fields as there are columns to be extracted in the table. Specify the names of the columns of the full text table with the exact names of the table fields created in the index family. This is required to be able to associate table fields with their free form rules. A full text table is composed of columns. Each column is composed of two elements:

- **A list of Target data formats**. A target data format answers to the question “What does the information I want to capture in the table column look like?” The information is mostly searched by regular expressions. For example, for the column date, use the regular expression `\d{2}[^\n\r\f]{0,2}\d{2}[^\n\r\f]{0,2}\d{2,4}` to retrieve all the date formats.

  You can also use a fuzzy regular expression. For more information, see “Fuzzy Regular Expressions” on page 100.

- **A list of Anchor findings**. An anchor finding answers to the question “How to confirm that the target data is what want to capture?” Anchor findings are optional. They improve accuracy. It is best practice to test the full text field with the target data formats and create anchor findings only if a gain of accuracy is needed.

  An anchor finding is composed of two elements:

  – **Keywords**: Usually, the keywords are the title of the column header such as “unit price”, “quantity” on invoices or “procedure”, “fee” on a dental claim form.

  – **Associated words**: Associated words are optional. They can improve detection of keywords. Define associate words and specify for each associated word if its presence close to the keyword validates or invalidates the keyword. Specify also the direction in which the associated must be searched relatively to the keyword: above, under, to the right or to the left of the keyword. Search can be set to all directions if the direction is unknown.

Field-specific Type File

Field-specific types can be used for target data formats, keywords and associated words. Formats are mostly composed of regular expressions although constants can be used when necessary. A field-specific type file has the extension *TFT*. It is best

practice to create a TFT file for each target data format that exists in different full text fields. The same TFT file is reusable for as many fields and as many *DFT* files that contain the same target data format. With the TFT file, it is possible to modify the output format of the target data formats by using a script. This is very useful for date fields which usually present variable formats on different documents (for example, Dec, 2, Dec 02 or 12/02). In this case, the TFT file enables, through the use of a script, standardizing the date output value so it is similar on all documents.

Order Relations and Script Relations

Relations enable identifying the primary rows. The primary rows contain the data to capture whereas the secondary rows contain additional data. In the following image, the primary row is in the red square and the secondary rows are in green squares. Optionally, the values of the secondary rows can be merged with the primary row values; merged data is separated by spaces. With the example above, merging would result in "005 GENERALPATIO GENERAL (36 X 77 7/8) 121S, WHITE, CHRF, SINGLE, 221S, 7/8 BUG, 824 LINE 5"

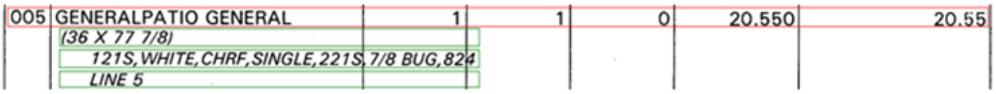


Figure 4-11: Freeform order script relations

- **Order relation**: An order relation defines the combination of columns that is required for the table row to be retained as a primary row. A combination of 2 to 4 columns is recommended. For example, in a dental claim form, "date + procedure code" is a correct combination to identify a primary row. The result is a list of possible primary rows.
- **Script relation**: A script relation defines an arithmetical relation between field values to differentiate primary rows and secondary rows. On invoices, primary rows contain columns such as "quantity", "unit price" and "amount" while the secondary rows contain extra data such as "discount rate" and "tax rate" or extra description. For example, a typical arithmetical relation in invoices can be implemented by the following script relation: `Quantity * UnitPrice = Amount`

Definition File

A definition file stores the free form rules defined in Free Form Designer. It is an *XML* format file, and its extension is DFT. In Recognition Designer, associate the definition file with the fields to be extracted with free form rules.

4.9.2 Understanding Free Form Algorithms

Recognition Designer features two algorithms for free form data extraction, one for full text fields and one for the full text table. The former is also known as free form engine and the latter is also known as line item free form engine (*LIFFE*). This section describes the main steps of the two algorithms. For the free form engine, this section also discusses how the algorithms calculate hypotheses.

4.9.2.1 Understanding the Free Form Algorithm for Full Text Fields

The free form algorithm processes all the full text fields of the definition file and performs the following steps:

1. Searches all the keywords.
2. Searches all the target data formats.
3. Selects the correct target data formats.
4. Builds the list of hypotheses.
5. Calculates the score of each hypothesis.
6. Retains the best hypothesis. The best hypothesis has the score 0 if no relations are applied. Learn more in the section "Understanding Hypotheses Returned by the Algorithm" on page 196.
7. Applies the full text relations if any are defined in the definition file. In this case, it modifies the scores of the hypotheses. The best hypothesis has the score +100 after relations are applied.

Note: If no keywords are defined in the definition file, the algorithm searches for target data formats. If no target data formats are defined, the algorithm searches for keywords.

4.9.2.1.1 Understanding Hypotheses Returned by the Algorithm

The hypotheses for full text fields are derived using a combinatorial analysis of all the hypotheses found for all the elements of the full text fields. A score is calculated for each hypothesis. The best hypothesis, the one that is retained, is the one that combines the best hypotheses for all full text fields. The best hypothesis has the score 0. The other hypotheses have decremented scores, -1, -2, etc. When full text relations are applied, all the scores are increased by the score of the relation and the best hypothesis has the score +100.

When testing free form rules in Free Form Designer, ensure the hypothesis that has the highest score is the one you are looking for.

Here is a simplified example with two fields to illustrate how hypotheses are calculated:

- Field A has three hypotheses: A1 (0), A2 (-1), A3 (-2)
- Field B has two hypotheses: B1 (0), B2 (-1)

A combinatorial analysis of these hypotheses produces the following scores:

- A1 B1 (0)
- A2 B1 (-1)
- A3 B1 (-2)
- A1 B2 (-1)
- A2 B2 (-2)
- A3 B2 (-3)

Scores can also be represented in a table:

Table 4-6: Scores

	B1	B2
A1	0	-1
A2	-1	-2
A3	-2	-3

Note: The combinatorial analysis can lead to a very long list of hypotheses. For example, 3 fields having respectively 3, 2 and 2 hypotheses result in 12 hypotheses and 10 fields with 10 candidates result in a million hypotheses. To speed up processing, use the option **Stop once a relation is confirmed** when defining relations.

4.9.2.2 Understanding the Free Form Algorithm for the Full Text Table Field

The free form algorithm for line items (also known as *LIFFE* algorithm) processes the full text table field as follows:

1. Performs a graphic analysis of the image: the output is a set of segments, grouped by rows. Figure 4-12 illustrates segmenting.

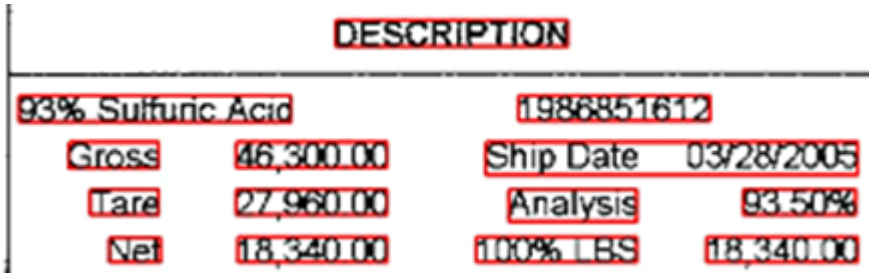
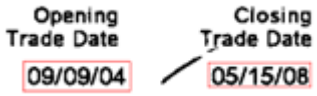


Figure 4-12: Segmenting

2. Searches the target data formats in the row segments: the output is a matching list of target data for each row segment.

3. Identifies the rows that match the primary row definitions: the output is a list of primary rows.

4. Relations are applied to the primary rows. The output is a set of hypotheses for columns positions on each row. There are three types of relations: header, order and script relations. Each time a relation matches, the score of the hypothesis is incremented. The aim of header relations is to detect whether the target data is detected under the column anchor. The following example illustrates the case where the target data "09/09/04" is found under the column anchor "Opening" and the target data "05/15/08" is found under the column anchor "Closing". In this case, it may not be necessary to define order or script relations.



   The score of the header relation is higher than that or order and script relations because header relations are considered more reliable. Header relations are automatically applied by the algorithm. Order and script relations are defined by the project designer. Learn to define order and script relations in "Defining Relations Between Columns" on page 166.

5. Creates a set of columns based on the scores of the hypotheses of column positions on each row.

6. Detects the set of rows related to each primary row: the output is a set of secondary rows.

7. Merges data in secondary rows with data in primary rows. This is an optional option.

4.10 Testing Data Extraction

Testing the recognition properties on different levels verifies the production environment recognition results. Testing on the field level applies only the field recognition properties, while performing a comprehensive recognition testing applies all recognition properties set within the template. Testing on these two different levels can isolate problems more easily.

4.10.1 Testing recognition of an index field

Recognition Designer can test recognition on one field. This type of recognition test also displays *OCR* data for the field.

To test recognition on an index field:

1. Select **Indexing** > **Index View**.
2. Select an index field.
3. Select the **Test** > **Unit Test** menu. The **Field Unit Test** window opens.
4. Select **Test base** > **Load images** and load a selection of images for which you want to launch a test.
5. Select **Launch test**. The test can be performed on all the images or on specific images. Results are shown in four areas:
   - The list of images with their associated test results (to the left). The result is **OK** if the field has been recognized and the contextual rule has been applied successfully.
   - The current image (to the right) with the recognized characters displayed in a dimmed zone on top of the index field (in a green frame).
   - The result of the image after filtering and zoom (to the bottom right). The zoom function has a capacity from 0.25 to 10.
   - The character details (select the checkbox) display the character value and the confidence level.

4.10.2 Testing Recognition of a Table Field

Recognition Designer can test recognition on one field. This type of recognition test also displays *OCR* data for the field.

To test recognition on a table field:

1. Select **Indexing** > **Index View**.
2. Select a table field.
3. Select the **Test** > **Unit Test** menu. The **Table Field Unit Test** window appears.
4. Select **Test base** > **Load images** and load a selection of images for which you want to launch a test.
5. Select the **Launch test** button. The test can be performed on all the images or on specific images. Results are as follows:
   - The upper left list displays the images loaded for the test.
   - The lower left list displays the lines detected for the selected image. These lines appear in a red frame on the current image.

- The current image (to the right) with the recognized characters displayed in a green frame.
- The **Image after filtering** pane displays results after filtering and zoom (to the bottom right). The **Zoom** function has a capacity from 0.25 to 10.

4.10.3 Performing a Template Test

To perform a template test:

1. Select **Indexing** > **Index View**.
2. Select **Test** > **Template Test**. The **Template Test** window appears. Images that will be used for the test are loaded automatically. These are images from the template base and the template reference image (`classifier.tif`). To load other images, select the menu **Test Base** > **Load Images**. To reload images from templates, select the menu **Test Base** > **Load the Template Base**.

   To load images from the template test base, select **Test base** > **Load Template Test Base**. The images are loaded from a directory named `test` in *`<ProjectName>`*`\Models`. For more information, see "Project Structure" on page 39.

   You can also use PDF files as a test base, which also loads their associated OCR data caches.
3. Click **Launch test**. Use the **Zoom** > **Zoom In/Out** menus to adjust image size to the screen and make it more readable.
4. Select **Tools** > **Table Wizard** to test the **Table Wizard** settings.

4.10.4 Tuning Recognition Settings

This section describes the main steps to test and tune the data extraction settings.

To test and tune recognition settings:

1. *Test and tune settings for each document class*. Within a document class, invest time fine tuning the recognition settings and field settings of the first template and then apply as much as possible the same settings across the other templates of the class.
2. *Run unit tests of each field.* Run the unit tests on the template images. The template images originate from the learning image base. In other words, tune recognition settings using the same image bases as those used to create the templates with automatic learning. For each field, ensure the anchor is detected and field value is recognized. Dedicate a significant amount of time to obtain the most accurate *OCR* results on the most critical fields of the images. Run several unit tests to find the best OCR engine settings, the best image filters and the best confidence threshold for the OCR engine. It is essential to obtain correct test results before going to the next step.

3. *Run template tests on the evaluation image base.*Continue to tune the settings to obtain the most accurate OCR results. The results are a good basis to estimate the recognition results that can be expected in production.

4. *Run template tests on the testing image base.* The aim is to decide whether tuning is final or not. Tuning is considered final when the OCR results on the testing base are as accurate as those obtained on the evaluation base. If results on the testing base are less accurate, continue tuning OCR settings on the evaluation base. Then check again the OCR results against the testing base. Always keep the testing base to validate the results; never fine tune settings on the testing base.

4.10.5 Testing Settings in an Capture Process

This section suggests a way to check that the recognition project gives correct *OCR* results when advanced recognition steps are integrated in a capture process.

To test the settings in a process:

1. Deploy the recognition project. Find recommendations in the section "Recommendations for Deploying the Project" on page 48.

2. With CaptureFlow Designer, create a capture process that includes the following steps: ScanPlus, Image Processor, Classification, Identification, Extraction, and Standard Export.

3. Create an export profile to export data to a simple format such as TXT (comma-delimited, one line per page). Then select the profile during Standard Export setup.

4. Organize the testing image base to enable targeted tests: Create folders per document class so that it is possible to run a test on all the folders (general test), or a test on some folders or on one folder (unit test).

5. Test a first document class: Run the production workflow and examine the output data of the Extraction step exported to the TXT file. Ensure the data format is correct. If *VB* rules are used in the recognition project to modify the data format, ensure the rules have applied correctly.

6. Examine the statistics from the Extraction module.

7. If the tests on one document class do not give correct results, stop the tests on the capture flow and go back tuning the settings in the recognition project. If the tests give correct results, run the test on the next document class.

4.11 Defining Index Families (Recognition Designer)

A project can include multiple index families, each describing the fields of a particular document class in the document flow. For instance, if your CaptureFlow is expected to process purchase orders, invoices, and delivery notes, each of these documents needs to be presented in the project with a separate index family.

Index families are designed in Intelligent Capture Designer as *document types*. When creating a document type, you link it to a particular recognition project. When you save a document type in Intelligent Capture Designer, an index family with the same name and set of fields appears in the linked project automatically.

You can view all index families belonging to the project in the Index Family Editor tool integrated in Recognition Designer. You can open any index family, view its fields and field properties, and edit certain field properties.

Editing an index family is not supported in Recognition Designer. Instead, you need to edit the related document type in Intelligent Capture Designer, which includes:

- Adding, renaming, and deleting document types
- Adding, renaming, and reordering fields in the document type
- Copying and pasting fields and field properties inside one document type or between document types

All document type updates are automatically synchronized with the linked recognition project.

To learn more about document types, see *Intelligent Capture Designer Guide*.

To learn more about Index Family Editor and index family editing, read the following topics:

4.11.1 Using Index Family Editor

To view the index families currently available in your project, open the project in Recognition Designer and click the **Index Family Editor** button on the toolbar. The Index Family Editor tool opens in a separate window.

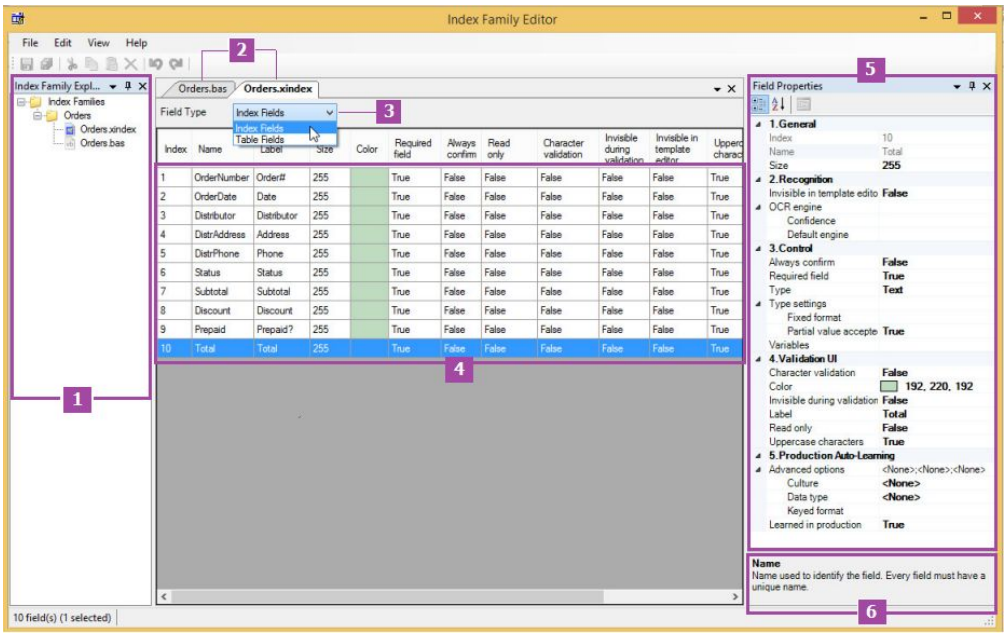


Figure 4-13: Index Family Editor

The **Index Family Explorer** panel displays all index families of the opened project in a tree view (1). Each index family in the tree view includes the following files:

- *<IndexFamily>*.xindex: Includes the definition of the index family, that is, its fields and field properties.
- *<IndexFamily>*.bas: Includes VBA scripting of this index family.

Double clicking any of these files opens the file contents in a separate *<FileName>* tab (2).

When viewing the *<IndexFamily>*.xindex file on the tab, use the **Field Type** list (3) options to switch between the *index fields* and *table fields* of the index family. Select a field in the fields list (4) to view the field properties in the **Field Properties** panel (5). The description of the selected field property appears in the message area (6).

The **Index Family Explorer** panel (1) and the **Field Properties** panel (5) can be undocked or docked to a different side of the Index Family Editor window. Click the panel header and drag it with a secondary mouse button pressed. Release the mouse button to undock the panel, or drag the panel to any of the arrows displayed near each of the window edges to redock the panel. To hide a panel, uncheck it in the **View** menu. To collapse the panel and expand it automatically when the cursor points at the panel button, click the **Auto Hide** icon on the panel header.

Related Topics:

"Editing Field Properties" on page 204

4.11.2 Editing Field Properties

A document type is always kept in sync with the index family in the linked recognition project. This synchronization works in one direction only, from the document type to the index family, and only applies to the set of fi elds, their **Name** and **Label** properties, and how they are ordered in the family (**Index** property). These properties cannot be modified in **Index Family Editor**.

Note: The **Label** property can be modified, however, the next attempt to open the document type in Intelligent Capture Designer will overwrite the changed field with the document type settings.

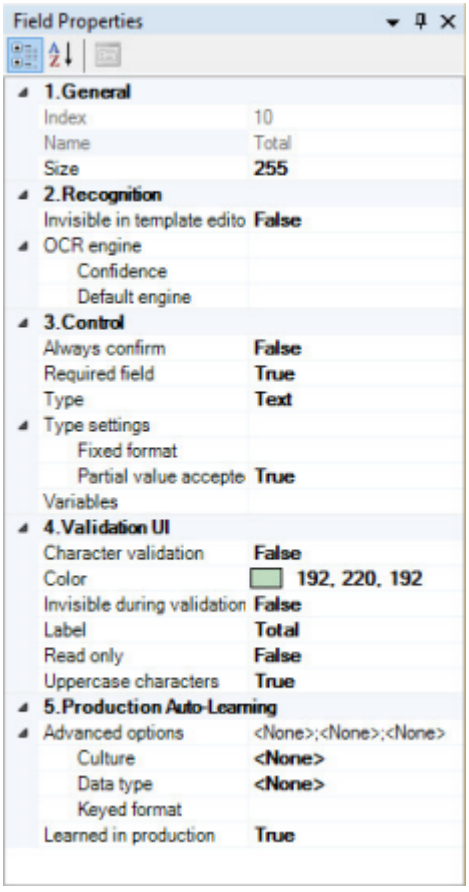


Figure 4-14: Index field properties

Other field properties in the **Index Family Editor** show the default values, irrespective of the values set in the document type. These are the properties in the **General** group (**Size** only), **Control** group, and **Validation UI** group. These settings are editable, but the updated values do not appear in the document type. The reason is that Completion and other advanced recognition modules read the field properties from the document type rather than from the index family. These field

properties still remain in the **Index Family Editor** for compatibility with projects created in earlier versions of the product.

The field settings in the **Recognition** and **Production Auto-Learning** groups are not duplicated in the document type. You need to edit these field properties in the following cases:

4.11.3 Configuring Fields for Manual Field Placement

Before you start placing index family fields on a template, you can configure some fields to be hidden in the **Index View** tool. Also, you can configure a field to use a particular OCR engine when extracting data from the zone.

To configure a field for manual placement:

1. Open the index family in the **Index Family Editor**.
2. Select the index or table field and expand the **Recognition** group in the **Field Properties** panel.
3. Set the **Invisible in template editor** property to **True** to hide the field in the **Index View** tool. To make this field visible, select **False** (default).

   Note: If hidden, the field is still displayed when performing a template test.

4. Optionally, expand the **OCR engine** group and specify the following OCR settings for the field:

   Note: You can also set the OCR engine and its confidence threshold value in **File** > **Project Options** > **Recognition** > **OCR engine for table Free Form fields and textual classification**.

   - **Confidence**: The minimum confidence threshold that a character must obtain to be recognized.
   - **Default engine**: The name of the selected OCR engine. Click the button in the property box and select the engine from the **Select Resources** dialog box.

   Note: You can specify the OCR engine settings for each field later when placing fields on a template.

5. Save the changes.

Related Topics:

"Using Index Family Editor" on page 202

"Recognition Engine Confidence Threshold" on page 110

"Assigning an Engine Configuration File to a Placed Field" on page 110

4.12 Defining Index Families (Dispatcher Manager)

4.12.1 Understanding the Index Family Editor Window

This section pertains to Dispatcher Manager.

The "Index Family Editor" on page 260 window is where index families are created and customized. Index families are composed of index fields, table fields, and scripts that define the data that is extracted during production.

The **Index Family Editor** window can have several families open at one time. Each family or script displays as a tab at the top of the main window. Open families or scripts using the **Index Family Explorer**. Select a tab and the main window displays the indexes or scripts associated with the selected family. Tabs display the name of the family or script, and show an asterisk (*) for modified families or scripts that have not been saved.

The **Index Family Editor** window presents the following elements to enable index family setup and scripting:

Main Window

The main window displays fields in the index family or a script editing window when working with a family or script respectively. Selecting an index family tab displays a table with all the fields defined for the family. Selecting a script tab displays the script editor for creating and modifying family scripts.

- To define or edit index fields or table fields for an index family, select the `<IndexFamily>.xindex` tab. The main window displays the defined fields for the selected family. The **Field Type** specifies either **Index Fields** or **Table Fields**. The index or table fields display in rows, and columns show the defined properties for each field. **Field Properties** are edited from the "Field Properties Panel" on page 262. Drag column headings to change the column display order. Click the column name to change the sort order on the selected column property.
- To work on scripts, double-click a field to open the script or corresponding tab, select the tab for the script to edit, or select the script from the **Index Family Explorer**. When working on scripts, the main window displays a script editor for creating family scripts. The script editor allows Unicode *VB* scripting

  Note: By default, scripts use VB-COM. In scripting windows, users can specify either VB.NET or VB-COM. On the first line of a script, `#Language "WWB-COM"` indicates that VB-COM is activated with some enhancements. Change the line to `#Language "WWB.NET"` to activate VB.NET. Also, if no line starts with `#Language`, VB-COM is activated without `WWB-COM` enhancements. There are differences between the two languages, and VB.NET offers more possibilities than VB.COM. Also, types and available methods can differ between the two. The WinWrap documentation, accessed from the **Help** menu in the script editor, provides more information about scripting options.

Panels

The **Index Family Explorer** and **Field Properties** panels enable field property definition and facilitate navigation to index families and scripts. Display panels by selecting them from the **View** menu. Click the **Auto Hide** icon to minimize the panel and display it as a tab to the left of the window. Mouse over the tab to open the hidden panel.

- The **Index Family Explorer** panel is a navigation tree used to open index families and scripts. The "Index Family Explorer Panel" on page 261 lists all XINDEX files found in the *<Recognition Project>*`\IdxClasses` folder, as well as any scripts files (BAS) defined for the current project. Each family has one index file (XINDEX) and one script file (BAS).
- The "Field Properties Panel" on page 262 panel enables setting properties for the selected index field or table field.

Menus and Toolbars

Menus, available menu selections, and available toolbar options change based on whether the selected tab is for an index family or a script.

- Context menus are also available for some objects in the **Index Family Editor**. When right-clicking an object, the available menu items are made available.
- The toolbar presents commonly used commands. Tool tips describe the function of the button. Although not discussed in many of the procedures, these tools provide shortcuts to many useful commands.

Notes

- The **Delete** menu item and toolbar button are available only when the index family parent node is selected, Renaming, or deleting an index family affects, both the index family and the script file, thus, these actions are only available when the parent node is selected.
- The **Index Family Editor** supports Unicode.

4.12.2 Creating, Modifying, and Saving Index Families

This section pertains to Dispatcher Manager. If you set up and use document types for indexing purposes, index families are created automatically, and you can later modify them in the **Documents** area in Intelligent Capture Designer.

Index families are saved as two files: an XINDEX file and an INDEX file. For each project, these files are located in the *<Project directory>*`\IdxClasses\` folder. When saving index family files, both files are saved with all the index family settings. The XINDEX files are accessed and modified during setup. The INDEX files are used during production. Both files contain identical information for the associated index family. Index family field properties can also be specified after associating a field with a template.

When opened, the **Index Family Explorer** displays all index families associated with the open recognition project. Index family files saved in the *<Project directory>*`\IdxClasses\` folder. If a template is assigned an index family, that index family is displayed in the main **Index Family Editor** window.

Note: Index family files (INDEX) from versions earlier than 6.5 cannot be opened directly in the Index Family Editor.

To create, modify, and save an index family:

1. Click **Index Family Editor** or select **Indexing** > **Index Family Editor** to open the "Index Family Editor" on page 260 window. If a template is selected at this time, the **Index Family Editor** opens with the index family associated with the template.
2. Create an index family, or open an existing one. Whenever a change is made to an index family, the name displays an asterisk (*) indicating that the file has been modified but not saved. When an index family is renamed, the associated folder, index, and script nodes are renamed.
   - To create an index family, select **File** > **New** > **Index Family**. The **New Index Family** window displays a default name `IndexFamily`*<1>*`.xindex`, where *<1>* increments with each new family. The default name takes focus. Type a meaningful name for the family , then click **OK** to create the index family and script files and display them as a new node in the **Index Family Explorer**. The index family is opened in a new tab in the **Index Family Editor** so fields can be defined and field properties assigned.
   - To open an existing index family, expand the family from the **Index Family Explorer** and double-click the *<IndexFamily>*`.xindex` file to open. The family is displayed on a tab in the **Index Family Editor** window. The tab name corresponds to the name of the index family file.

     Note: If you open an index family file from outside the current *<Project directory>*`\IdxClasses\` folder, use **File** > **Save As** and specify the current *<Project directory>*`\IdxClasses\` folder. Otherwise the index family is not associated with the project.
3. Select the index family tab, and select a **Field Type** to create or modify index fields or table fields.
   - For new index families, type **CTRL+INSERT** or select **Edit** > **Add Field** to create as many fields as necessary.
   - For open index families, this operation displays existing index or table fields.
4. If necessary, create a script or modify an existing script. An empty BAS script file is created at the same time as the XINDEX file.

- If the family is open, double click a field or select a field and then select **File** > **Edit Script** to open the script file in the editor. If the family is not open, expand the **Index Family Explorer** and double click the BAS file for the family.
- To create a script for an existing index family, select **View** > **Index Family Explorer**. Expand the tree, and double-click the script file associated with the family. A script editor tab opens, displaying the selected script.

  Although each family has only one family script, additional script can be created and included in the family script. To create a script, select **File** > **New** > **VB Script** and select a script type. For additional help creating and modifying index family scripts, see "Understanding Index Family Scripts" on page 225
- To open an existing script, select **View** > **Index Family Explorer**, expand the tree, and double-click the script. A script editor tab opens, displaying the selected script.

5. Save the file. File names can use Unicode characters.

   - Select **File** > **Save**. The XINDEX and INDEX files are saved in the *<Project directory>*\IdxClasses\ folder.
   - Select **File** > **Save As** to save the family in a location other than the current *<Project directory>*\IdxClasses\. When saved in another location, the node is removed from the **Index Family Explorer** and is not automatically available to the current project.

     Note: Index families saved in locations other than the project IdxClasses folder are still available using the **Open** command. These index families can be incorporated into the current project by selecting **File** > **Save As** and specifying the current project *<Recognition Project>*\IdxClasses folder.

4.12.3 Understanding Index Fields and Table Fields

An index field is a component of an index family that finds specific types of data on a classified document. One index field is created for each data item, and any number of index fields can be created for an index family. A table field is based on a single column of data in a table. One table field is created for each column in a table. Each row of data within a table field (column) is referred to as a line (or cell).

Each index field can be customized to define the type of data to extract, and set up to find data that meets certain specifications. Index and table fields are defined in the "Index Family Editor" on page 260 window and field properties control recognition and validation. When saving an index family, both XINDEX and INDEX files are saved with all the settings of the index family.

Index fields can also be assigned a Free Form Designer definition file. The definition file contains specific instructions on where to locate data on unstructured or semi-

structured documents. Additionally, some fields can be modified automatically by extracting values from an associated xml file, or populating the field using a script.

After creating the fields and defining field properties, there are different ways to associate fields with data on the templates for accurate recognition.

- Position the fields manually
- Place fields using anchors, which improve recognition when images are offset relative to the template. For more information, see "Understanding and Using Anchors" on page 222.
- With large numbers of templates, access the Template Wizard which is useful for "Placing Fields Automatically During Setup Using the Template Wizard" on page 224.

Caution

Too many fields in an index family (in excess of 200) can affect performance.

4.12.4 Creating or Modifying Index Fields and Table Fields

Create and modify index fields and table fields in the **Index Family Editor** window. Click **Index Family Editor** or select **Indexing** > **Index Family Editor** to open the "Index Family Editor" on page 260 window. If a template is selected, the **Index Family Editor** opens with the index family associated with the template displayed.

To create or modify an index field or table field:

1. In the **Index Family Editor** window, create a family or open an existing one.
   - Select **File** > **New** > **Index Family** to create an index family.
   - Select an existing index family from the **Index Family Explorer**. Expand the family node and double-click the `<IndexFamily1>.xindex` file to open the family and display existing fields.
2. Select the index family tab, and then select the **Field Type** and create or modify index fields or table fields.
3. Select **Edit** > **Add Field** or type **CTRL+INSERT** to create a field. Create as many fields as needed for the project, one for each item of data to extract.
4. To define properties for a new field or edit properties for an existing field, select the field from the list in the main window. Select **Field Properties** from the panel. If necessary, select **View** > **Field Properties** to display the panel. Index field properties display when **Index Fields** is selected in the **Field Type** list box. When **Table Fields** is selected, table field properties display.
5. When a field is selected, the properties displayed in the **Field Properties** panel. Specify properties by typing in a text box or selecting from a listbox or dialog box. depending on the property. When applied, changes are reflected in the

main window. For a description of each property and help selecting properties, see "Understanding Field Properties" on page 211.

Several fields can be selected simultaneously and any changes are applied to all selected fields.

6. If necessary, change the order of fields in the list using **Edit** > **Move field up**, **Edit** > **Move field down**, or by clicking the appropriate arrow icons. The order of fields in the list controls the order of recognition and validation activities.

7. Save the index family file. Select **File** > **Save** or **File** > **Save As**. The XINDEX and INDEX files, including field settings, are saved in the folder *<Project directory>*`\IdxClasses\`. Only files saved in this folder are automatically available to the associated project.

Note: When saving index family files, both XINDEX and INDEX files are saved with all the settings of the index family.

Related Topics

"Creating, Modifying, and Saving Index Families" on page 207

"Placing Fields Automatically During Setup Using the Template Wizard" on page 224

"Testing recognition of an index field" on page 199

4.12.5 Understanding Field Properties

This section describes the options available in the **Index Family Editor Field Properties** panel, where properties are defined for each of index and table field. Select an index or table field to display the properties for that field, then click in the panel to modify a setting.

Three buttons on **Field Properties** panel toolbar alter the display of field properties:

- **Categorized**: Properties are displayed grouped by the following categories:
  - **General** properties are basic to all fields, such as field name and field size.
  - **Recognition** field properties improve data recognition and defining the type of data to expect in a field.
  - **Control** field properties influence the status of fields during recognition and what is passed to validation.
  - **Validation UI** field properties control the way the fields are displayed in template tests.
  - **Production Auto-Learning** field properties improve data recognition using automatic detection and placement of fields.
  - **Lines Extraction** field properties improve data recognition when extracting data from tables (table fields only).

- **Alphabetical**: Properties are displayed alphabetically, independent of their categories.
- **Property Pages**: Option not available.

4.12.5.1 Selecting Index and Table Field Properties

This topic describes setting **Field Properties** in the **Index Family Editor**. Properties, grouped by category, appear in the **Properties** panel.

To set field properties:

1. From the **Index Family Editor**, select the table or index field on which to set properties.
2. Set **General** field properties. Select from the following:
   - **Index**: Defines the order in which fields are processed in Extraction or the order in which they are displayed in the template editor. The value cannot be edited, but changes when the user selects **Move Up** or **Move Down** index. The default value is the value of the last index created, plus one.
   - **Name**: An internal representation of the field. This name is displayed in the template editor, but only when the label is empty. The name is used in scripts where events depend on this name and element in `DpDocFields` collection can be retrieved based on this value. Names can consist of:
     - A - Z (upper or lower case)
     - 0 - 9 (except as the first character of the field name).
     - The underscore "_" (except as the first character of the field name).
     - Unicode characters

     The default value is: **Field<*x*>** where <*x*> is a number incrementally increased by one, each time the user adds a field.

     Note: Spaces are not allowed.
   - **Size** (index fields only): Controls the value of the field based on its size. For example, when a value exceeds the field size, the value is stored but the field is set in error. The field size is expressed in number of characters. This option is available only for **Text** type fields if the **Fixed format** option is cleared. For the **Date** or **Amount** field types, the field size is computed automatically.

     The default value is <*255*>.
3. Set **Recognition** field properties. Select from the following:
   - **Invisible in template editor**: If **True** for a field, this field does not display in the Dispatcher Manager indexing view during design, reducing the size of the indexing field list.

Example: Place a field called "year" on a template. Then a list of checkbox options is assigned to this field and the checkboxes display during production, enabling an operator to select the correct year. During production, this "year" field is populated based on which checkbox is selected. It is unnecessary to display this field during setup. Set the **Invisible in template editor** to **True** so the "year" field does not display in the Dispatcher Manager indexing view.

- **OCR engine**: Defines the technology used for reading a zone on the image. The **OCR engine** is chosen depending on the type of info to extract. The engines are named to help in making the selections. For more information on data types and engine selection, see "Recognition Types Supported by Recognition Engines" on page 429 and "Languages Supported by Recognition Engines" on page 427.

  Note: Defining custom settings is enabled by editing an *OCR* engine and saving as an engine configuration file.

  You can also set the OCR engine and its confidence threshold value in **File** > **Project Options** > **Recognition** > **OCR engine for table Free Form fields and textual classification**.

  – In the **Default engine** text box, select an OCR engine configuration file or specify the name of the OCR engine.

  – To configure a default engine, click the browse button to open the **Select Resources** window. In the "Select Resources" on page 274 window, select an engine configuration file either from the **Global Resources** tab or from the **Local Resources** tab. Local resources are the custom recognition files created and saved. The global files are provided with Dispatcher, and cannot be overwritten.

  – **Confidence** aids in detection of characters when conflicts occur. In the text box next to the OCR engine configuration file, specify a confidence threshold value in the form *<X>*, *<Y>*. For more information on confidence levels and understanding how they work , see "Recognition Engine Confidence Threshold" on page 110

4. Set **Control** field properties. Select from the following:

   - **Always confirm**: Used when a field must be checked on all documents and the use of scripting control is inadequate. The operator must confirm the field value even if it has been correctly recognized. When **True**, the message "Confirm your choice" appears in the template test window. The operator must press **ENTER** in the field input box to confirm the field value. This option can also be defined when the field is placed on the template.

   - **Required field**: Requires that the field contains data. When OCR does not read any value, the field is rejected without any script required. When **True**, the message "Field cannot be empty" appears in the template test window when a field contains no data. If it has not previously been recognized, the operator must type the field value. This option can also be defined when the field is placed on the template.

- **Type** and **Type Settings** (index fields only): Provides basic control on the field value. For example, if **Date** is selected, the field is rejected if its value does not match the selected date format.

  Select the valid format for the field. The available types are:

  – **Text** recognizes text characters. When selecting **Text**, expand the **Type Settings** option to set:

    ○ **Fixed Format** which allows user specified formats.

    ○ **Partial Value Accepted** accepts partial format matching.

    For more information on fixed formats and partial formats, see "Understanding Text Field Fixed Formats" on page 219.

    Note: When using the **Text** field type and not using the **Fixed Format** option, the field has a default size of <*255*> characters. Set a different size by entering a new value in number of characters in the **Size** field.

  – **Date** specifies a date format. When selecting **Date**, expand the **Type Settings** option to set:

    ○ **Format**: Enables selection of numerous predefined date formats from the drop-down list.

    ○ **Limit Type**: Can be either **Fixed** or **Floating**.

    ○ **Limit Type Settings**: Enables setting the parameters for **Fixed** or **Floating Values**. For **Fixed**, select fixed dates by typing the dates or selecting them from the pop-up calendar for the **Maximum** and **Minimum** dates. For **Floating Values**, set a **Lower Limit** and **Upper Limit** (in number of days), based on an interval relative to the processing date. A value of 0 corresponds to processing date. Negative values indicate days in the past. Positive values indicate days in the future and can represent, for example, payment due dates.

  – **Amount** specifies a value. When selecting **Amount**, expand the **Type Settings** option to set:

    ○ **Decimals**: Sets a required number of digits after the decimal point.

    ○ **Minimum** and **Maximum**: Sets the limits for the amount.

    ○ **Separator**: Specifies the type of separator to use when formatting the amount.

    ○ **Units**: Sets the maximum the number of digits. For example, type *<5>* and the number of digits can range from one to five. So, for example, *<1>*, *<5.95>*, *<4,000>*, and *<99999>* are all valid, but *<100,000>* is invalid when digits are limited to five.

  Note: During production, a field value cannot contain a question mark ("?") because the question mark is a reserved character for internal use.

Any fields containing question marks are passed to Completion the same as any field containing unrecognized characters. The question mark cannot be entered again during the validation step. These characters must be removed or replaced by another unreserved character.

- **Variables**: Associates variables with field. This association is defined by a script when a control varies depending on the associated template. Variables for the current document are initialized with the values from the associated template.

  – To define variables, click the **Variables** browse button to display the **Variables Editor**. Click **Add** to add new variables and edit the names in the **Variable** properties.

  Example: In the **Indexing View**, set a value of **True** for the variable on several templates where a control is applied. Leave the variable empty on other templates. Inside the control script, specify that if the variable value is equal to `True`, apply the control. If the value is not equal to `True`, do not apply the control.

5. Set **Validation UI** properties. Select from the following:

   - **Character validation**: Defines how the field is validated. If **True**, the operator validates a value character-by-character. If **False**, all the information is selected and the operator validates the value as a whole. Character validation is appropriate when the number of characters read is expected to be correct, and only a few characters are expected to be false. Under these circumstances, validation will be faster. On the other hand, when the number of characters is incorrect or too many characters are invalid, it can be faster to key the entire field.

     If **True**, Dispatcher Manager sets the focus on the first character requiring correction, within the field, so that the operator can correct the character and move to the next character requiring correction. To facilitate navigation, select the **Go to next field automatically** option.

     For more information, see "Understanding and Enabling Character Validation" on page 220.

   - **Color**: Used to give visual information about the meaning of a field, and colors can be used to categorize information. For example, all "Bill To" information can be assigned the same color for enhanced visual recognition. Select the color for the field to appear in **Indexing** view, and Identification.
   - **Field Width** (table fields only): Sets the field width, in pixels, of the column displayed in the template editor. Set field width based on the type of data to be displayed in the field. For example, an item quantity generally has a smaller width compared to the item description. Correctly setting this value results in more efficient use of validation user interface space.
   - **Invisible during validation**: Useful for fields which contain internal values used for computation, and which need to be hidden from operators. When **True**, the field does not appear when performing a template test. Also, pre-indexed fields with this setting will not display in Identification.

- **Label**: Provides for more descriptive information than the **Name** property. When performing a template test, the **Label** is displayed if provided, rather than the **Name**. **Name** has a limited set of allowed characters and is used for scripting, where shorter names are preferred. Use labels to provide a description of the field, which can be useful during validation.
  - Type an optional label for the field. There is no restriction on the characters used for labels. The label identifies input boxes for fields in Identification.
  - If a label is not specified, the fields use the field name.
- **Read only**: If **True**, the field appears in the template editor but the operator cannot change the field value, only copy the field value.

  Note: If a read-only field is in error, the operator must reject the document because the field cannot be corrected.

  Even if **Read only** is **True** for the field, a script can still use the `ReadOnly` property of the `DpDocField` object.
- **Uppercase characters**: If **True**, creates uppercase characters during recognition, when all lowercase letters are transformed to upper case. During validation, all typed values are written as upper case, even if the operator types the letter in lower case.

6. Set **Production Auto-Learning** field properties. During learning, the `@<fieldname>` setting is interpreted and replaced by the according value from an XML file. If the value of an index field is empty, or the field does not exist, a warning naming the field for which no value is found is sent to the log. This message is a warning only and no error displays in the Windows Event Manager. Learning continues but the search of the reformatting hypotheses is skipped for remaining documents.

   Example 4-1: Culture Field Property Populated Automatically

   In the recognition project, the **Culture** property is set to **@IdxCulture**, where “IdxCulture” is a named field in the associated index family.

   During production, this field is populated by reading the value of the “IdxCulture” field for the current image in the associated xml file.

   - Image1 is classified with template “FR” and associated with and index family containing a field named “IdxCulture”.
   - Image2 is classified with template “US” using the same index family as Image1.
   - During recognition, the “IdxCulture” field is automatically populated by scripting based on the associated classification template.

     Thus, for Image1 the value of “IdxCulture”=“fr-FR”. For Image2, “IdxCulture”=“en-US” During Learning, @IdxCulture

Select from the following Production Auto-Learning properties:

- **Advanced Options**: This property groups subproperties based on the field type selected. It displays advanced settings that enhance specific data type detection.

  This is a read-only field that displays selected subproperties separated by a ";" and without spaces. For example, in the format : "Date;FR,".

- **Culture**: Improves data type detection by specifying a country/language. The default selection is all available countries. Country selections are defined in the `Autolearning.rules` file.

  To define the **Culture** setting at runtime, populate this property with the name of a field in the index family used to extract regional settings. Prefix the field name with the `@` symbol in the **Culture** property field. For example, `@<FieldName>`.

  If the field specified does not exist in the index family, an error displays.

- **Data type**: Improves data detection by searching for a particular data type. Options include **None**, **Date**, **Amount**, **Rate**, and **Phone**. **Data type** selections are defined in the `Autolearning.rules` file.

- **Keyed format**: Defines a specific format to improve data detection based on the selected **Data type**.

  Like the **Culture** property, the **Keyed format** property can be populated with `@<FieldName>` and populated at runtime with values extracted from the specified field.

  If the field name specified in the **Keyed format** property field does not exist in the associated index family and the project designer tries to save the project, the index family is not saved and an error message displays.

- **Learned in production**: Set to **True** by default, this property enables *PAL* to position index and table fields on templates created with automatic learning. To disable this functionality, select **False**. This option is useful for when it is not necessary for PAL to search for and place certain fields. For example, populated fields, since they are not read.

  Note: PAL does not manage *OMR* and barcode field types.

7. Select **Lines Extraction** field properties (table fields only). Select from the following:

   - **Border Detection**: If **True**, detects the edges for both **Right/Left** and **Top/Bottom** sides of the table field. When templates contain borders, Dispatcher Manager automatically sizes fields to incorporate all table data. When reading the table field, if border lines are detected, the width and height are adjusted in relation to the borders.

   - ***Detect all lines***: When set to **True**, all lines including those without a value are detected. When this option is cleared, only lines having values are

detected. For more information on line detection, see "Table Recognition: A Simple Example" on page 144.

- **Line Height**: Used to set the limits for font size during recognition and improve the accuracy of detected lines. Expand **Line Height** to specify **Maximum** and **Minimum** height values in millimeters. Any text height outside the range of the specified values is not detected.

  Note: If the minimum is low and the maximum is high, the number of lines detected is higher but also the number of incorrect lines.

- **Paragraph Settings**: Paragraphs are groupings of text in a table. Paragraph headers define the beginning of a paragraph in a table, and all following text is part of the paragraph until the next paragraph header is detected. It is possible to read and display only the first line of the paragraph (that is, the header) or the header followed by all the article lines inside the paragraph. The next paragraph begins at the next header. For help using paragraph settings, see "Table Recognition: A Complex Example" on page 146.

  The available options are:

  – **Field Status** determines the type of table field. If **Linked to Paragraph Field** duplicates paragraph headers to empty lines, so the value of the first detected field is repeated on each line of the paragraph. With **Paragraph Field**, the current table field is used to detect all paragraphs in the table fields, and activates the **Headers Only** and **Keywords** options. Only one field is used for paragraph detection. **<None>** disables **Paragraph Settings** for this field.

  – **Headers Only** when set to **True**, only displays the header line of a paragraph.

  – **Keywords** defines the keywords to identify the paragraph headers. Click the browse button to display the "Keyword Editor" on page 270 window.

  Note: If the table contains lines of text before the first paragraph header is found, all the lines found before the header are read and displayed, since the detection of paragraphs mode starts effectively from the first header found.

4.12.6 Understanding Text Field Fixed Formats

Fixed formats control the order and type of characters validated for a field. If a field value does not conform to the fixed format, it is not valid. Fixed formats are defined using a combination of numbers and designated characters. The designated characters are:

- <*A*>: Represents letters only (a-z, A-Z, no accent allowed).
- <*N*>: Represents numbers only (0–9).
- <*X*>: Represents any alphanumeric character (a-z, A-Z, 0–9).
- <*C*>: Represents alphanumeric and special characters (all characters, including accents, special characters, and Unicode characters).

Creating Fixed Formats

Placing a value in front of the characters indicates the number of that character type allowed in the string. And any combination can be defined. For example, the format <*4N2A3X5C*> is interpreted in this way:

- <*4N*> represents four numeric digits, such as <*2345*>.
- <*2A*> represents two alphabetic characters, such as <*MN*>.
- <*3X*> represents three alphanumeric characters, such as <*2BC*>.
- <*5C*> represents five characters including special characters, such as <*#456#*>.

Based on this syntax, <*2345MN2BC#456#*> is an example of a valid format.

Partial Fixed Formats

Partial values for fixed-format fields can also be accepted. For a partial format to be recognized as valid, the number and type of characters recognized in the field must be equal to or less than the defined format. Characters are read individually from left to right, and each format designation must be met from left to right to be valid. For example, consider the format *<3A2N4A>*. For partial format acceptance, <*3A*> must be met before <*2N*> is considered. Similarly, <*3A2N*> must be recognized before <*4A*> is considered. Valid partial formats for *<3A2N4A>* could include:

- <*XY*>
- <*XYZ3*>
- <*XYZ34BC*>

Valid formats would not include:

- <*X4*>
- <*XYZA8*>
- <*XY5BC*>

4.12.7 Understanding and Enabling Character Validation

Dispatcher Manager provides validation so that the operator can correct individual character recognition or entire fields during validation. Individual character validation is appropriate when the number of characters read is mostly correct and only a few characters are misread. When the number of characters recognized is incorrect, or many characters are invalid, it can be faster to correct the entire field.

To enable character correction:

1. Click **Index Family Editor** or select **Indexing** > **Index Family Editor**. If a template is selected, the **Index Family Editor** opens the index family associated with the template.
2. Open the index family containing the field requiring character validation from the **Index Family Explorer**.
3. Select **Index Fields** or **Table Fields**, based on the **Field Type** requiring validation.
4. Select the field for validation. Select **Character validation** options from the **Validation UI** area of the **Field Properties** panel. Set to **True** for the operator to validate unrecognized characters. If **False**, all information is selected and the operator validates the entire field.
5. Save the index family and close the **Index Family Editor**.
6. Select **File** > **Save** to save the project with the new index family settings and project options.

4.12.8 Assigning a Default Value to a Field

When creating a project, assign a default value to a field when no *OCR* engine is selected for this field or if no value has been returned after recognition. If no value is returned after recognition and a default value is not provided, the operator will have to manually validate a value in the field. Assigning a default value prevents an empty value being returned.

To define a default value for a field:

1. Click **Index View** from the main toolbar.
2. Select a template from the template list (to the left of the main interface).
3. Select the field to define a default value for from the field list (to the right of the main interface).
4. In the **Field Properties** (at the bottom right of the window), enter the default value in the **Default value** text box.
5. Press **ENTER** to validate the default value.

4.12.9 Creating Indexing Templates

During project setup, users create index families and associate them with templates. The index and table fields are then placed on the classification templates so these templates become indexing templates. After fields are placed on the templates, the initial settings created in the index family can be fine-tuned to process each indexing template accurately.

The following methods are useful for setting up and placing index and table fields on templates or documents, either manually or automatically:

- "Manually Positioning Index and Table Fields on a Template" on page 221
- "Understanding and Using Anchors" on page 222
- "Placing Fields Automatically During Setup Using the Template Wizard" on page 224

During production, there are times when table fields are not correctly recognized. In Completion, placement of the unrecognized table fields can be accomplished manually, although the use of the Table Wizard is generally a more efficient placement method.

4.12.9.1 Manually Positioning Index and Table Fields on a Template

After creating index and table fields, place them on the template for accurate recognition. This section explains how to position fields on templates and how to change the default template image.

Using anchors when placing fields on templates can improve recognition when images are offset relative to the template. For more information, see "Understanding and Using Anchors" on page 222.

When working with large numbers of templates, use the Template Wizard for "Placing Fields Automatically During Setup Using the Template Wizard" on page 224.

To position a field:

1. Associate an index family containing defined fields to detect with the template:
    a. In the **Classification View**, select the template from the template list. The template properties display in the **Template properties** panel.
        - If the **Template properties** panel is hidden, right-click the template in the template list and select **Template Properties** from the context menu to display the panel.
    b. Select one or several templates from the template list.
    c. Select an index family from the **Index family** listbox in the **Template properties** panel, associating the index family with the selected templates.

2. Place the field on the template:

   a. In the main window, select **Index View**. The right pane displays the fields from the associated index family.

   b. Drag and drop the field box on the template image at the approximate field position. For zonal recognition, adjust the size of the field using the handles so that the field box covers exactly the field value. For free form recognition, extend the field to cover the whole search zone. This zone corresponds to the search zone defined for testing the free form settings. More information on free form settings is provided in "Designing Free Form Rules" on page 155. For table fields (column fields), adjust the field so that it covers the entire column.

3. For standard templates, it is sometimes necessary to change the default index image. During automatic classification, Dispatcher Manager selects an image for the template. This image is displayed as the default image in the **Index View**. If necessary, you can change the default image to a more appropriate one:

   a. Click **Index View**.

   b. Select a standard template from the template list and right-click.

   c. Select **Change the Index Image** to change the default image in the **Index View**. This option is dimmed if the selected template is not a standard template.

   d. Select the image to use as a replacement. This image must have the same format and resolution as the current index image.

   Note: To change the index image of an *HPA* template, see "Editing HPA Templates" on page 72.

4.12.9.2 Understanding and Using Anchors

Anchors are optional features used to define relative positions of fields on a template, as opposed to absolute positions on a template. When placing a field on a template, that position becomes absolute in terms of the template. If images in a production environment are offset relative to the template, due to scanning issues for example, the index position can be inaccurate for those images and recognition errors can occur. An anchor can be placed on a template relative to the field position. Anchors are placed on a static piece of information, like a logo or mark, that appears on all images in the same relative position to the index location. If one or more of the images in a process are offset relative to the template, the anchor enables Dispatcher Manager to find the static piece of information, determine the index position defined relative to that anchor, and extract the correct information.

There are two types of anchors that can improve field detection: Global anchors and local anchors. There are two global anchors per template and one local anchor per field.

- *Global anchors* are useful for adjusting the overall position of the image, such as when an image scanned in production is not in line with the reference image.

Global anchors are placed at opposite corners of a template image on unique and recognizable marks. In most cases, global anchors are sufficient.

- *Local anchors* are placed relative to each index or table field. Local anchors can improve detection of fields when global anchors are not accurate enough. They can also be used to locate fields whose position is independent from the overall document structure.

Set the following anchor properties when creating anchors:

- **Search Zone**: Defines how far from the center of the anchor to look for the object targeted by the anchor. The size of the anchor itself defines the extent of the search. When an image is evaluated during production, that image can be offset up to the search zone distance and still be valid. Define both **H** (vertical offset) and **W** (horizontal offset) in millimeters.
- **Graphic matching threshold**: The graphic matching threshold is set by moving the slider at the right of the sample anchor image. The threshold determines how closely the document image must match the anchor image. If scanning results in minor variations in the quality of the image, an exact match is not required when setting thresholds below 100%. The default of 50% is useful in most situations.
- **Text matching threshold**: The text matching threshold is set by moving the slider at the right of the sample anchor image. The threshold determines how closely the document image must match the anchor image. If scanning results in minor variations in the quality of the image, an exact match is not required when setting thresholds below 100%. The default of 80% is useful in most situations.
- **Conditional anchor** and **Field substitution value** (index fields only): Select **Conditional anchor** when the anchor image data is not found. In this case, the **Field substitution value** is displayed in the place of the index field information. This property is not available for table field or global anchors.

When more precise anchors are necessary, such as when the images in a batch are closely related or have relatively minor variations (such as checks from different banks), consider using High Precision Anchors (HPA).

To place anchors on an image:

1. In the **Indexing View**, associate the index family containing the fields to detect with the templates. The fields appear on the right and the anchors are displayed.
2. Place anchors on the image.
    a. To place **Global anchors**, drag and drop them on opposite corners of the image over unique image patterns that occur in every image such as title text or a logo.
    b. To place local anchors, place each field on the image and then drag and drop the associated index anchor on the image in position close to the field. The anchor position must be a unique image pattern that occurs on all images.

3. Adjust the size of the anchor with the handles. Results are improved when small anchors are placed on a highly contrasted pattern present in the same relative location on every document.

4. Specify the anchor properties or accept the defaults:

   - Set **Search zone** values that define how far the offset can be for images during production.
   - Set the **Graphic matching threshold** and **Text matching threshold** to correct for minor variations in image quality.
   - For index fields only, select the **Conditional anchor** and define a **Field substitution value** for instances when anchor data is not found.
   - Enter the anchor text in the **Anchor Text Value** field.

5. To verify the placement of anchors, perform an anchor test:

   a. Select an anchor, then select **Test** > **Unit Test**. The **Anchor Unit Test** window appears.

   b. Select the images from the image base to use for the test.

   c. Click **Launch test** to run the test on all the images, or select an image from the list to run the test on one image only. If the anchor is found, it appears in a green square, otherwise in a red square. The **Graphical Result** column shows graphical matching accuracy for the selected threshold value, and the **Text result** column shows textual accuracy compared with 80% default threshold value. Modify the anchor properties and run the test again.

4.12.9.3 Placing Fields Automatically During Setup Using the Template Wizard

With large numbers of templates, use the Template Wizard to facilitate placing each field. Based on free form settings defined in Free Form Designer, the Template Wizard can run a full text search to place fields automatically on a selected set of templates. Several steps are required before running the Template Wizard to place fields.

To place fields automatically using the Template Wizard:

1. From Free Form Designer, create **Full-text tables**, **Full-text fields**, or both.

2. Define the free form settings using only **Anchor findings** and **Keywords**. Other properties, such as **Associated words** and **Target data formats**, are not taken into account by the Template Wizard.

3. Save the settings to a *DFT* definition file.

4. Associate the index family containing the defined fields with the templates on which the Template Wizard runs:

a. In the **Classification View**, right-click and select **Template Properties** from the **Template** list. The **Template Properties** pane appears at the bottom of the main window.

b. Select one or several templates in the **Template** list.

c. Select an index family from the **Index family** list to associate the index family with the selected templates.

5. With the field selected, click the **Free Form** tab at the bottom of the anchor panel, and associate the definition file containing the free form settings with the fields.

6. Run the Template Wizard. The Template Wizard uses the free form settings as follows:

- Index fields and table fields are placed on the document based on the free form settings. Table fields are placed to cover the height and width of table columns.

  Note: The columns in the document must have full lines. Dotted lines or columns without separation lines are not recognized.

- If used, local anchors are placed on the detected keywords if the definition file uses anchors. If placing local anchors manually before running the Template Wizard, and the definition file uses anchors, the Template Wizard updates the position of the local anchors. The Template Wizard does not place global anchors or modify existing global anchors on the templates.

4.12.10 Understanding Index Family Scripts

An index family script defines controls that help determine how objects are handled during recognition and validation. Each index family contains a single script. Additional scripts used by an index family are included in the index family script as inclusions in the index family script. A best practice for complex scripts is to use the index family script as the main script, and call additional scripts as inclusions.

Scripts are listed in the **Index Family Explorer** under the **Index Families** node. Only the index family script is defined and listed. This file is located in the `<Project directory>\IdxClasses\` folder, in the same location as the index families

Some properties related to recognition and validation, such as field size, required field, or field format can be set without scripting using index or table field properties. Although these settings provide minimal control over fields, most projects require more sophisticated controls enabled through an index family script. In the index family script, objects are elements, such as index families, index fields, and table fields, used to perform data extraction.

Scripts are event-based. The list of events available to index family scripts is provided in the *Dispatcher Event Model* section of the *Scripting Guide*. Each module has different events available, and multiple events are available for each module. For example:

- In Recognition, a script action can bypass field recognition before it is processed using the *<Field1>*_BeforeRecognition event or control the field value after recognition using *<IndexFamily>*_DocumentControl.
- In Completion, a script action can present valid values to an operator in a selection box when a field takes focus by using the *<Field>*_Enter event, or to check a value when an operator changes it using *<Field>* _ValidateValue.

Together with the properties assigned to them, objects define the data to extract from the documents in a batch and what to do with the data during production. Because index family scripts are associated with an index family, all documents in a batch associated with the index family use the script. Scripts can therefore access all fields and properties for every document in a batch. Whereas events trigger actions, methods enable a specific action to be performed on a specific object, such as adding or deleting a row from a table field. For example, one script could populate some fields and another script could read the data and use it to trigger subsequent events. The *Dispatcher Object Model* section of the *Scripting Guide* describes all available objects and the properties, methods, and events for each object.

A simple and common use of a script is the computation of expected field values. For example, consider an invoice where the "Subtotal" field captures the pretax amount of the invoice, the "Tax" field captures the amount of tax collected, and the "Total" field captures the final invoice amount including tax (Subtotal+Tax=Total). A script can perform this calculation on each document, sending to validation only those documents where Subtotal+Tax do not equal the captured Total amount.

The *Scripting Guide* provides detailed diagrams of the *Object model schema* and *Event sequences*. *Matrix tables* of methods, components, events, objects, and parameters are also provided. The guide provides information and instructions for customizing as well as recommendations for writing good code. The *Examples Using VBA Scripts* section demonstrates some common uses for scripts appropriate syntax.

By default, scripts use VB-COM. In scripting windows, users can specify either VB.NET or VB-COM. On the first line of a script, `#Language "WWB-COM"` indicates that VB-COM is activated with some enhancements. Change the line to `#Language "WWB.NET"` to activate VB.NET. Also, if no line starts with `#Language`, VB-COM is activated without `WWB-COM` enhancements. There are differences between the two languages, and VB.NET offers more possibilities than VB.COM. Also, types and available methods can differ between the two. The WinWrap documentation, accessed from the **Help** menu in the script editor, provides more information about scripting options.

4.12.10.1 Creating or Editing an Index Family Script

The following procedure provides general instructions for creating or editing an index family script. For more information on editing an index family script, see *Editing an index family script* in the *Scripting Guide*.

To create or edit an index family script:

1. Select **Indexing** > **Index Family Editor**. The **Index Family Editor** window displays.
2. Create an index family, or open an existing family where a script is implemented. Create fields and properties for the family as required. For more information on creating fields, see "Creating or Modifying Index Fields and Table Fields" on page 210.
   - When an index family is created, both the XINDEX file and the associated BAS script file are created and display in the **Index Family Explorer** tree.
   - All index families defined for the project are listed in the **Index Family Explorer** tree pane. Expand or collapse nodes in the tree to view associated XINDEX and BAS files where index family information is stored. To display the index family fields, double-click the associated XINDEX file from the **Index Family Explorer** tree. To display the family script, double-click the associated BAS file.
   - When the family script is displayed, or a script file is created, a Visual Basic (*VB*) script editor opens as a new tab in the detail pane and the **Index Family Editor** menu bar displays options for working with scripts, including **Macro** and **Debug** menus.
   - By default, index families are named using an incremental numbering system. For clarity, rename the family with a meaningful name.
   - Index families and the associated index family script are located in the `<Project directory>\IdxClasses\` folder.
3. In the **Index Family Explorer** panel, expand the tree to display nodes for the index family (XINDEX) and script (BAS) files. Double-click the BAS filename to display the family script in the script editing pane. Begin creating or editing the associated script. Type or edit the script in the script editing pane.
   a. Select an object from the **Object** list box to add an event for the object. Available objects include index families, index fields, and table fields, as well as a main table. The objects available in the **Object** list box are the objects defined for the index family.
      - Select field and table field objects to modify their behavior.
      - Select the index family object to modify the current index family behavior.
      - Select the main table object to trigger an event before or after the insertion or deletion of a row.

See the *Scripting Guide Using Events* section for help on selecting an event. The *Using scripts* section introduces and provides examples of scripting. For more information on selecting an object type, see *Object Types* in the *Scripting Guide*.

b. Type the script text, or select an available event from the **Proc** listbox to insert the event script automatically. Available events differ based on the object selected. Events customize recognition and validation in an index family.

- When an object is selected, all the events available to that object are listed in the **Proc** listbox. Events present in the script are displayed in bold text.
- Selecting a function from the **Proc** listbox inserts the event at the end of the current script. If the event is already present in the script, selecting it from the listbox places the cursor inside the correct subroutine.
- When the **(General)** object is selected from the **Object** listbox, all implemented events plus all user subroutines display in the **Proc** listbox.

4. Add the required events for each object type to the script. For common functions or lengthy and complex scripts, consider creating separate scripts and referencing them by inclusion in the family script.

Note: Index family events must remain in the `<IndexFamily>.bas` file. Scripts referenced through an inclusion must contain only objects and associated functions.

Do not include a project script directly in an index family script. To perform functions common to project and index family scripts, write a separate script and save it to the `\Resources\Scripts\` folder. Then include the script in a project or index family script with the `#Uses` command.

5. Save the script using the **Save**, **Save All**, or **Save As** options on the **File** menu. The name of the script file plus the BAS extension corresponds to the index family and is saved to the `<Project directory>\IdxClasses\` folder when the index family is saved.

The **Save All** option saves all script and index family files open in the **Index Family Editor**. The **Save All** option is available from the **File** menu and **Index Family Editor** toolbar whenever an opened index family or script has unsaved changes. If there are no unsaved changes in opened scripts or index family files, the **Save All** option is unavailable.

Note: If a field name is modified in the **Index Family Editor**, then it is also modified in the index family script.

Example 4-2: Modifying Field Names

```
Private Sub IndexingFamily_AfterDocumentRecognition() MsgBox (Field1.Name) End Sub
```

If the user replaces “Field1” by “ToDo”, then the script is automatically modified as follows:

```
Private Sub IndexingFamily_AfterDocumentRecognition() MsgBox (ToDo.Name) End Sub
```

Events associated to this field are not modified.

For additional help with scripting, and for examples of syntax to perform common functions, see the *Writing good VBA code* section of the *Scripting Guide*.

Example 4-3: Comparing Field Values

Many projects require a check to determine whether one field value is greater than another field value. This control must be performed during recognition to route incorrect documents to validation. The same control must be applied in validation when an operator updates the value of one of the fields.

This example provides the code to check that a field “A” is greater than a field “B”. Two event types are used:

- The `IndexingFamily_ControlDocument` is the event used to apply this control in recognition.
- `A_ValidateValue` and `B_ValidateValue` are the events used to apply this control when the operator modifies the value of field “A” or field “B”.

.

The example also uses an included file named `GreaterThan.bas`, to make the `CheckGreaterThan` routine reusable by other index families. The inclusion path uses an “*”, indicating that the `GreaterThan.bas` file is saved in the project … `\Resources\Scripts\` folder. More information about the syntax for file inclusions is available in the topic “Using Script Inclusions” on page 230.

```
  1 '#uses "*GreaterThan.bas"
  2
  3 ' ********************************************************************************
  4 ' Event called when an operator modifies the value of field A in validation.
  5 ' ********************************************************************************
  6 Private Sub A_ValidateValue(ByVal CurrentRow As Long, ByVal Validate As Boolean,
ByVal PreviousValue As String, NextField As DispatcherObjectModel.DpDocField,
CheckFieldFormat As Boolean)
  7   CheckGreaterThan(CurrentField,CurrentDocument.Fields("B"))
  8 End Sub
  9
 10 ' ********************************************************************************
 11 ' Event called when an operator modifies the value of the field B in validation.
 12 ' ********************************************************************************
 13 Private Sub B_ValidateValue(ByVal CurrentRow As Long, ByVal Validate As Boolean,
ByVal PreviousValue As String, NextField As DispatcherObjectModel.DpDocField,
CheckFieldFormat As Boolean)
 14   CheckGreaterThan(CurrentDocument.Fields("A"),CurrentField)
 15 End Sub
 16
 17 ' ********************************************************************************
 18 ' Event called at the end of the recognition of the document.
 19 ' ********************************************************************************
```

```
 20 Private Sub IndexingFamily_ControlDocument()
 21   CheckGreaterThan(CurrentDocument.Fields("A"),CurrentDocument.Fields("B"))
 22 End Sub
 23
 24
 25 The GreaterThan.bas code is:
 26 ' ********************************************************************************
 27 ' Check that first field in argument is greater than second field.
 28 ' When the test is not ok, the status of the fields are set in error and an error
message is provided
 29 ' ********************************************************************************
 30 Public Sub CheckGreaterThan(Field1 As DpDocField, Field2 As DpDocField)
 31   ' Used to initialized the error message
 32   Dim message As String
 33
 34   If Val(Field1.Value)>Val(Field2.Value) Then
 35     ' Set field status to OK for both fields. The field does not require validation
from an operator
 36     Field1.SetStatusOK
 37     Field2.SetStatusOK
 38     message=""
 39   Else
 40     ' Set field status to error for both fields. The fields require validation from
an operator
 41     Field1.SetStatusError
 42     Field2.SetStatusError
 43
 44     ' Set the error message that is displayed in validation when the field takes the
focus.
 45     ' The message explains the error to the operator
 46     message=Field1.Field.Name+" must be greater than "+Field2.Field.Name
 47   End If
 48   Field1.OutputMessage=message
 49   Field2.OutputMessage=message
 50 End Sub
```

4.12.10.2 Using Script Inclusions

A best practice for complex scripts is to use the index family script as the master script and then create separate script files and reference them as inclusions in the family script. Script files called from an include statement can contain both objects and functions. Maintaining scripts in separate files can be useful. If multiple objects or families use the script frequently, maintaining it in a separate file can greatly facilitate reuse. If a script is lengthy or complex, it can be split into several smaller files to enable debugging and troubleshooting.

To create a script file for inclusion in an index family script:

1. From the **Index Family Editor**, select **File** > **New** > **VB Script** and choose from the available options. See the *WinWrap Basic Language* section of the *WinWrap Editor Help* file for more information on each script type.

   - **Macro**: See *WWB-NET Macro commands* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*.
   - **Code Module**: See *WWB-NET Code Module* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*.
   - **Object Module**: See *WWB-NET Object Module* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*.

- **Class Module**: See *WWB-NET Class Module* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*.

2. Type the script in the script editing window. As objects and functions are added, they display in the **Proc** listbox. Index family events must remain in the `<IndexFamily>.bas` file. An included script can only contain objects and functions.

3. Save the script to a location that is available during runtime. See step 5 for more information on where inclusion files can be saved.

   - Unlike scripts created automatically with an index family, scripts that are created separate from an index family, such as a script used in an include statement, must be saved to a directory that is available at runtime. The syntax of the include statements used in the family script define the location where the inclusion script file is saved.

     This functionality differs from the default behavior for project scripts. Whereas family scripts are created in the project `Project directory\IdxClasses` folder where the index family resides, project scripts are saved to the `Project directory\Resources\Scripts\` folder.

4. Add the appropriate include statements to the family script. Enter the include statements at the top of the script, before any event subroutines, ensuring that the syntax of the statement is correct based on the location where the inclusion script is saved.

   Example 4-4: Syntax for Script Inclusion

   A script called `CommonFunctions.bas` is created and used by adding an include statement to an index family script.

   - If the script `CommonFunctions.bas` is saved in the same folder as the index family and index family script, the include statement must be typed as `"#Uses "CommonFunctions.bas"`
   - If the script `CommonFunctions.bas` is saved in the `...\Resources\Scripts\` folder, the include statement must be typed as `"#Uses "*CommonFunctions.bas"`.

     Note: Notice that the "*" in the include statement determines where the included script can be found. If the file `CommonFunctions.bas` is saved in the index family folder, then no asterisk is required. If it is saved in the project resources folder, the asterisk is required.

   - Scripts intended for inclusion can reside in custom paths, though it is recommended you place them within the project folder, in either the `...\IdxClasses` or `...\Resources\Scripts` folder. If a custom path is required, the correct syntax is `#Uses "<Script full path>.bas"` or `#Uses "<Script relative path>.bas"`.

4.13 Setting Up Extraction

The **Recognition** tab of the **Project Options** window enables setting up the behavior of the Extraction module in production.

To define recognition options:

1. In the **Project Options** window, select the **Recognition** tab.
2. Set **Default threshold for graphic anchors** and **Default threshold for textual anchors** which become the default matching thresholds for all local and global anchors (graphic and textual, respectively) when they are placed for the first time on the template.
3. Free Form image filters can improve the accuracy of recognition by *OCR* engines:
   - **Reverse video zones**: Detects and inverts reverse video text boxes in the image so they can be read by OCR engines.
   - **Matrix font**: Automatically detects the presence of matrix characters on the image and makes them bold so they are better read by OCR engines.
   - **Table lines**: Deletes all horizontal and vertical lines in the image and is useful when table characters might overlap or touch table borders.
   - **Shaded areas**: Removes the shaded background from shaded areas without altering the other areas of the image. This filter is not recommended when shaded areas contain thin or very thin characters as these characters may be eroded by the filter so poorly read by the OCR engines.
   - **Text box reading**: Segments the whole page into individual text lines that are then passed individually to the OCR engine. This filter is recommended for OCR engines whose line detection is not very good. Instead of calling the OCR engine once in the full page mode, the OCR engine is called as many times as there are text lines in the image so this can have license impact for some engines.
   - **Recognition quality**: Can be set for accuracy or speed selecting **Accurate** or **Fast** options respectively.
4. Specify the **OCR engine configuration** options to be used during recognition.

OCR engine for table Free Form fields and textual classification	Specify the *full-text*OCR engine and confidence threshold to use in free-from table recognition and textual classification.
OCR engine for anchor text values	Specify the OCR engine and the confidence threshold to apply to anchor text values. This setting can be used to change to an OCR engine more suitable for non-Latin character sets.

Default OCR engine for rubberband	Specify the OCR engine and confidence threshold defaults to apply to rubberbanding.

5. If you want to use the OCR data cache instead of running full OCR on PDF and PDF/A documents and image files, then select the appropriate options on the **Standard OCR** tab.

Notes

- In Recognition Designer, PDF and PDF/A pages are displayed and processed as images. The resolution of these images is determined by the project resolution. If the project resolution is not defined or the project contains only generic templates, then the resolution for images is 300 DPI.

Property	Description
Enable OCR data cache from Standard OCR	When this option is selected, the following behavior for PDF and PDF/A documents (converted from Microsoft Office documents and original PDF and PDF/A documents only) and images is enabled (except for the OCR engines specified in **Select OCR engines to use instead of Standard OCR cache**): • Classification When performing textual, keyword, and text matching classification on PDF pages and images, the text (including coordinates and line/word separation) in the OCR data cache is used. In addition, PDF pages are not converted to images, which could result in better performance. **Note: Project Options** > **Recognition** > **Free Form image filters**, **Project Options** > **Text matching image filters**, and **Indexing** > **Index View** > **Image Clean Up** filters are skipped. • Identification and Completion For a PDF page, rubberbanding uses the OCR data cache. **Notes** – Annotations are disabled. – The PDF page cannot be rotated. • Extraction When performing extraction on PDF pages and images, zone and free-form recognition uses the OCR data cache. In addition, if PDF pages are not converted to images, then better performance could result. If the following elements do not exist on the page, then the PDF page is not converted to an image: – Table fields – Graphical anchors However, if the aforementioned elements do exist on the page, you could ignore them as follows:

Property	Description
	– To ignore graphical anchors, enable the **Disable graphical anchors for zonal fields** option. – To ignore table fields, disable assignment of table fields. **Note: Project Options** > **Recognition** > **Free Form image filters** and **Indexing** > **Index View** > **Image Clean Up** filters are skipped.
Apply text-based classification before graphical classification	With the OCR data cache, text-based classification (Textual, Keyword, and Text-Matching) could be faster than graphical classification. If the aforementioned is true, then select this option to improve performance.
Disable graphical anchors for zonal fields	Select this option to disable graphical search for zonal index or table field anchors, which prevents the loading and rendering of these input pages. Thus, enabling this option could result in improved performance. In addition, if a text value is assigned to the anchor, then the anchor is determined by the OCR data cache.
Select OCR engines to use instead of Standard OCR cache	Specify the OCR engine to use in place of the OCR data cache. The following images on PDF pages can only be recognized by OCR: • Barcode images • US and/or French check images • Checkboxes In addition, if PDF pages are composed of images only and free-form rules are optimized for a particular OCR engine, then the recognition accuracy of the OCR data cache would be reduced.

6. Click **OK** to save changes to the **Recognition** options.

Chapter 5

Windows

5.1 Advanced Learning Wizard

Run the Advanced Learning Wizard to create templates based on templates codes. Select **File** > **Advanced Learning** to start the wizard.

Table 5-1: Advanced Learning Wizard Components

Element	Description
Start	The left pane displays the steps for creating a template from annotated images. Each step is preceded by a color square. When the colored square is dimmed it means that a step has been successfully completed. The right pane displays available configuration options and enables navigation and buttons for canceling or closing the wizard.
Annotated Base Selection	Images are classified by code, not by name. **Awaited format** displays the expected folder format for all the images in a code group.
Automatic Learning	Enables automatic template creation. During the automatic learning process, information on the templates is displayed, along with information on the number of images processed and the number of templates created.
Generated Templates	Displays the number of templates created. Templates are listed by template code. The **Total images** column indicates the number of images used to create the template. The image of the selected template is displayed in the right pane of the window.
End	Displays the results of the automatic learning process. It also displays the number of created templates. As soon as you close the wizard, the new templates are displayed.
Advanced Learning Wizard toolbar	Provides tools that assist and complement use of the Advanced Learning Wizard.

Element	Description
Cancel button	Closes the Advanced Learning Wizard window without adding the templates to the project.
Previous button	Click to return to the previous step.
Learn button	Runs the automatic advanced learning process. If the selected images are not homogeneous then a warning message appears asking you if you want to select a unique resolution. If you click **Yes**, a window appears to define a common resolution for all the images in the base.

5.2 Table Wizard

The **Table Wizard** runs in the Completion module, where it helps operators to place table fields. In the **Index View**, select **Template Test**, and then **Tools** > **Table Wizard**.

Table 5-2: Table Wizard

Element	Description
Image pane	Displays the reference image where table fields are automatically placed.
Visible Table Fields pane	Displays a **Located** column which indicates fields placed automatically. Fields without a tick must be placed either using potential placements or manually.
Table pane	When running *OCR* reading, data lines display at the bottom of the window. Click  to show/hide the table pane. Whenever you move or delete a field the OCR data becomes obsolete and data lines are dimmed.

Element	Description
Table Wizard toolbar	Enables you to: • Start automatic placements. • Apply fields placed in the document previous to the current document. • Adjust the height of all table fields. • Delete current or all placed table fields. • Start the reading of table line data. • Show/hide possible placements or extracted table line data. • Apply an anti-aliasing filter to the whole image. • Apply a dilation filter to the whole image.

5.3 Main Window

Most application design is done from the main window. The development environment features an automatic learning module that builds templates based on sample documents. It also has a complete range of modules used to create, edit, and test the project before exporting it to the production environment. Capabilities can be further enhanced and customized by implementing *VBA* scripting, which enable data lookup and management capabilities.

Table 5-3: Recognition Designer Main Window

Element	Description
Left pane	The left pane is divided into three areas: the project panel, the template panel, and the template properties panel. • The project panel displays the root directory, represented by an icon, and followed by the project template name. Add or delete subdirectories from the root directory by right-clicking on the project template name and selecting **New directory** or **Delete directory**. • The template panel displays a table with the list of the templates associated with the project. The table assumes the following columns: **ID**, **Separator**, **Name**, **Code**, and **Family**. Templates can have different colors depending on their type: Standard, *HPA*, Text Matching, or passive templates. For more information on the template panel, see Classification View or the Indexing View respectively. • The **Template Properties** panel displays information about the selected template, like the **Name**, **Code**, **Index family**, and **Separator type** associated with the template. To hide or display the panel, select a template from the templates list, right-click, and select **Template Properties** to toggle the panel on or off.
Right pane	The right pane is divided, displaying the image pane, the **Image base**, indexing fields, and field properties. The elements displayed change based on the selected view: The **Classification View** or the **Indexing View**. • The image pane displays the image corresponding to the selected template. The image pane displays in both the **Classification View** and **Indexing View**. • The **Image base** displays a thumbnail of the images belonging to the template image base. The **Image base** displays in the **Classification View**. • Indexing fields and field properties are displayed in the **Indexing View**.

Element	Description
File menu	• **New**: Creates a project (recognition project). This command is available in Dispatcher Manager only. • **Open**: Opens an existing project. This command is available in Dispatcher Manager only. • **Re-open**: Reloads a project tree structure among the previously opened projects tree structure. This command is available in Dispatcher Manager only. • **Close**: Closes the current project. This command is available in Dispatcher Manager only. • **Save** : Saves the current modifications. This command is available in Dispatcher Manager only. • **Save As**: Enables saving the current project in any location. This command is available in Dispatcher Manager only. • **Send To Production**: Prepares a copy of the project with all the elements needed for running the project in production. Project elements used for development, like the image base and unclassified images gathered by Collector, are not sent to production. • **Export the Project**: Exports the project as a zip file that can be exported either for development or production. This command is available in Dispatcher Manager only and allows to compile and save the project before exporting the project. – **Compress the Whole Project**: Exports the development project. All the project parameter files are exported together with the template image bases, images gathered by Collector, and the content of the compiling cache directory. – **Compress for Production**: Exports the project for production. The project parameter files are exported without the image bases or the compiling cache directory. • **Import Templates**: Imports templates using the **Template Import Wizard**. • **Edit Project Script**: Opens a VBA editor for editing the project script. This

Element	Description
	command is available in Dispatcher Manager only. • **Project Options**: Opens the **Project Options** window. For more information on setting the options, see "Project Options" on page 276. • **Advanced Information**: Opens the **Advanced Information** window displaying information about the project like the name, number of templates size, directory, and template list. From this window, print or export a list of the project templates. • **Update Project**: Opens the Project Update Wizard for creating new templates based on new and unrecognized document images. • **Advanced Learning**: Opens the Advanced Learning Wizard for creating new templates from an annotated image base. • **Connected Users**: Opens the **Users connected to current project** window displaying information about users currently connected to the project, including the **User**, **Machine Name**, and **Connection date**. • **Exit**: Quits.

Element	Description
Classification menu	• **Classification View**: Displays classification options, information, and tools. • **Keyword Classification**: Opens the **Edit Keyword Classification** window for defining text-based classification based on keywords. • **Text Matching Designer**: Opens the **Text Matching Designer** window for creating classification rules relating to text matching templates and references. • **Image Base Viewer**: Displays or clears the template image base pane at the bottom of **Recognition Designer** window when in **Classification View**. • **Search**: Opens the **Search a Template** window for searching templates by ID, name, code, or associated index family. • **Compile**: Runs the project compilation, including automatically preprocessing the project parameters. The project must be compiled before sending it to production and whenever a significant change is made. For example, creating a project, adding templates by any method, deleting templates, merging templates, or changing a template code. The compiling time depends mostly on the number of standard templates. Other template types compile more quickly. The compiling time is optimized using a cache that is enabled by default. When compiling a project for the first time, one compiled file per template is generated. When the project is updated, new templates and changed templates are saved to the project `\cache` folder.  **Caution** Do not disable the compiling cache or empty the cache directory. Instead, upgrade the existing machine or move the project to a machine with more resources. Compiled files in the cache directory are exported when saving the project to a new directory and when exporting the project.

Element	Description
Indexing menu	• **Index View**: Displays the **Indexing View**. Fields, anchors, and field properties are displayed on the properties panel. • **Template Wizard**: Opens the **Template Wizard** to create a standard template. This wizard creates one template at a time. • **Copy the Indexation Parameters**: Copies all index family settings including fields, field properties, and field placements from the selected field to the Clipboard. Use this feature to copy settings from one field to another. • **Paste the Indexation Parameters**: Pastes index family settings including fields, field properties, and field placements from the Clipboard to the selected field. • **Index Family Editor**: Opens the **Index Family Editor** for creating and modifying index family properties.
Test menu	• **Classification Test**: Runs a classification test. For more information on checking the classification performance, see "Testing Classification" on page 104. • **Unit Test**: Runs a unit-test and opens a window appropriate for the selected field: Anchor, table, or index field. Unit tests are available for anchor fields, table fields, or index fields. This option is dimmed until a field is selected in the **Indexing View**. See topics on the Anchor Unit-Test, Field Unit-Test, or Table Field Unit Test for more detailed information on **Unit Tests**. You can also use PDF files as a test base, which also loads their associated OCR data caches. • **Template Test**: Opens the **Template Test** window and runs a template test based on the project recognition settings. The Template Test is available from the **Indexing View**.

Element	Description
Tools menu	• **Project Analyzer**: Opens the **Project Analyzer** and checks the integrity of project templates. • **Image Analyzer**: Opens the **Image Analyzer** for checking image resolution. • **Free Form Designer**: Opens **Free Form Designer** for defining free form data extraction rules. • **OCR/ICR Engine** – **New**: Creates an engine configuration file and displays the engine window with parameters specific to the selected engine. – **Edit**: Opens the **Select Resources** window to edit the settings of an engine configuration file To customize a standard engine configuration file, copy and modify the file as described in the topic "Adding Engine Configuration Files to the Project" on page 109
Help menu	• **Help**: Opens the documentation. This guide provides detailed instructions on understanding, configuring, designing, and using this application. • **About**: Displays the version number and copyright.

Element	Description
Toolbar	• **New**: Enables you to create a recognition project. • **Open**: Opens an existing project. • **Save**: Saves modifications to the project file. • **Options**: Opens the **Project Options** window for setting and modifying project options. • **Send to production**: Sends the project to production, including preparing a copy of the project containing the required elements for running the project in production. Development components, such as the image base, are not sent to production. • **Classification View**: Displays classification view. • **Indexing View**: Displays indexing view, including fields and field properties. • **Index Family Editor**: Opens the **Index Family Editor** for creating and modifying index families and index family settings. • **Zoom in**, **Zoom out** and **Default zoom**: Respectively enable zooming capabilities and restoring the initial display size of the image. • **Image base**: Display or hide the image base thumbnails in the **Classification View**. • **Search a template**: Opens the Search a Template window to search for templates by criteria such as name, code, ID, or index family. • **Template properties** Display or hide the **Template properties** panel that displays in the left pane. • **Classification Test**: Opens a **Classification Test** window. • **Unit Test**: Runs a unit-test and opens a window appropriate for the selected field such as anchor, table, or index field. See topics on the Anchor Unit-Test, Field Unit-Test, or Table Field Unit Test for more detailed information on **Unit Tests**. • **Template Test**: Opens the **Template Test** window and runs a template test based on the project recognition settings. The

Element	Description
	Template Test is available from the **Indexing View**.

5.3.1 Classification Test Window

The **Classification Test** window provides classification performance, shows classified and unclassified images, checks pre-classification and decision rates, and exports test results as well as classified and unclassified images.

Table 5-4: <Project Name> Window

Element	Description
File menu	Presents options for opening the tree structure, project images, and the project image base. Also enables exporting images and results, and closing the window. • **Open**: Select from the following: – **Tree Structure**: Loads directories containing the image base. – **Tree Structure in duplex mode**: Loads project images in duplex mode with the name and path of both sides of a duplex document, though classification is done on the front page only. – **Images**: Loads one image at a time. – **Project Images**: Reloads all the project images. • **Re-Open**: Reloads an image tree structure among the previously opened image tree structure. • **Export the Following Images**: Displays the **Image Export** window for selecting images to export. • **Export Results**: Exports the test results including pre-classification rates and decision rates. • **Close**: Closes the **[Project Name]** window.
Edit menu	Enables you to delete images or select all the images from the **File** list. • **Delete**: Removes the selected document from the **File** list. • **Select All**: Selects all the documents in the **File** list.

Element	Description
Display menu	Displays different information before and after testing classification. **Display** menu before testing classification: • **Advanced Information**: Displays the pre-classification and decision rates columns in the left pane. Pre-classification and decision rates depend on the template type. See "Understanding Pre-Classification and Decision Rates" on page 68. • **Template Viewer**: Displays the template reference image in a separate window. • **Zoom In**: Zooms in the document selected in the **File** list. • **Zoom Out**: Zooms out the document selected in the **File** list. • **Default Zoom**: Restores the initial display of the image. After running the classification test, the **Display** menu changes and includes the following additional options: • **Single List**: Displays image per image without specific classification. • **Per Code**: Displays the images per template code. • **Per Template**: Displays the images per template name. • **Per Code and then Per Template**: Displays the images per template and grouped per template code.
Test menu	Runs the classification test on all templates in the project or simply on specific types of template: • **Run**: Runs the classification test. • **HPA Classification** • **Standard Classification** • **Handwritten Classification** • **Keyword Classification** • **Text Matching Classification**

Element	Description
Left pane	Displays the **File** column that lists all the template images. After running the classification test, the left pane displays the following tabs: • **Classified** tab • **To Confirm** tab • **Not Classified** tab
Right pane	Displays the reference image selected in the **File** list.

5.3.2 Classification View

Table 5-5: Classification View

Element	Description
Project panel	Displays the root directory followed by the project template name. You can add or delete subdirectory from the root directory by right-clicking on the project template name. Two options open from the root directory: **New Directory** or **Delete Directory**.

Element	Description
Template panel	Displays a table with the list of the templates associated to the project. Templates can have different colors depending on their type: standard, *HPA*, textual, graphical, textual and graphical, or generic templates. When a file has a textual template associated with it, additional *OCR* properties appear in the lower-right corner of the window. Use the **Anchor text value** field to type required value. When you right-click the template list, a menu appears with several options: • **New Template**: Opens the **New Template Wizard** window. This wizard enables you to create one standard template at a time. • **New Generic Template**: Enables creation of a generic template. A generic template has a reference image but no associated image base so that it is not used to run classification. Create a generic template to use classification with keywords rules or to set a default template. • **New HPA Template**: Select a reference image and then the **HPA Template Editor** opens. This option is only available if there is an HPA template in the project. • **Convert to HPA Template**: Converts a standard template to an HPA template. • **Merge Templates**: You can merge standard templates but you cannot merge HPA templates or generic templates. Templates are merged when numerous templates are created from a base of images for which only one template is needed. • **Delete Templates**: Delete the selected templates. • **Template Properties**: Rename templates, give them templates codes and assign them index families. Displays the **Template Properties** at the bottom of the window. • **Rotate the Template**: Rotate templates to display them in a natural reading orientation on the screen. Automatic learning creates templates with the same orientation as the orientation of images at

Element	Description
	scan time. To rotate for easier reading, select from three rotation options: **Left**, **Right** or **180**. • **Edit HPA template**: Only available if you select an HPA template. It opens the **HPA Template Editor** window. • **Edit Text Matching template**: Only available if you select a text matching template from the template list. It opens the **Text Matching Designer** window. **Note:** An alternative to merging two templates is to assign them the same template code.
Template properties panel	**Template properties**: Rename templates, give them templates codes and assign them index families.
Image pane	Displays the image corresponding to the selected template.
Image base panel	Displays a thumbnail of the images belonging to the template image base. The number of thumbnails is defined in the option **Maximum number of images linked to an image reference** when creating the project with the New Project Wizard. Only standard templates have many thumbnails. HPA and generic templates have only one thumbnail that corresponds to the template reference image. There is only one reference image for an HPA template or a generic template.

5.3.3 Index View

Table 5-6: Index View Window

Element	Description
Project panel	Displays the icon followed by the project template name. Add (**New directory**) or delete (**Delete directory**) subdirectories from the root directory by right-clicking on the project template name.

Element	Description
Template panel	Lists the templates associated to the project. Templates are color-coded by type. Templates can have different colors depending on their type: standard, *HPA*, text matching, or generic templates.
Template properties panel	Enable to rename templates, give them templates codes and assign them index families.
Image pane	Displays the image corresponding to the selected template with any existing fields.

Element	Description
Fields and anchors panel	Displays index fields, table fields and global anchors. Table fields have a dashed border to differentiate them from index fields, which have a solid border. When an anchor is selected and placed on a template, the following **Anchor** properties display: • **Search Zone**: Defines how far from the center of the anchor to look for the object targeted by the anchor. The size of the anchor itself defines the extent of what is searched for. When an image is evaluated during production, it can be offset by any distance within the defined search zone. Define both **H** (vertical offset) and **W** (horizontal offset) in millimeters (mm). • **Graphic matching threshold**: The graphic matching threshold is set by moving the slider at the right of the sample anchor image. The threshold is how closely the document image matches the anchor image. If scanning results in minor variations in the quality of the image, an exact match is not required when setting thresholds below 100%. The default of 50% is useful in most situations. • **Text matching threshold**: The text matching threshold is set by moving the slider at the right of the sample anchor image. The threshold is how closely the document image matches the anchor image. If scanning results in minor variations in the quality of the image, an exact match is not required when setting thresholds below 100%. The default of 80% is useful in most situations. • **Anchor text value**: a field for entering an anchor text. • **Conditional anchor** and **Field substitution value** (index fields only): Select **Conditional anchor** for conditions when an index anchor is not found. In this case, the **Field substitution value** is in the place of the index field information. This property is not available for table field anchors.

Element	Description
Default Value	Defines a default value for unrecognized fields, and for those fields with no recognition engine selected.
Recognition tab	Select an engine configuration file and assign a confidence threshold. • **OCR engine** field contains the default engine and the confidence threshold value assigned to the field in the **Index Family Editor**. If no engine is defined in the index family, assign an engine configuration file to the field in the **Index View**. • **Field layout** Select the reading orientation for the current field: **Normal** if reading from left to right, **90 top** if reading from bottom to top and **90 bottom** if reading from top to bottom.
Image Clean Up tab	Manage image filters to improve the quality of recognition. • Filter list displays the selected filters. •  Changes the order of filters in the list, moving a filter upwards. •  Changes the order of filters in the list, moving a filter downwards. • **Add** opens the **Select Resources** window and enables addition of filters to the list. • **Delete** clears filters from the list.
Variables tab	Variables are created for scripting use only. In the index families, for each field, define the variables that are available from the script. Variables can be used in some index family events. This tab displays all the variables defined in the index family. These variables are scripting variables sent to or returned from the script. They enable you to retrieve a defined value in any event of the project.

Element	Description
Keying tab	Enables the **Invisible during validation** option. **Always confirm** appears dimmed and cannot be modified. • **Always confirm** is only available if this option is not selected for an index field in the index family. When selected, the operator must confirm the field value when performing a template test, even if the field has been correctly recognized. This option is dimmed if it has already been selected in the **Index Family Editor**. • **Invisible during validation** results in fields not appearing when performing a template test. Also, pre-indexed fields on which this parameter is set are not displayed in Identification. • **Custom fixed format** controls the type of data validated for the field. They can be assigned to index fields only and are usually defined when the field is created in the index family. A **Custom fixed format** can also be defined when the field is placed on the template. For help customizing fixed formats and working with partial values, see "Understanding Text Field Fixed Formats" on page 219. • **Partial value accepted** enables partial values to be accepted for fixed format fields. A format must be typed in the **Custom fixed format** text box before the **Partial value accepted** checkbox is enabled.

Element	Description
FreeForm tab	Associate definition (*DFT*) files containing free form settings with fields and to import *OCR* data. **Import OCR data from field** is only available for index fields. • **Free Form settings** populates the free form settings field with the name of the DFT file created in Free Form Designer. • **Import OCR data from field** is only available for index fields. Select the field whose recognition data you want to use for the current field. You must not use pre-index fields for importing OCR data. Pre-index fields are populated at the Classification step and not during the Recognition step. • **Selection of Free Form Parameters** window is displayed by selecting a field and then selecting the **Free Form** tab and clicking  . – **File** pane displays a list of definition files. Select the definition file that contains the free form settings you have defined for the current field. Files will be dimmed because they do not contain an index field or a table field with the same name as the current field. If no DFT file is available, make sure that you saved it to the correct location *<Project directory>*`\Resources\OCR\` – **OK** button validates the selection and exits the **Selection of Free Form parameters** window. – **Cancel** button cancels any operation and exits the **Selection of Free Form parameters** window.

Element	Description
AutoFormat	• **Fuzzy Regular Expression** For more information, see "Fuzzy Regular Expressions" on page 100. • **Output Format** A regular expression that specifies the format of the output to the operator. You can also reorder the results using groups; the first group is identified with the number 1. Unrecognized characters (identified by question marks) are included in the appropriate group. You can use this option to replace existing scripting that produces the same output. For example, you could reorder a date, `MM/DD/YYYY` to `DD/MM/YYYY`. – If the following fuzzy regular expression is defined: ○ **Value**: `(\d{1,2})/(\d{1,2})/(\d{2,4})` ○ Sample input: `01/31/2015` – Then, the output value format is as follows: ○ **Output Format**: `{2}/{1}/{3}` ○ Sample output: `31/01/2015` • **Automatically fix value** Select this box to enable OCR engines to automatically substitute low-confidence scanned characters with an appropriate alternate character. However, if an appropriate alternate is not available, then a question mark is substituted. For example: – If a fuzzy regular expression specifies digits only for a zip code (for example, `\d{5}`) and the following conditions are also true: ○ A zero is recognized as the letter `O`. ○ A zero exists as an alternate character. Then, a zero is substituted. **Note:** The alternate characters available for substitution are provided by each particular OCR engine; that is, the available

Element	Description
	alternate characters can vary between OCR engines. • **Hit threshold (%)** Enables a single fuzzy regular expression to have a wider range of text matches.
Borders tab	Only available for table fields. Enables you to detect the right/left, top/bottom borders detection and to remove horizontal lines. • **Right/Left border detection** detects the right and left edges of the table field because borders can prevent correct line detection. • **Top/Bottom border detection** detects the top and bottom edges of the table field because borders can prevent correct line detection. • **Line Removal** filters out horizontal lines because horizontal lines can prevent correct line detection. This option is not available in the index family.
Lines tab	Only available for table fields. Enables detection of only those lines with values, or all the lines, including those without values. • **Detect all lines** includes all lines when selected. When this option is cleared, only lines having values are detected. • **Height of lines** enables setting the minimum and maximum line height required for the detected line. By default, **Min** is set to 15 and **Max** is set to 50. Any line with a height outside the specified range is not detected.

5.3.3.1 Template Test

Template Test tests the project recognition settings including the project and family scripts. It simulates the Completion module interface. This is only available when in **Index View**.

Table 5-7: Index View Template Test tab

Element	Description
Test Base menu	Loads images, reloads images from templates, reloads images from a template test base, and closes the **Template Test** window.

Element	Description
Zoom menu	Zooms in or zooms out the reference image.
Tools menu	Opens the Table Wizard or refreshes the resources. For information on how to configure and use the Table Wizard, see "Table Wizard" on page 238.
Launch test button	Runs the template test.
Test pane	Displays a table with all the tested images. It assumes the following parameters: **Image**, **OK?** and the recognition time.
Image pane	Displays the reference image used for the test. Images used for the test are loaded automatically. These are images from the template base and the template reference image (`classifier.tif`). To load other images, select **Test Base** > **Load Images**. To reload images from templates, select **Test Base** > **Load the Template Base**.
Table pane	Displays the table fields results on the reference image. This test depends on the options applied in the **Field properties** of the **List of Table Fields** tab in the **Index Family Editor**. For each table field, a message appears in the message area. If the value is correct a valid message appears: **Indexing is finished**. If it is not, an explanation appears in the **Message** area, such as "Format not respected".
Index pane	Displays the index fields results on the reference image. This test depends on the options applied in the **Field properties** area of the **List of Table Fields** tab in the **Index Family Editor**. For each index field, if the value recognized is correct a valid message appears: **Indexing is finished**. If it is not, an explanation appears in the **Message** area, such as "Format not respected".

5.3.4 Index Family Editor

The **Index Family Editor** window enables configuration of recognition. The **Index Family Editor** window can display two main views: The "Index Family Editor" on page 260 view, and the "Index Family Editor Script Definition" on page 272 view. The two views enable creation and definition of all aspects of an index family including the data to be extracted from a specific type of document and scripts that can be customized to perform a variety of tasks. More information about index families is provided from "Index Family Editor" on page 260.

5.3.4.1 Index Family Editor

The index and table field definition configuration of the **Index Family Editor** window provides definition of fields that define the data that is extracted during production. The **Field Properties** panel specifically enables setting field parameters.

Table 5-8: Index Family Editor: Index and Table Field Definition View

Element	Description
File menu	The **File** menu enables these options: • **Save** saves the current index family. Both files XINDEX and INDEX are saved in the current project folder: `<Project directory>\IdxClasses\` • **Save All** saves all opened index families. • **Close** closes the selected index family and leaves all other open index families and the **Index Family Editor** open. • **Close All** closes all open index families and script editors and leaves the **Index Family Editor** open. • **Exit** exits the **Index Family Editor** window.
Edit menu	All menu commands are dimmed.
View menu	Enables display of the **Field Properties** and the **Index Family Explorer**. These can be undocked.
Help menu	Displays a link to open the *Recognition Designer Guide*.
Index Family Editor toolbar	The toolbar provides common operations also available in the menus. The toolbar options available differ based on what is displayed in the main window. Tooltips describing the available operations are displayed when the mouse hovers over the tool.

Element	Description
Main window	Displays the tabs for open index families and scripts. • Selecting an index family tab displays the list of **Index Fields** or **Table Fields** depending on the selection from the **Field Type** list box. The columns represent the parameters selected for each index field. • Selecting a script tab displays the script editor with the script content displayed for editing.
Field Properties, and **Index Family Explorer** panels	Displays the "Field Properties Panel" on page 262, or the "Index Family Explorer Panel" on page 261 • Enable display of these panels from the **View** menu. • Undock the panels by dragging the title bar into the center of the window and reposition as necessary. Re-dock the panel by double-clicking the title bar. • Auto-hide the panel by clicking the **Auto Hide** button at the top of the panel. Tabs display on the side of the **Index Family Editor** when the panels are hidden.

5.3.4.1.1 Index Family Explorer Panel

The **Index Family Explorer** provides a navigation tree for the list of index families and scripts found in the `IdxClasses` folder of the current opened project. This provides quick access for management of indexes and scripts. A single project can contain one or more index families. Click to expand the branches and select a branch to open the selected item in the details pane.

The **Index Family Explorer** displays one node for each index family. The following actions are available:

- Double-click an index family to open the index editor.
- Double-click a script to open the script editor.

5.3.4.2 Field Properties Panel

Set the field properties for each of the defined index fields or table fields using the **Field Properties** panel. Select an index field or a table field to display the properties for that field. Click in the panel to modify a setting. Parameters can be displayed by category or can be alphabetized. Three buttons alter the display of field properties:

- **Categorized**: Properties are displayed according to the categories described in this table.
- **Alphabetical**: Properties are displayed alphabetically, independent of their categories.
- **Property Pages**: Option not available.

Table 5-9: Field Properties panel grouped by category

Group	Property	Description
General group:		
	Index	Not editable. The order in which fields are processed in during extraction and tabbed by the Completion operator during validation.
	Name	Not editable. The field's name in the document type/index family. Is used to reference a field, including in scripting. Is not displayed to the Completion operator.
	Size	The maximum number of characters that can be entered. In production, if a value exceeds the field size, the value is stored but the field is set in error. This option is available only for **Text** type fields if the **Fixed format** option is cleared. For the **Date** or **Amount** field types, the field size is computed automatically.
Recognition group: set these settings to configure the field's OCR engine and visibility in the **Index View** panel		

Group	Property	Description
	Invisible in template editor	Indicates whether the field is visible in the **Index View** panel of Recognition Designer. Values: **True** (hidden) | **False** (visible). If hidden, the field still displays when performing a template test.
	OCR engine	Optional. Specifies the OCR engine for the selected index field/table field: • **Confidence**: The minimum confidence threshold that a character must obtain to be recognized. Learn more about thresholds in topic "Recognition Engine Confidence Threshold" on page 110 • **Default engine**: The name of the selected OCR engine. Click the button in the property box and select the engine from the **Select Resources** dialog box as described in topic "Assigning an Engine Configuration File to a Placed Field" on page 110. **Note:** You can specify the OCR engine settings for each field later when placing fields on a template.
Control group:		
	Always confirm	Specifies whether the Completion operator must confirm the field's value manually. Values: **True** (always confirm) | **False**. If **True**, the field cannot be considered validated until the operator has confirmed it.

Group	Property	Description
	Required field	Specifies whether the field must contain data or can be validated if blank. Values: **True** (data required) | **False**.
	Type	Displays for index fields only. Specifies the data type of the field. Values: **Text** | **Amount** | **Date**.

Group	Property	Description
	Type settings	Displays for index fields only. Specifies additional settings for the selected **Type** property: • **Type** property set to **Text**: – **Fixed format**: Optionally, enter the format string to restrict the value to the specified format. Learn more about formatting in section "Regular Expressions" on page 96. – **Partial value accepted**: Specifies whether a value partially matching the format string can be accepted. Values: **True** | **False**. • **Type** property set to **Amount**: – **Decimals**: Specifies the number of decimals after the decimal separator. – **Maximum**: Specifies the maximum allowed value. – **Minimum**: Specifies the minimum allowed value. – **Separator**: Specifies the character to be used as a decimal separator. Values: "," (comma) | "." (period). – **Units**: Specifies the number of digits before the decimal separator. • **Type** property set to **Date**: – **Format**: Specifies the expected date format. Select the date format

Group	Property	Description
		from the list in the property box. – **Limit type**: Specifies the way to limit the date interval. Values: **Fixed** | **Floating**. – **Limit type settings**: Specifies the date interval: ○ **Limit type** is set to **Fixed**: Specify the exact dates in the **Minimum** (start date of the interval) and **Maximum** (end date) options. Use the calendar that can be expanded in each property box. ○ **Limit type** is set to **Floating**: Specify the number of days relative to the processing date in options **Lower limit** (first day) and **Upper limit** (last day). For example, set **Lower limit** to `-5` and **Upper limit** to 5.
	Variables	Optional. Can be accessed in the scope of advanced recognition (VBA) scripting only. Defines the list of variables associated with the field to be used in scripting. Clicking the button in the property box opens the **Variables Editor** dialog box in which you can create variables.
Validation UI group: impacts the way the fields are displayed when performing a template test.		

Group	Property	Description
	Character validation	Specifies whether the operator will validate the value in this field character by character. Values: **True** (character validation enabled) | **False**.
	Color	Sets the color to the field to provide visual information about the field's meaning. Colors can be used to categorize information.
	Field Width	Displays for table fields only. Sets the field (table column) width in pixels.
	Invisible during validation	Specifies whether the field will be displayed to the Completion operator. Values: **True** (hidden) | **False**. If set to **True**, the field does not appear when performing a template test.
	Label	Specifies the field's label displayed to the Completion operator in the data entry form. During template tests, the field's label is displayed rather than the field name.
	Read only	Specifies whether the field is read-only to the Completion operator. Values: **True** (read-only) | **False**.
	Uppercase characters	Specifies whether all characters in the field's value are automatically converted to uppercase. Values: **True** (uppercase) | **False**.
Lines Extraction group: displays for table fields only		

Group	Property	Description
	Border Detection	If **True**, detects the edges for both **Right/Left** and **Top/ Bottom** sides of the table field. When templates contain borders, this enables Recognition Designer to automatically size fields to incorporate all table data. When reading the table field, if border lines are detected, the width and height are adjusted in relation to the borders.
	Detect all lines	When set to **True**, all lines including those without a value are detected. When this option is cleared, only lines having values are detected. For more information on line detection, see "Table Recognition: A Simple Example" on page 144.
	Line height	Used to set the limits for font size during recognition and improve the accuracy of detected lines. Expand **Line Height** to specify **Maximum** and **Minimum** height values in millimeters (mm). Any text height outside the range of the specified values will not be detected.

Group	Property	Description
	Paragraph settings	Paragraphs are groupings of text in a table. Paragraph headers define the beginning of a paragraph in a table, and all text below that is part of the paragraph until the next paragraph header is detected. It is possible to read and display only the first line of the paragraph (that is, the header) or the header followed by all the article lines inside the paragraph. The next paragraph begins at the next header. The available options are: • **Field Status** determines the type of table field. If **Linked to Paragraph Field** duplicates paragraph headers to empty lines, so the value of the first detected field is repeated on each line of the paragraph. With **Paragraph Field** the current table field is used to detect all paragraphs in the table fields, and activates the **Headers Only** and **Keywords** options. Only one field is used for paragraph detection. **<None>** disables **Paragraph Settings** for this field. • **Headers Only** when set to **True**, only displays the header line of a paragraph. • **Keywords** defines the keywords to identify the paragraph headers. Click the browse button to display the "Keyword Editor" on page 270 window.

5.3.4.3 Keyword Editor

This window enables definition of **Paragraph Settings** keywords. This is only activated when the **Paragraph Settings Field Status** is set to **Paragraph Field**.

Table 5-10: Keyword Collection Editor Window

Element	Description
Keywords	Click **Add** to create a keyword. The keyword is added with default parameters listed in the **Properties** pane.

Element	Description
Properties	Displays the **<***Keyword***> properties** for the selected **Keyword**. If multiple **Keywords** are selected, the parameters are filled if all selected members use the same value. Where parameters vary between members, the fields are blank. Click in the right column to change a parameter. The parameters for keyword members are: • **Name** is an internal name for listing the keyword during setup. • **Type** provides the categories to define a keyword: – **Constant**: Searches for a specified string. Specify the **Type Settings**. ○ **Case Sensitive**: If **True**, the string must match the case specified in the **Value** field. ○ **Threshold**: Set the percentage of characters that must match **Value** for the field to be accepted. If set at *<100>*, the string must be an exact match to the **Value**. If set at *<70>*, only 70 percent of the characters must match. ○ **Value**: Type the string to match. – **Date**: Searches for dates in a specific format. Specify the **Type Settings**. ○ **Format**: Click the drop down button and select the date format to search for. – **Regular expression**: Searches for strings based on a regular expression. Specify the **Type Settings**. ○ **Regular Expression**: Click the browse button to display the **Regular Expression Editor** window. Select a predefined regular expression, or select <Custom> and type the regular expression needed in the **Validation expression** field.
Enter text here to test	Expand the window to make this option available. Enables testing of constants, dates and regular expressions. Type an expression in the edit box and click **Test**. The results are displayed.

Related Topics

"Edit Keyword Classification" on page 297

"Defining Keywords" on page 177

5.3.4.4 Index Family Editor Script Definition

The **Index Family Editor** window provides options for creating scripts that control how index families behave during processing. Scripts control different aspects of behavior based on the type and content of the script.

Many options available in the **Index Family Editor** scripting window are customized from the third-party scripting product WinWrap. The *WinWrap Editor Help* and *WinWrap Language Help* files are accessible from the **Index Family Editor** script definition **Help** menu. Additional information on available and custom menu selections are discussed in this topic. For more information on scripting functionality, see the *Programming Reference Guide*.

Table 5-11: Index Family Editor Script Definition View

File menu	The **File** menu displays options for creating new index families and scripts, opening, saving, and closing index families and scripts, and exiting the **Index Family Editor**. Scripts can also be edited and printed from the **File** menu. Selecting **Open Uses** opens one tab for each module or macro declared in a `#Uses` statement at the start of the script being edited. Four script types can be created from the **File** > **New** menu. See the *WinWrap Basic Language* section of the *WinWrap Editor Help* file for more information on each script type. • **Macro**: See *WWB-COM Macro* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*. • **Code Module**: See *WWB-COM Code Module* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*. • **Object Module**: See *WWB-COM Object Module* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*. • **Class Module**: See *WWB-COM Class Module* in the *WinWrap Basic Language* section of the *WinWrap Editor Help*.

Edit menu	Use the **Edit** menu to perform common editing tasks. This menu also provides specific functions for working with scripts For more information on specific script editing functions, see the *Edit Menu* topic in the *WinWrap Editor Help*, displayed from the **Index Family Editor Help** menu when viewing a script.
View menu	Use options on the **View** menu to display or hide the available explorer tabs. Scripting options are also available from this menu. For more information on specific script view functions, see the *View Menu* topic in the *WinWrap Editor Help* displayed from the **Index Family Editor Help** menu when viewing a script.
Macro menu	Displays options for working with macros, including running the complete macro, pausing the macro or module for later resume, and ending or terminating the execution of the macro or module. For more information on specific macro functions, see the topic *Macro Menu* in the *WinWrap Editor Help* displayed from the **Index Family EditorHelp** menu when viewing a script.
Debug menu	Displays options for debugging scripts. For more information on specific debug functions, see the topic *Debug Menu* in the *WinWrap Editor Help* displayed from the **Index Family EditorHelp** menu when viewing a script.
Help menu	Displays the *WinWrap Language Help*, *WinWrap Editor Help*, and information about the version of WinWrap included with the **Index Family Editor**.
Index Family Explorer panel	This panel displays a tree of all the index families defined for the project. Each family has an XINDEX file holding information about the index family, and a BAS file created to hold the family script. Expand the tree and double-click the index family BAS file to display the family script in the editing pane.

Script editor pane	The editing pane enables working with scripts. The default text displayed depends on the script type selected. Each open script is displayed as a tab in the editor pane. Use the editor to create and edit scripts, and associate objects with the script. • **Object**: Lists available objects with which the script can be associated in a listbox. Objects can include the index family, fields, tables, and the main table object. • **Proc**: Lists all events available for the selected object. Events already present in the script display in bold. If the **(General)** object is selected, all implemented events plus all user subroutines in the script are listed. Select an available object and then an available event to add a new event to the script, or to position the cursor in an existing subroutine.

5.3.4.5 Select Resources

When selecting an *OCR* engine configuration file or a filter, open the **Select Resources** window, which presents the following options:

Table 5-12: Select Resources Window

Element	Description
Global Resources tab	Global resources are predefined configuration files supplied with all recognition projects. Each engine configuration file is displayed in a table along with a description of the file, its type (barcode, hand printed, machine printed...) and its provider. Sort the engine configuration files by type or by provider using the **Sorted by** list (default value is **Single list**).
Local Resources tab	Local resources are engine configuration files you have created or customized specifically for the current project and saved to the following directory: `[Project directory]\Resources\OCR`. For help creating an engine configuration file, see "Adding Engine Configuration Files to the Project" on page 109.
Select button	Validates the selected engine configuration file.

Element	Description
Cancel button	Cancels any operation and closes the **Select Resources** window.

5.3.4.6 Index Family Editor Toolbar

The **Index Family Editor** toolbar includes several useful shortcuts to command commands. Mousing over an icon provides a tool tip explaining the function of the tool. This table provides a more detailed explanation of the tools. More icons are available when a script file is opened. For more information, see the WinWrap documentation.

Table 5-13: Index Family Editor Toolbar

Name	Description
New	Creates an index family or a script. • With the focus on the **Index Family Editor**, an index family is created. With the focus on a script editor, a macro is created. Otherwise, clicking on the drop down arrow displays these options: – Index Family – Macro – Code Module – Object Module – Class Module • With the focus on a script editor, a Macro is created by default. Otherwise, clicking on the drop down arrow displays these options: – Macro – Code Module – Object Module – Class Module
Open	Displays the **Open** window to select an existing index family or script to open.
Save	After a file is modified, this saves the modifications to the file.
Save All	Saves all open files.
Delete	Deletes index or table fields, or scripts, script text, or field property values.
Add	Adds index or table fields, depending on the current field view.

Name	Description
Move field up	Moves the selected field upwards in the list.
Move field down	Moves the selected field downwards in the list.

5.3.5 Project Options

The **Project Options** window provides access to most project settings. Either click the Project options icon from the toolbar, or select **File** > **Project Options**.

Table 5-14: Project Options Window

Element	Description
General tab	Displays project options like the project name, version (applicable only for Dispatcher Manager), production, report and backup directory location, creation date, and additional information about the project in the **Advanced Information** view.
Classification tab	Displays and modifies project settings including options related to the *OCR* engine, pre-indexing, and document codes. Adjusts threshold settings for the classification engine, such as pre-classification and decision rates values.
Text Matching tab	Displays and modifies **Text Matching** parameters for the project, such as image filters, OCR engines, confidence threshold and matching threshold. Fine tune these parameters or reset to default values by clicking **Advanced** and modify settings in the **Text Matching advanced parameters** window.
Folder Management tab	Select or modify project settings related to folder creation, folder populated field, and to enable a folder binding field.
Classification Edit tab	Select or modify options related to how the Identification application performs, including setting application options for the project, specifying keyboard shortcuts, and customizing user interface color schemes.
Recognition tab	Select or modify options related to recognition, such as the default threshold for graphic and textual anchors, table field detection, and free form image filters to be applied.

Element	Description
Standard OCR tab	Options for using the OCR data cache instead of running full OCR on PDF and PDF/A documents and image files.
OK button	Validates the configuration, and exits the **Project Options** window.
Cancel button	Cancels any operation and exits the **Project Options** window.

5.3.5.1 General Tab

The **General** tab displays fields for naming and versioning projects, defining production folders, and provides general project information.

Table 5-15: Project Options Window General Tab

Element	Description
Main options (applicable only for Dispatcher Manager)	Modify the project identification elements such as the project name, the project author, the name of the company and the project version number. When creating a project, give it a project version number. By default the project version number is set to 1.0. Each time you update the project, modify the project version number to distinguish the project evolutions. The version number is not incremented automatically.
Production options	
Production directory	The directory receiving production files when the project is sent to production.
Backup directory for production project	Keeps a copy of the project in production as a backup. This is applicable if the project has already been sent to production at least once.
Production report directory	If reporting is enabled, the report is sent to this directory. The production report is a *CSV* file, named `Report <date>` with the format *<YYYY-MM-DD-hh-mm>*. This file lists all the templates sent to production. There is a line per template and each line records the template ID, the code, the name and the index family.
Check all templates are linked to an index family before sending to production	Select to ensure that the procedure to move the project to production will not be interrupted.

Element	Description
Information	Displays general information about the project and advanced information on templates.
Notes	For adding comments.
Browse (...) button	Used to locate a directory.
Advanced information... button	Displays additional information about project templates.

5.3.5.2 Advanced Information

This window is available when you click the **Advanced information** button in the **General** tab of the **Project Options** window.

Table 5-16: Project Options Advanced Information Window

Element	Description
General information	Displays information such as project directory, size of the project in the production directory, and size of the project in the development directory.
Template list	List can be sorted by **Single list**, **Lists sorted by family**, or **List sorted by code**. Click  to display the composition of each group.
Template name	The template name.
Path	The path relative to the **Project directory**.
Index family	The index family name.
Code	The template code.
Frequency (%)	The (total images of the template / total images in the project) * 100.
Total images	The number of all *TIF* and *JPEG* files which are in the template folder (associated with the template)
Image base (Kb)	The combined size of the images associated with each template.
Export template list	Displays the **Save as** window to export data to a specified file.

5.3.5.3 Classification Tab

In the **Classification** tab, the following settings are available:

Table 5-17: Project Options Window Classification Tab

Element	Description
Engine options	
Test 180 rotations	Rotates the scanner output image by 180 if it is upside down. This option applies to standard and *HPA* templates.
Test 90 and 270 rotations	For documents scanned on both sides, changes the orientation of the scanner output images from landscape to portrait. This option applies to standard and HPA templates.
Test side flipping	For documents scanned on both sides, displays scanner output images in the correct order in Identification. This option applies to standard and HPA templates.
Assign handwritten documents to template	Assigns handwritten documents that are not classified to a given template.
Assign a template by default (if not recognized)	Assigns a default template to all non-classified documents to avoid sending the document to manual classification.
Removal of black edges	Automatically eliminates black edges from the scanned images. This option is available when creating a project with the **New Project Wizard**.
Enable text matching classification	Must be selected for text matching settings to be active.
HPA default search zone	Displays the height **H (mm)** and width **W (mm)** values that define the HPA search zone applied to HPA anchors when placed on an HPA template. Once placed, these values can be modified for each HPA anchor.

Element	Description
Advanced Engine Parameters	Displays the **Advanced parameters for classification engine** window for adjusting the pre-classification threshold of standard templates. It is highly recommended that you keep the default parameters. However, for some specific cases, you may want to adjust the pre-classification threshold and the decision threshold: to do so, follow the indications provided in the **Advanced parameters for classification engine** window. After adjusting any of these two thresholds, check the resulting pre-classification and decision rates by running a classification test.
Pre-index	Pre-indexing consists of running recognition on index fields during the classification phase in production and not just during the recognition phase. Those fields can be validated in the Identification module. Select fields from the available fields and click the **Add** button. The available fields are grouped by index family. As soon as a field is added to the list of **Selected fields**, it no longer appears in the list of **Available fields**. Click **OK** and the selected index field(s) to be pre-indexed are automatically proposed as folder binding fields in the scroll-down list in **File** > **Project Options** > **Folder management**.
Document code management	Define and import document codes used in the project. The digit in the **Used** column indicates how many times the template is used in the project. Click  to create a code. The template code must not exceed 30 characters. Click  to delete one or several codes. If one of the codes to be deleted is used in a template (1 in the **Used** column), then a window prompting for confirmation is displayed. Click  to edit a code.
Compilation cache	Optimizes the time it takes to compile a project.

Element	Description
Import codes	Import a list of codes from a TXT file containing one code per line. Imported codes are added to any existing codes. For Unicode support, the project designer must select the encoding format of the TXT file to load when importing data in Recognition Designer or Free Form Designer: • Autodetect • ANSI • UTF-7 • UTF-8 • Unicode (UTF-16 Little-Endian) • Big-Endian (UTF-16 Big-Endian)
Empty	Empties the cache directory.

5.3.5.4 Text Matching Tab

In the **Text Matching** tab, the following settings are available:

Table 5-18: Project Options Window Text Matching Tab

Element	Description
Image filters	
Reverse video zones	Detects reverse video text boxes in the image and changes their background from black to white and the characters from white to black so that they can be read by *OCR* engines. This filter is recommended for most projects. It does not significantly impact the processing time.
Matrix font	Detects automatically the presence of matrix characters on the image and make them bold so they are better read by OCR engines.
Table lines	Deletes all horizontal and vertical lines in the image. This filter is recommended for most projects whenever table characters overlap or touch the table borders.

Element	Description
Shaded areas	Detects all the shaded areas in the image and removes the shaded background (by removing pixels) without altering the other areas of the image. This filter is not recommended when shaded areas contain thin or very thin characters. These characters may be degraded and then poorly read by the OCR engines.
Text box reading	Segments the whole page into individual text lines that are then passed individually to the OCR engine. This filter is recommended for OCR engines whose line detection is not very good. Note: instead of calling the OCR engine once in the full page mode, the OCR engine is called as many times as there are text lines in the image so this can have license impact for some engines.
OCR options	
OCR engine	Select the OCR engine configuration file used for text matching. Click the button to the right of the text box to open the **Select Resources** window to select engine configuration files (`*.reco`) either from the **Global Resources** tab or from the **Local Resources** tab.
Engine confidence threshold	Define the confidence threshold value for the **OCR engine**. If no value is specified a default value of *<60>* is used. The confidence threshold value is the minimum confidence level value (or recognition rate value) that a character must obtain to be considered recognized. If a character obtains a confidence level value less than the confidence threshold value, the character is not recognized.
Browse (**...**)	Browse for directory and select an OCR engine configuration file.
Parameters	
Matching threshold	Corresponds to the graphic matching rate from which a given document is considered to be classified. The default value for this option is *<45>*%.

Element	Description
Minimum difference between two best candidate results	If the difference between the two matching values exceeds the **Minimum difference between two best candidate results** value, then the first template is selected. If there is not enough difference, the image is set in conflict. The default value for this option is *<20>*%.
As a second attempt, lower the matching threshold by	If the best match is lower than the matching threshold and both candidates are associated with the same template, a further test takes place to determine if the first candidate template rate is higher than **Matching Threshold** minus **As a second attempt, lower the matching threshold by**. If yes, the first template is selected. Otherwise the image is set as unclassified.The default value for this option is *<10>*%.
Advanced	Learn to use the advanced parameters in the topic "Speeding Up Processing" on page 87.
Default	Reverts back to the initial default values for all text matching parameters.

5.3.5.5 Folder Management Tab

The **Folder Management** tab features the settings to be specified for folders management.

Table 5-19: Project Options Window Folder Management Tab

Element	Description
Enable folder creation	Select so that documents are grouped into folders during production. A folder is a logical group of documents, often identified with separators in the production flow when running the Identification, extraction, and Completion modules. If this option is not selected, each document will form one individual folder.

Element	Description
Folder creation using a folder binding field	A folder binding field initiates the creation of a folder during classification. A comparison is made between values on different documents and a folder is created whenever the folder binding field value read on a document is different from the one read on the previous document (or if two documents have the same folder binding field value but different template codes). This option is most useful to combine multiple-page invoices based upon their invoice number. **Note:** There are two other methods to generate folders: the separators and the pre-indexed fields. For more information see "Setting Up Folder Management" on page 47.
Enable a folder binding field	Select this option and then scroll down the list to select the index field to be used as the folder binding field. The list box contains the index fields to be pre-indexed as defined in the **Classification** tab.
Folder populated field	Assigns one or more field values to all documents in the folder. For example, if an invoice date was identical in all the invoice pages in all the documents of a folder, then it would be very useful to automatically assign this value, after validation, to all the documents in the folder. **Note:** A field that has already been used as a pre-indexed field cannot be used as a folder binding field. This can cause problems in the Identification interface.
Folder populated field window	To enable folder populated fields and assign an index field to the whole folder, click + (this option is unavailable if the **Enable folder creation** option is not selected). To delete a field, select it and click −. The **Folder populated field** window displays. Select fields from the **Available fields** and click **Add**. The available fields are grouped by index family. As soon as a field is added to the list of **Selected fields**, it no longer appears in the list of available fields.

5.3.5.6 Classification Edit Tab

The **Classification Edit** tab features the settings of the manual classification in the Identification module.

Table 5-20: Project Options Window Classification Edit Tab

Element	Description
Application options	
Reject folder if one of its document is rejected	Rejecting one document in Identification rejects the whole folder that contains the document. Not selecting this option means that rejecting a document will only reject that document.
Display active folder thumbnails	All folder images are shown as thumbnails in Identification. Not selecting this option enables only the current image to be shown in Identification.
Document-code oriented keying	Only template codes can be typed in the identification zone of Identification (the list of the suggested templates only displays the template codes). If you do not select this option, then the operator can type template codes or names in the identification zone of Identification. The list of the suggested templates displays the template codes and names.
Display next document automatically	Only available if the option **Document-code oriented keying** is selected. If selected, as soon as a valid template code has been attributed to a document in Identification, the system automatically goes to the next document to be classified. If you do not select this option, the operator must press **ENTER** to go to the next document to be classified.
Go to next field automatically	Applies to fields for which the option **Character validation** is selected in the **Index Family**. If selected, a field is automatically validated if it contains no character in error in Identification. It enables faster data correction as the operator does not have to stop on fields that do not need to be corrected. In Identification, the operator can enable/disable this option by pressing **F9** (**F9** is the default hotkey; you can select another one in the **Keyboard Shortcut Settings** pane).

Element	Description
Show only generic template in template list	When selected, this option enables to hide graphical templates in the template list. This option can be useful for projects using the *PAL* feature. It enables the operator to save time selecting only collectable templates. Option not selected by default.
Confirm closing session after task completed:	A confirmation window opens at the end of each task asking the operator if he wants to close the session.
Default zoom	Sets a default zoom level to display images in Identification. Sets also a default zoom level for viewing pre-indexed unplaced fields. Each time an unplaced field is selected, the default zoom will be applied. Select **Full page**, **Page width**, **Page height** or a zoom percentage.
Persistent zoom	When a document is opened and a pre-indexed unplaced field is selected, the **Default zoom** is always used for that first field. Each time a new unplaced field is selected, the zoom level for that field is determined in one of two ways: • When **Persistent zoom** is activated, the current zoom level is used. In other words, if you reset the zoom level for a field, that becomes the active zoom level while viewing unplaced fields in the current document. This new zoom level persists until another zoom level is defined or the next document is opened. • When **Persistent zoom** is not activated, each selected field uses the **Default zoom** setting.
Keyboard shortcuts settings	Assign a shortcut to each function in Identification. For the operator, a recall of the defined hotkeys is available through a hover help which can be displayed by keeping the mouse cursor stationary over the associated buttons. **Note:** For **Reject document**, select **No reject** to disable document rejection in Identification. The  will be hidden from the operator in Identification.

Element	Description
Color settings	To change a color, select the color square to open a color look-up table in which you can then select another color. You can use color codes for a better display of even/odd folders and fields in Identification.

5.3.5.7 Recognition Tab

The **Recognition** tab provides options for selecting or modifying options related to recognition.

Table 5-21: Project Options Window Recognition Tab

Element	Description
Default threshold for graphic anchors	Type the default matching threshold for all local graphic anchors when they are placed for the first time on the template.
Default threshold for textual anchors	Type the default matching threshold for all local textual anchors when they are placed for the first time on the template.
Free Form image filters	Select the image filters to be applied to fields associated with free form settings. The free form image filters improve the accuracy of recognition by *OCR* engines. Filters run before OCR. Filters apply to the whole image even if the fields are not placed to cover the whole image. Filters apply to the image in memory (image files are not modified). Processing time is not significantly longer when images filters are selected. **Note:** These filters are the only filters that must be used for fields associated with free form settings; do not use any of the filters from the **Image CleanUp** tab in the **Indexing view** as these are to be used for fields for which zonal recognition is used.
Reverse video zones	Detects reverse video text boxes in the image and changes their background from black to white and the characters from white to black so that they can be read by OCR engines. This filter is recommended for most projects. It does not significantly impact the processing time.

Element	Description
Matrix font	Detects automatically the presence of matrix characters on the image and make them bold so they are better read by OCR engines.
Table lines	Deletes all horizontal and vertical lines in the image. This filter is recommended for most projects whenever table characters overlap or touch the table borders.
Shaded areas	Detects all the shaded areas in the image and removes the shaded background (by removing pixels) without altering the other areas of the image. This filter is not recommended when shaded areas contain thin or very thin characters. These characters may be degraded and then poorly read by the OCR engines.
Text box reading	Segments the whole page into individual text lines that are then passed individually to the OCR engine. This filter is recommended for OCR engines whose line detection is not very good. Note: instead of calling the OCR engine once in the full page mode, the OCR engine is called as many times as there are text lines in the image so this can have license impact for some engines.
Recognition quality	Depending on whether you want greater accuracy or speed: • **Accurate (default value)**: accurate, but slower recognition. Accurate is best when using free form recognition. • **Fast**: faster, but less accurate recognition. Fast is best when using free form recognition for text-based classification (namely keyword classification and text matching). If you select the **Fast** option, it is not recommended to select any free form image filters, otherwise you get a slower recognition.
OCR engine configuration	
OCR engine for table Free Form fields and textual classification	Specify the full-textOCR engine and confidence threshold to use in free-from table recognition and textual classification.
OCR engine for anchor text values	Specify the OCR engine and the confidence threshold to apply to anchor text values. This setting can be used to change to an OCR engine more suitable for non-Latin character sets.

Element	Description
Default OCR engine for rubberband	Specify the OCR engine and confidence threshold defaults to apply to rubberbanding.

5.3.5.8 Production Auto-Learning Tab

In the **Production Auto-Learning** tab, the following settings are available:

Table 5-22: Project Options Window Production Auto-Learning Tab

Element	Description
Collector	Select from these collector parameters: • **Activate Dispatcher Collector** - Enables the Collector module. • **Document storage** - Select the path and directories where collected images are to be saved. • **Delete collected documents older than** - Purges collected data x days after the learning process is performed. Option selected by default. • **Delete collected documents exceeding** - Purges collected data x GB after the learning process is performed. Option not selected by default.

Element	Description
Templates to be learned	When activating Collector, the **Templates to be learned** table displays and enables the selection of templates from which the unknown images must be collected. This table gives information on the template name, code and the index family linked to the templates to be learned. • **Assign Code From**: Select from these options: – **Use template code**: Assigns code from the template used to collect data. – **Use custom code**: Value defined by the project designer. – **None**: No code is assigned. – **Use field value**: Assigns code from an index field. This option is recommended to prevent conflicts between templates. Use to discriminate a document using a field value. • **Code Value**: Depends on whether the **Use custom code** or the **Use field value** is selected. – If **Use custom code** is selected, edit the box and key a value. – If **Use field value** is selected, **Code Value** lists all the index fields created in the index family.

Element	Description
Production Auto-Learning	Set the following options: • **Minimum number of documents required to create a template**: This number depends on whether you are creating a project for classification only or classification and indexation. For classification projects only, the minimum number of documents required can be 2. This enables a quick creation of templates and speed performances. For classification and indexing projects, we recommend a minimum number documents of 5. This number enables a high classification rate and returns relevant data to send to production. To create more accurate projects and of higher quality that is to say high classification and recognition rates, we recommend to set a minimum number of 8 documents. • **Maximum number of project backups**: A backup is a copy of the entire project with its image base and the compiling cache directory. *PAL* back ups the project each time new templates are automatically created by the Supervisor. Backup copies reside in the recognition project repository path. By selecting this option, a maximum number of backups is enabled. Default value is 10. So for example, if the number of project backups is higher than 10, then the oldest backup copies are deleted to keep only a maximum of 10 project backups. • **Maximum number of templates**: The maximum number of templates that can be in a project at any one time. When the maximum is reached, then new document types and variations are no longer learned. • **Template creation method**: a list for selecting preferred way of templates setup: **Textual templates**: enables good accuracy and reasonably small project size, **Graphical templates**: enables high speed; used when the image base is stable; **Textual and Graphical templates**: although the project gains better accuracy by using both types of templates, the

Element	Description
	project can grow in size and performance might be slower. • **Apply LIFFE settings on templates where table learning fails**: If no Table Wizard values are extracted, PAL can use the line item free form engine settings instead to place table fields automatically on templates. To do so, this option must be activated. If *LIFFE* settings are defined in the project, PAL automatically places a field at the top left of the template to be associated to a *DFT* file. For additional information on the line item free form engine, see "Creating Free Form Templates" on page 187. • For **Delete learned templates not used in production in the previous _ month(s)**, specify that you want to delete PAL templates that have not been used in production for the specified prior number of months. However, because PAL templates that have been modified by a person might contain valuable updates, they are not deleted. Because the deleted PAL templates no longer have to be compiled and loaded and the overall size of the recognition project is reduced, deployment and production performance can improve. **Note:** If the resulting day is not a valid day in the resulting month, then the last valid day of that month is used. For example, (in a non-leap-year) one month before March 31 is February 28. • **Send automatically to production**: Sends automatically the project to production if new templates have been created by the Supervisor. Make sure a production directory is defined in the **Project Options General** tab, otherwise this option is not available. This option is not selected by default. If it is not selected, new templates only apply to the project in development but updates are not available in production. This option enables the project designer to control that enough updates have been made in development to put the project in production.

5.3.5.9 Standard OCR tab

Property	Description
Enable OCR data cache from Standard OCR	When this option is selected, the following behavior for PDF and PDF/A documents (converted from Microsoft Office documents and original PDF and PDF/A documents only) and images is enabled (except for the OCR engines specified in **Select OCR engines to use instead of Standard OCR cache**): • Classification When performing textual, keyword, and text matching classification on PDF pages and images, the text (including coordinates and line/word separation) in the OCR data cache is used. In addition, PDF pages are not converted to images, which could result in better performance. **Note: Project Options** > **Recognition** > **Free Form image filters**, **Project Options** > **Text matching image filters**, and **Indexing** > **Index View** > **Image Clean Up** filters are skipped. • Identification and Completion For a PDF page, rubberbanding uses the OCR data cache. **Notes** – Annotations are disabled. – The PDF page cannot be rotated. • Extraction When performing extraction on PDF pages and images, zone and free-form recognition uses the OCR data cache. In addition, if PDF pages are not converted to images, then better performance could result. If the following elements do not exist on the page, then the PDF page is not converted to an image: – Table fields – Graphical anchors However, if the aforementioned elements do exist on the page, you could ignore them as follows: – To ignore graphical anchors, enable the **Disable graphical anchors for zonal fields** option.

Property	Description
	– To ignore table fields, disable assignment of table fields. **Note: Project Options** > **Recognition** > **Free Form image filters** and **Indexing** > **Index View** > **Image Clean Up** filters are skipped.
Apply text-based classification before graphical classification	With the OCR data cache, text-based classification (Textual, Keyword, and Text-Matching) could be faster than graphical classification. If the aforementioned is true, then select this option to improve performance.
Disable graphical anchors for zonal fields	Select this option to disable graphical search for zonal index or table field anchors, which prevents the loading and rendering of these input pages. Thus, enabling this option could result in improved performance. In addition, if a text value is assigned to the anchor, then the anchor is determined by the OCR data cache.
Select OCR engines to use instead of Standard OCR cache	Specify the OCR engine to use in place of the OCR data cache. The following images on PDF pages can only be recognized by OCR: • Barcode images • US and/or French check images • Checkboxes In addition, if PDF pages are composed of images only and free-form rules are optimized for a particular OCR engine, then the recognition accuracy of the OCR data cache would be reduced.

5.3.6 Template Properties

In the **Classification View**, right-click and select **Template Properties** from the **Template** list. The **Template Properties** appear at the bottom of the window.

Table 5-23: Template Properties Window

Element	Description
Name	Rename a template from the template list. Type a template name and press **ENTER** to validate the name. Rename several templates at the same time by selecting several templates in the **Template** list and typing a template name in the **Name** field. The template names will automatically inherit a suffix, for example *<Invoice>*1, *<Invoice>*2, etc.
Code	The template code is used to facilitate the work of the operator in Identification. If you create several templates and they all have the same fields to extract, you can assign them the same template code so that the operator does not need to know all the template names but can use the template code and classify them more rapidly in Identification. If several templates having the same code potentially match the document, the operator enters the document code to classify the document and is automatically determined which template best matches the document. If you want the operator to classify documents only by means of template codes in Identification, rather than allowing the use of template names, you must enable the function **Document-code oriented keying**. Then select one or several templates in the template list. Type a template code in the **Code** field. Press **ENTER** to validate the code name. For more information on this function see "Creating Templates Based on Template Codes" on page 61.
Index family	The **Index family** field enables assignment of an index family to a template. The index family list displays all index families available in the project.

Element	Description
Separator type	In Recognition Designer, a document and its attachments are kept together in a folder. Two methods are used to split the document flow into logical folders: separators and folder binding fields. The **Separator type** list offers you three separator types: • **Natural separator**: Used to define a document page as being the logical start of a folder. It is identified by the  symbol in the template list. A new folder is created each time a document matches a natural separator. • **Artificial separator**: Used to define a patch document or a template which has been created specifically to fix the limit of a new folder. It is identified by the  symbol in the templates list. • **No separator**: A document that matches a template that has no separator automatically belongs to the same folder as the previous image. If all the templates of the project have no separator, then each batch will create one folder to contain all the documents.

5.3.7 Edit Keyword Classification

Keyword classification is a text-based classification method. The **Edit Keyword Classification** window is displayed by selecting **Classification** > **Keyword Classification**. Documents are classified based on the keywords they contain, as defined in the **Edit Rules** pane.

Table 5-24: Edit Keyword Classification Window

Element	Description
Left pane	The left pane provides the different steps to define keyword rules: • "OCR Parameters" on page 298 • "Edit Rules" on page 299 • "Test" on page 300

Element	Description
Right pane	The right pane enables you to set up the *OCR* parameters, edit the keyword rules, and test the keyword classification, depending on the selection from the left pane. If an OCR data cache is being used, then **[Standard OCR]** is displayed in the **Content** pane.
Close button	Closes the **Edit Keyword Classification** window.

5.3.7.1 OCR Parameters

Defines the recognition engine to use and the page zone used for reading.

Table 5-25: Edit Keyword Classification OCR Parameters Pane

Element	Description
Reading zone	Zone where *OCR* reading is performed. The image of the first generic template in the project appears to the right of the reading zone. Display order is based on the document ID. The available zones are: • **Full page** • **Upper third** • **Middle third** • **Lower third** • **Custom size** enables definition of the size of the reading zone. Define the coordinates (X,Y) for the upper left corner of the zone and the zone height and width.
OCR engine	Select a full page OCR engine. Set the recognition engine confidence threshold. If no threshold is specified, all characters are recognized with a default threshold. The default threshold value depends on the selected OCR engine. For details on how the confidence threshold works, see "Recognition Engine Confidence Threshold" on page 110.

5.3.7.2 Edit Rules

Table 5-26: Edit Keyword Classification *OCR* Parameters Edit Rules Pane

Element	Description
Rules	Displays all the project rules. Click + to create a new rule. Right click in the Rules list and select **Export the List of Rules...** to generate a file that contains the list of keywords and associated parameters. Having the rules exported to a spreadsheet may help if the project contains lots of rules (100 or 150 for example) and you want to filter on them to search for redundant rules for example.
Edit a rule	Displays rule properties for the selected rule: • **Template**: Only displays passive or text matching templates. Select the template to associate to the keyword rules. If you do not want to associate any template to the selected keyword rules, then select **<None>** from the template list. • **Priority**: Sets a priority level for the current rule. If a document matches several keyword rules, the document is classified to the template that matches the rule with the higher priority level. There are three priority levels: **Standard**, **High** or **Maximum**. **Note:** If a document matches two rules from two different templates, and the two rules have the same priority level, then the document cannot be classified. • **Keywords nearby**: Select this option if you have at least created two keywords. This option enables you to scroll and select the maximum number of characters to be found between the two keywords for the image to be classified. • **Keywords** pane: Add or delete keywords from the rules. When you click + to add a keyword, the **Define Keywords** window appears.

5.3.7.3 Test

Table 5-27: Edit Keyword Classification *OCR* Parameters Test Pane

Element	Description
All rules	Uses all the rules created in the **Edit Rules** window.
Rule	Uses a specific rule from the **Edit Rules** window.
Results per image	The status bar indicates the total number of images and the total number of classified images. • **File**: displays the name of the image. • **OCR**: recognition can be performed using the OCR file or the engine configuration file selected in the **OCR Parameters** window. Before performing a test, this column displays either a red cross or a green tick: – Red cross: no OCR file exists or it does not match the engine configuration file selected in the **OCR Parameters** window. Thus, recognition is performed using the engine configuration file and an OCR file is automatically created and saved to the directory where the original images reside. – Green tick: the OCR file already exists and it is used for recognition. After the test is performed, the tick symbol indicates that recognition has been successfully performed on the image. • **Template**: displays the template name where the image is classified. This column remains empty if the image could not be tested or classified.

Element	Description
Results per rule	Displays the test results per rule: The **Rule** column displays the rule created in the **Edit Rules** window. • The **Template** column indicates if a template is associated to the rule and the **Priority** column indicates the priority associated to the rule in the **Edit Rules** window. • The **Keywords** column indicates the template name if the document has been classified, or it shows the **<Conflict>** string if several templates are potential candidates to classify the document. • The **Score** column indicates **OK** if all keywords have been found or it indicates the score (for example 2/3) which is the number of keywords found to the total number of keywords.
Content pane	This pane shows the results of the recognition performed on the reading zone as well as the results of the keyword search made on raw text (highlighted content). Select **CTRL+F** to carry out a search on the content read. A good example could be to use "ga" when searching for "garage" in the content read if the word "garage" has not yet been found, even though it is normally present in the read area. The search enables to check if OCR reading was done correctly or not.
Right pane	Displays the image associated to the rule.

5.3.7.4 Define Keyword

Displays when clicking the add icon to add a keyword to the **Keyword** list from the **Edit rules** pane.

Table 5-28: Edit Keyword Classification Define Keyword Window

Element	Description
Number	Number of keywords created and added to the list of rules. The number is automatically incremented.
Name	Name of the created keyword.

Element	Description
Category	Select one of the three categories: • **Constant**: Search a specific term, amount or date. Select **Case sensitive** if required. For constants, specify the **Hit threshold** percentage. The higher the threshold, the closer the characters should be to the specified value. For example, if the threshold is 80%, 80% of the characters must match. If the threshold is 100%, 100% of the characters must match. • **AN Format**: Accepts alphanumeric characters. Type a valid format syntax like *<1N5A7X2C>* for example. This format indicates that a value with the format one numeric, five alphabetic, seven alphabetic or numeric, and two characters of any type will be valid. • **Regular expression**: Use regular expressions to search patterns. Learn about regular expressions in the topic "Regular Expression Syntax Elements" on page 97. • **Fuzzy regular expression**: Unlike the **Regular expression** option, fuzzy regular expressions take advantage of the **Hit threshold** option, which enables a single regular expression to have a wider range of text matches. For more information, see "Fuzzy Regular Expressions" on page 100.

Element	Description
Parameters	Specify the following parameters: • **Value**: Enter a term, alphanumerical characters or regular expressions formats, according to the category you have selected. • **Case sensitive**: Indicates that uppercase and lowercase letters are treated as distinct characters. • **Automatically fix value**: Enables OCR engines to automatically substitute low-confidence scanned characters with an appropriate alternate character. However, if an appropriate alternate is not available, then a question mark is substituted. For example: – If a fuzzy regular expression specifies digits only for a zip code (for example, `\d{5}`) and the following conditions are also true: ○ A zero is recognized as the letter `O`. ○ A zero exists as an alternate character. Then, a zero is substituted. **Note:** The alternate characters available for substitution are provided by each particular OCR engine; that is, the available alternate characters can vary between OCR engines. • **Isolated word**: Sets that the keyword is a single string, that is preceded and followed by spaces. • **Hit threshold**: Sets a tolerance with respect to read values. For example, when searching the constant “amount”, if the read value is “amovnt(=91%)”, it is accepted if the threshold value is set to 90%. • **Excluded**: Sets the keyword as an anti-keyword. An anti-keyword serves to validate the template: the document matches the template when the anti-keyword is NOT found in the document.

Element	Description
Test button	Tests the current keyword. For help using the **Regular Expression Builder** see "Search Word in Content" on page 304 if searching for a constant or "Regular Expression Builder" on page 305 if searching an *AN* format, a regular expression, or a fuzzy regular expression.
OK button	**OK** is only available after adding a keyword in the **Keywords** list.
Add button	Adds the created keyword to the **Keywords** list.
Exit button	Exits the **Define Keywords** window.

5.3.7.5 Search Word in Content

When the **Constant Category** is selected for a defined keyword, click **Test** in the **Define Keywords** window to search for a word.

Table 5-29: Edit Keyword Classification Search Word in Content Window

Element	Description
Word searched	Displays the keyword to find in the **Search zone**.
Isolated word	Finds a single string that is preceded and followed by spaces.
Case sensitive	Uppercase and lowercase letters are treated as distinct characters.
Search zone	In the **Search zone**, type text that contains the searched word. Run the test by clicking  . If the searched word is found, it is highlighted in the text, and results are displayed at the bottom of the **Search zone**, for example: Found at position 354 (score 100). This option ensures that the searched word is correctly recognized during recognition.
Apply button	Applies tuning made in the current window to the keyword settings for constants and regular expressions.
Cancel button	Cancels any operation and exits the **Search Word in Content** window.

5.3.7.6 Regular Expression Builder

When the **AN Format**, **Fuzzy regular expression**, or **Regular expression Category** is selected for a defined keyword, click **Test** in the **Define Keywords** window to search for a word.

Table 5-30: Edit Keyword Classification Regular Expression Builder Window

Element	Description
Regular expression	Displays the regular expression to find in a document.
Isolated word	Finds a single string that is preceded and followed by spaces.
Search zone	In the **Search zone**, type text that contains the regular expression or the format searched. Run the test by clicking  . If the searched word is found, it is highlighted in the text, and results are displayed at the bottom of the **Search zone**, for example: "Found at position 354 (score 100)." This option ensures that the searched word is correctly recognized during recognition.
Apply	Applies tuning made in the current window to the keyword settings for constants and regular expressions.
Cancel	Cancels any operation and exits the **Search Word in Content** window.

5.3.8 Search a Template

If a project contains several templates, enter search criteria to search for a specific template by selecting **Classification** > **Search** with the project open.

Table 5-31: Search a Template Window

Element	Description
ID equals	Represents the template identifier.
Name includes	Enter the name or part of the template name.
Code starts with	Enter the beginning of the template code, and **<No code>** searches the templates to which no code is attributed.
Family starts with	Enter the beginning of the index family name, and **<No family>** searches the templates to which no index family is attributed.

Element	Description
Search results pane	Displayed at the bottom left of the main window after the search is complete and the **Search a Template** window closes.
Reset button	Empties the field values.
OK button	Validates and exits the **Search a Template** window.
Cancel button	Cancels any operation and exits the **Search a Template** window.

5.3.9 Classification Test Results

When running a classification test, the **Classification Test Results** window appears. This window enables you to check the classification performance, visualize the classified and unclassified images, check the pre-classification and decision rates and export test results and classified/unclassified images.

Table 5-32: Classification Test Results Window

Element	Description
Project	Gives the name of the project.
Total test images	Represents the number of tested images.
Total time	This timer indicates the total time the process will last.
Time remaining	This timer indicates the remaining time before the process ends.
Processing speed	• **Images/hour**: Indicates the number of images classified per hour. • **Images/second**: Indicates the number of images classified per second.
Performances	Represents, in percentage, the number of images that were successfully classified, those that could not be classified, and those for which more than one candidate template was identified.
Processing speed (images/hour)	This line graph compares two variables: the number of classified images per hour.
Recognition percentage	The recognition percentage graph represents a curve with the number of classified images.

Element	Description
Template representativeness	**Template representativeness**: The template list enumerates the number of templates to display in the horizontal bar graph. By default the value is set to five templates. Horizontal bar graph: By default it represents the five best covered templates. Each bar represents a template and indicates the number of images per template. **Note:** You can only modify the number of templates while the test is running. As soon as it ends, the template list is dimmed and cannot be used.
OK button	Validates and exits the **Classification Test Results** window.
Details button	Displays the **Details** pane including the processing speed, the recognition percentage and the template representativeness zones.

Related Topics

"Testing Classification" on page 104

5.3.9.1 Classified Tab

Displays the images successfully classified to the project templates. The number of classified images is also indicated in brackets beside the tab name.

Note: The **Rotation** and **Side flipping** columns only appear if these options are selected in the **Project Options** window.

Table 5-33: Classification Test Results Window Classified Tab

Element	Description
File column	Indicates the complete path of the image.
Name column	Indicates the name of the template to which the image is classified. When you double-click a template name, a thumbnail of the image displays.
Code column	Indicates the code of the template to which the image is classified.
Rotation	and buttons indicate a rotation and the direction of the rotation.

Element	Description
Side flipping	 indicates a double-sided inversion.

5.3.9.2 To Confirm Tab

Unclassified images are displayed either in the **To Confirm** tab or the **Not Classified** tab. The **To Confirm** tab displays unclassified images for which the system finds several candidate templates. Those candidate templates are displayed in a specific pane.

Table 5-34: Classification Test Results Window To Confirm Tab

Element	Description
File column	Indicates the complete path of the image.
Conflict column	Displays the ID numbers of the candidate templates. The ID number is found in the **Classification View** in the **Template Properties**.
Results for current document pane	Appears at the bottom left of the window. It displays the **Name** and **Code** of each candidate templates or the pre-classification and decision rate columns. Double-click a template name to display a thumbnail of the image.

5.3.9.3 Not Classified Tab

Displays the images that cannot be classified. For these images, the system does not supply a list of possible templates and the operator has to select the template in Identification module.

Table 5-35: Classification Test Results Window Not Classified Tab

Element	Description
File column	This column indicates the complete path of the unclassified images.
Image 1	Displays the image reference.

5.3.10 Table Field Unit Test

Unit tests evaluate the current field settings, filters, *OCR* results, and scripts (contextual rules).

Table 5-36: Table Field Unit Test Window

Element	Description
Test Base menu	Enables you to load more images or to reload images from templates and to close the **Table Field Unit Test** window.
Zoom menu	Enables you to zoom in, zoom out or center the current image.
Resources menu	Enables you to refresh the resources.
Image base	Presents the list of images with their associated test results. The image base assumes the following default parameters: **OK?** and **Line count**. **OK?** displays **Yes** if the field format set in the **Index Family Editor** window is respected. If not, the **OK?** column displays **No**. The **Line count** column displays the number of lines detected during recognition. The number of selected images is displayed at the bottom of the image base. It also displays the time needed to analyze the image, and the percentage of matching images.
Image pane	Displays the current image with the detected lines in a green frame.
Image after filtering pane	Displays the image result after filtering. To apply filters see "Applying Filters to Improve Recognition" on page 111.
Zoom pane	Zoom from 0.25 to 10. Default value is set to 1.
Segmentation pane	Displays each character details: character value and confidence level. If the confidence level does not match 100%, an alternative candidate is proposed. When you move the cursor over the confidence level, a hint appears with the alternative candidate.

Element	Description
Table fields pane	Displays the list of lines detected on the image. The table fields pane assumes the following default parameters: **Reco**, **Then CTRL**, **OK?** and the **Height**. Other columns may appear according to the options set in the **Index Family Editor** window.
Launch test button	Runs the **Table Field Unit Test**.

5.3.11 Anchor Unit Test

Unit tests evaluate the current field settings, filters, *OCR* results, and scripts (contextual rules).

Table 5-37: Anchor Unit Test Window

Element	Description
Test Base menu	Enables you to load more images from a directory, reload images from a template, reload images from a template test base, or close the **Anchor unit test** window.
Zoom menu	Enables you to zoom in, zoom out, or center the current image.
Image base	Presents the list of images with their associated test results. The number of selected images is displayed at the bottom of the image base. It also displays the time needed to analyze the image, and the percentage of matching images. The image base assumes the following default parameters: • **OK?** displays **OK** if the field format set in the **Index Family Editor** window is respected. • The **Results** column represents the matching percentage. The matching percentage can be set in the **Index View**.
Image pane	Displays the current image. If the anchor appears in a green square, the test hits the target. If the anchor appears in a red square, the test misses the target.
Launch test button	Runs the **Anchor Unit Test**.

5.3.12 Field Unit Test

Unit tests evaluate the current field settings, filters, *OCR* results, and scripts (contextual rules).

Table 5-38: Field Unit Test Window

Element	Description
Test Base menu	Enables you to load more images or to reload images from templates, reload images from a template test base, or close the **Field Unit Test** window.
Zoom menu	Enables you to zoom in, zoom out center the current image.
Resources menu	Enables refreshing the list of resources.
Image base	Presents the list of images with their associated test results. The number of selected images is displayed at the bottom of the image base. It also displays the time needed to analyze the image, and the percentage of matching images. The image base assumes the following default parameters • The **Recognition** column displays **Yes** or **No** accordingly if the field has been recognized. • **After controls** displays the results after the *VBA* script control. • **OK?** displays **Yes** if the field format set in the **Index Family Editor** window is respected. If not, the **OK?** column displays **No**.
Image pane	Displays the current image with the recognized characters.
Image after filtering pane	Displays the result of the image after filtering. It represents the dimmed zone on top of the index field, on the current image. For help using filters, see "Applying Filters to Improve Recognition" on page 111.
Segmentation pane	Displays each character details and the confidence level. If the confidence level does not match 100%, an alternative candidate is proposed. When you move the cursor over the confidence level, a tool tip appears with the alternative candidate.
Zoom pane	The zoom function has a capacity from 0.25 to 10.

Element	Description
Launch test button	Runs the **Field Unit Test**.

5.3.13 Users Connected to Current Project

This window is available when you select **File** > **Connected users**.

Table 5-39: Users Connected to Current Project Window

Element	Description
Table of connected users	Checks if any user is already working on the project. This table presents the name of the user, the references of his post, the connection date and hour. **Note:** Several users should not amend the same project at the same time. If several users need to create templates for the same project at the same time, it is better that they work on separate projects and that templates are imported into the project by means of the Template Import Wizard.
OK button	Closes the **Users Connected to Current Project** window.
Refresh button	Refreshes the users list.

5.4 Free Form Designer

Free Form Designer enables configuration and testing of user defined settings to detect and extract values of index fields and table data on documents where specific data does not appear in the same location on the documents. These are considered unstructured documents where fields cannot be placed on standard templates. Unstructured documents are often classified using text-based classification methods such as keyword classification or test matching.

The **Free Form Designer** window **Settings** pane displays options for defining **Target data formats**, **Anchor findings**, **Full text relations**, **Full text tables**, **Order Relation Definitions**, and **Field-Specific Types**. The functions available from these windows, combined with the options in the **OCR Reading** and **Search Keywords** panes are used to create a Free Form Designer project.

Table 5-40: Free Form Designer Windows

Element	Description
Settings	Defines parameters for **Full text fields** including **Target date formats**, **Anchor findings**, **Full text relations**, and **Full text tables**. Available windows, menus, and toolbars enable customization and testing of a wide variety of settings that control how Free Form Designer identifies and extracts data from unstructured documents.
OCR Reading	Opens or creates a base of test images on which optical character recognition is performed. Use the **Test Base Manager** to load sample images and perform recognition to obtain *OCR* results that can be used to customize and refine Free Form Designer project settings, and evaluate and recognition accuracy.
Search Keywords	Displays results of a file definition test (**Settings** > **File definition test** ()), along with information about the images in the test base such as field information, OCR results, potential hypotheses and the accuracy of keyword recognition. Use this information to refine project settings to improve accuracy and data extraction during recognition.

5.4.1 Settings

The **Settings** window defines parameters for **Full text fields** and **Full text relations**.

Table 5-41: Free Form Designer Settings Window

Element	Description
Tree structure	This template tree structure enables you to create and set up **Full text fields**, **Full text relations** and, if necessary, a **Full text table**. **Full text fields** are composed of two elements: • **Target data formats**: The element found in the document. It can be a date (a delivery date or an invoice date) or a number (an invoice number, a file number or a customer number). • **Anchor findings**: Composed of keywords and associated words that enable detection of the position of the target data format. For complex target data formats, several anchor findings may be necessary. **Note:** The name of the full text field must be exactly the same as the name of the field in the index family.
Parameters pane	Provides options for setting up **Full text fields**, **Full text relations** and a **Full text table**, based on the option selected in the tree structure pane.
File menu	From the **File** menu, create a definition file, reopen a definition file, save your settings, edit Free Form Designer options, link a full text field to an index family or close Free Form Designer. • **New**: Creates a new definition file. • **Open**: Opens an existing definition file. • **Re-open**: Reloads a definition file among the previously opened definition files. • **Save**: Saves modifications to the open file. • **Save as**: Saves the definition file in the following path: `\\<Recogniton project directory>\Resources\OCR\`. • **Options**: Opens the **Options** window. • **Link to an index family**: Links a template to an index family. • **Exit**: Closes Free Form Designer.

Element	Description
Edit menu	Add, delete, duplicate, or reorder a full text field, anchor, keyword, relation, full text table or column. • **Add**: Adds an element to the tree structure. • **Delete**: Deletes an item from the tree structure. • **Move up**: Moves an item up in the order of the tree structure. • **Move down**: Moves an item down in the order of the tree structure. • **Copy**: Duplicate the selected item and settings to the Clipboard. • **Paste**: Pastes the copied item and settings as a new element in the tree.
Tools menu	Define field-specific types and check the consistency of the definition file: • **Edit field-specific types**: Opens the Edit Field-Specific Types window to select, edit or create field-specific types. A field-specific type combines regular expressions and constants. • **Definition File Analyzer**: Opens the **Definition File Analyzer** window for testing the validity of the definition file.
Help menu	Displays help, version and copyright information.
Settings toolbar	Use the **Settings** toolbar to create definition files, open existing definition files, save definition files, edit options, move, delete or add items in the tree structure, run unit tests or edit field-specific types.

Related Topics

"Full Text Fields" on page 316

"Defining Full Text Relations" on page 184

5.4.1.1 Full Text Fields

Table 5-42: Free Form Designer Settings Full Text Fields Pane

Element	Description
Left pane	Displays the **Settings**, **OCR Reading**, and **Search Keyword** options, each of which are used to create a Free Form Designer project.
Tree pane	Displays available nodes. Select the desired node and click **Add** on the toolbar to add a new element of the type selected. Add defined elements, such as full text fields, anchor findings, full text relationships, and full text tables. The tree is populated as elements are added.
Right pane	When a full text field, anchor finding, full text table, or other node is created and selected in the tree pane, the **Parameters** pane for that element displays editable parameters and options available for the selected element.

5.4.1.2 Target Data Formats

To create or modify a **Target data format**, run Free Form Designer and select **Settings**. Then expand **Full text fields** select the **Target data formats** node. Click  in the toolbar to create a format. The new data format displays in the **Target data formats** node and the **Parameters** pane appears. A full text field is defined by three parameters:

- **Name**: The name of the full text field must be exactly the same as the name of the field in the index family
- **Target data formats**: The target data format identifies the element found in the document by type. The format can be a date, such as a delivery date or an invoice date, a number, such as an invoice, file, or customer number, or a piece of text, such as a company or customer name. A full text field may have several **Target data formats** with unique parameters defined by type.
- **Anchor findings**: The anchor findings use keywords and associated words to help the application detect the position of the target data format. For complex target data formats, several anchor findings may be necessary.

Table 5-43: Free Form Designer Full Text Fields Target Data Formats Parameters Pane

Element	Description
Constant	Use constants to search strings as target data formats.

Element	Description
Regular expression	Use regular expressions when searching patterns as target data formats. A regular expression is a string used to describe or match a set of characters, according to certain syntax rules. During recognition or validation steps, regular expressions can, for example, search for specific characters, the position of characters in a string, or specific grouping of characters.
Fuzzy Regular Expression	Uses a combination of a regular expression and the **Hit threshold** option, which enables a single regular expression to have more text matches than one defined in the **Regular expression** option. For more information, see "Fuzzy Regular Expressions" on page 100.
Field-specific type	A **Field-specific type** combines regular expressions and constants. Constants are longer to search than regular expressions and many constants in a definition file will decrease processing speed. Select and edit field-specific types and create field-specific types. **Note:** Use the `Value` variable to write a script to format the output data values. For example, to format different date formats such as 01.02.2004, 01-02-04 and 01-02-2004, apply a unique output format <*DD\MM\YYYY*>, resulting in 01\02\2004. To open the script editor from Free Form Designer, select **Tools >Edit field-specific types** and select the **Output Format** tab.
Isolated word	Select this box to find a single string that is preceded and followed by spaces.
Case sensitive	Select this box to treat uppercase and lowercase letters as distinct characters. **Note:** This option is not available for **Regular expression** and **Field-specific type**.

Element	Description
Automatically fix value	Select this box to enable OCR engines to automatically substitute low-confidence scanned characters with an appropriate alternate character. However, if an appropriate alternate is not available, then a question mark is substituted. For example: • If a fuzzy regular expression specifies digits only for a zip code (for example, `\d{5}`) and the following conditions are also true: – A zero is recognized as the letter `O`. – A zero exists as an alternate character. Then, a zero is substituted. **Note:** The alternate characters available for substitution are provided by each particular OCR engine; that is, the available alternate characters can vary between OCR engines.
Value	Type in a string, a regular expression or a label, depending on the **Type** you have selected.
File	This option enables you to select a field-specific type definition file. **Note:** This option is not available for Constants and Regular expressions.
Description	Type in any comments required.
Hit threshold	The higher the threshold, the closer the chain of characters must be to the specified value. This means that if the threshold is 80%, then 80% of the characters must match. If the threshold is 100%, then 100% of the characters must match. For example, the threshold is set to 90% and the searched keyword is “amount”: if the read value is “amovnt” (= 91%), the read value will be accepted as a keyword.
Test button	The **Test** button opens the tool that corresponds to the type.
Select button	Opens the **Selection of a Field-Specific Type** window.

5.4.1.3 Anchor Findings

Anchor findings detect the position of the target data format. An anchor finding consists of keywords and associated words. Multiple anchor findings can be defined to increase precision during recognition. Keywords and associated words are defined by:

- The type of value: **Constant**, **Regular expression**, **Fuzzy regular expression**, or a combination of constants and regular expressions known as a **Field-specific type**.
- The relative position of the elements.
- The location of the element, as specified by the selected search zone.

Table 5-44: Free Form Designer Full Text Fields Anchor Findings Parameters Pane

Element	Description
Search zone	This is the zone in which *OCR* reading is performed. The available zones are full page, upper third, middle third, lower third and custom size. The custom size zone enables you to define exactly the size of the search zone. Define the zone height and width as well as X,Y coordinates corresponding to the upper left corner.
Description	For entering optional comments.

Related Topics

"Defining Keywords" on page 177

"Defining Associated Words" on page 179

"Testing Free Form Rules for Index Fields" on page 160

5.4.1.4 Keywords and Associated Words Panes

Keywords are the words to search for on the document. Associated words can validate or invalidate the keyword. For example, if "smoker" is specified as a keyword, and the associated words are "yes" and "no", "yes" can be set to validate "smoker", and "no" can be set to invalidate "smoker".

The Keywords and Associated words panes are nearly identical.

Table 5-45: Free Form Designer Anchor Findings Keywords and Associated Words Panes

Element	Description
Type	Use this selection to specify the type and format of the values used. Available types include: • **Constant**: You use constants when you want to search strings as target data formats. • **Regular expression**: Use regular expressions to search for data patterns as target data formats. A regular expression is a string used to describe or match a set of characters, according to certain syntax rules. During recognition or validation steps, regular expressions can, for example, search for specific characters, the position of characters in a string, or specific grouping of characters. • **Fuzzy regular expression**: Unlike the **Regular expression** option, fuzzy regular expressions take advantage of the **Hit threshold** option, which enables a single regular expression to have a wider range of text matches. For more information, see "Fuzzy Regular Expressions" on page 100. • **Field-specific type**: Field-specific types combine regular expressions and constants. Constants require more processing time to locate than regular expressions, thus many constants in a definition file will decrease processing speed. **Note:** Use the `Value` variable to write a script to format the output data values. For example, to format different date formats such as 01.02.2004, 01-02-04 and 01-02-2004, apply a unique output format such as *<DD\MM\YYYY>*, that would result in 01\02\2004. To open the script editor from Free Form Designer, select **Tools** > **Edit field-specific types** and select the **Output Format** tab.
Isolated word	Select this box to find a single string that is preceded and followed by spaces.

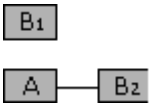
Element	Description
Case sensitive	Select this box to treat uppercase and lowercase letters treated as distinct characters. **Note:** This option is not available for regular expressions or field-specific types.
Value	Type in a string, a regular expression or a label, depending on the **Type** selected.
File	Select a field-specific type definition file. **Note:** This option is not available for constants or regular expressions.
Description	Type any necessary comments.
Hit threshold	The higher the threshold, the closer the chain of characters must be to the specified value. In other words, if the threshold is *<80>*%, then 80% of the characters must match. For example, the threshold is set to *<90>*% and the searched keyword is "amount": if the read value is "amovnt" (= 91%), the read value will be accepted as a keyword.
Position of target data relative to keyword **Position of keyword relative to associated word**	Specify the position of target data relative to keyword or the keyword relative to the associated word to limit the area searched and are identical to those in the Position in Relation to Keyword window. Click Add to specify these parameters. As an example, if the date (target data) always displays to the right of the keyword "DATE" and occurs within a specific distance, define the position relative to the associated word. In this example, specify the position of the target data to the right at a maximum distance of *<50>* mm from the keyword "DATE". Defining relative positions helps increase precision during recognition and minimize false positives.
Action of the associated word when it is present	Validates the keyword if the presence of the associated word specifies that the keyword is valid, or invalidates the keyword if the associated word specifies that the found keyword is not valid.

5.4.1.5 Full Text Relations

Full text relations improve accuracy in finding full text fields. They are based either on field alignment, or their relative positions, else on rules defined through a script that processes the relation using field values. A relation is processed for all the possible combinations of fields in the relation. The aim of a relation with a script is to define and apply a rule to the candidate field values (combinations) found for the set of fields assigned to the relation. For more information on scripting rules, see the *Programming Reference Guide*.

Table 5-46: Free Form Designer Full Text Relations Alignment Parameters Pane

Element	Description
Relation type	• **Alignment**: The relation between two fields is defined by their relative orientation. For example, field A is assumed to be to the left of field B or field D is under field C. • **Script**: The aim of a relation with a script is to define and apply a rule to the candidate field values (combinations) found for the set of fields assigned to the relation. Field values cannot be modified by the relation if the latter is confirmed. For more information on scripting rules, see the *Programming Reference Guide*.

Element	Description
Score	A score is calculated for each combination and the sum of scores for all relations gives the score of the combination. After all relations have been processed, the combination with the best score is kept as the output value. Keep the default score value of *<100>*, or, if necessary, specify a different positive value. The following example shows score settings on an alignment relation where A is assumed to be to the left of B: • Hypothesis No. 1 = A-B1 • Hypothesis No. 2 = A-B2  • No score is applied: The first hypothesis A-B1 automatically takes the score of 0 and the score decreases for the next hypotheses (Hypothesis No. 2 A-B2 takes -1). Hypothesis No. 1 has the best score although it does not match the alignment condition. • A score of 100 is applied: Hypothesis No. 2 A-B2 takes a score of 100 as it matches the alignment. This hypothesis has the best score and is kept.
Stop once a relation is confirmed	Select this option so that the relation will not be processed after the first combination that matches the relation is found. This combination will be kept as output value if no other relation exists. Clear this option if you want to process the relations for all the combinations. **Note:** When the relation uses scripting (for example, to check a keyword or retrieve a value in a database), selecting this option enables faster processing because the first combination that matches the relation is often the best.

Element	Description
Limit processing time for the relation (ms)	Select this option to set a limit on the time allowed to process the relation. The default value is *<1000>* milliseconds (one second) and values can be set from *<1>* to *<99999>* milliseconds. Clear this option to eliminate a processing time limit. However, in a typical scenario the processing time per image should be limited if the definition file results in a large number of hypotheses that take a long time to process.
Positioning fields	Drag and drop the first field from the **List of available fields** to the central cell of the **Positioning fields** area. Drag and drop the second field to the cell that corresponds to the correct position relative to the first field.
List of available fields	Lists the available fields you have created in the **Full text fields** tree structure.
Maximum/Minimum distance (mm)	These are the maximum and minimum distances in millimeters between the edges of each field. To evaluate the distance between a keyword and its target, go to the **OCR Reading** window or to the **Search Keywords** window, hold down the left-button of the mouse and draw a frame between the keyword and its target on the image. Hold down the left-button of the mouse and read the values **L** (length) and **H** (height) that appear in the status bar.

Related Topics

"Defining Associated Words" on page 179

"Script" on page 324

5.4.1.6 Script

Table 5-47: Free Form Designer Full Text Relations Script Parameters Pane

Element	Description
Script editor pane	The editing pane enables working with scripts. • **Object**: not used for full text relations. There is only one object available in full text relation scripting: `DpFreeFormFields` is the collection of free form fields associated to the current relation. `DpFreeFormFields` has two properties: – *`Item`* (Type = String or Integer): This is a `DpFreeFormField` called by its name or its rank in the **Fields** pane (starting from 0). It is a read-only property. – *`Count`* (Type = Integer): This is the number of fields in the collection. It is a read-only property. `DpFreeFormField` is one item of `DpFreeFormFields` and has two properties: – *`Value`* (Type = String): This is the field candidate value for the relation. It is a read/write property. – *`Name`* (Type = String): This is the field name. It is a read-only property. • **Proc**: Lists all functions present in the script. It can be the full text relation scripts or other functions included in the relation scripts. When a function is selected in the list, it takes focus in the script editor.
Fields pane	Displays the fields associated to the relation selected in the tree structure. Click **+** or **-** to add or remove fields.
Test Field Value pane	The **Fields** column displays the fields associated to the full text relation selected in the tree structure. The **Input field values** column displays the value which you entered and the **Output field values** column displays the results of the relation test for each field.
Test Result pane	The green box indicates that the relation script is successful and the red box indicates that the relation script failed.

Element	Description
Run Test button	Runs a test on the full text relation selected in the tree structure: checks the coherence of the relation script and applies the full text relation to the **Input field values**.

5.4.1.7 Full Text Table

Select the **Template** folder at the top of the tree, right-click, and select **Add** > **Add a full text table**.

Table 5-48: Free Form Designer Settings Full Text Table Window

Element	Description
Primary rows definitions	Specify the combination of columns to be found in a row for the row to be considered a primary row. When no primary row is found, the document is not processed.
Order relations	The **Order relations** table lists the **Name** and **Description** of each relation. The **Name** column lists the number of order relations, and the **Description** column displays the items selected in each order relation. Click to add an order relation, or  to edit a relation, and the **Order Relation Definition** window displays. To help *LIFFE* detect the columns in the document and exclude those columns that will not be extracted, specify the order in which the columns appear in the document and define relations. Relations specify that there are additional columns before or between columns you want to extract. For example: if a row has the following configuration: <Quantity> <????> <Amount> and the relation is “Quantity” “Unit Price” “Amount”, then it is assumed that “????” is the “Unit Price”.

Element	Description
Script relations	In the **Script relations** table the **Name** column lists the number of script relations, and the **Columns** column displays the items selected in each script relation. Click  to add a script relation, or  to edit an existing relation and the **Script Relation Definition** window displays. Click  to edit the *VBA* project script file. To help LIFFE detect the columns in the document, you can write a VBA script to specify a relation to be matched between the columns. For example: the typical arithmetical relation in invoices can be implemented by the following script relation: `Quantity * UnitPrice = Amount`.
Grouping column	Merge the data of the secondary lines with the data of the primary row. Select the column you want to group in the list box.

5.4.1.8 Edit Field-Specific Types

This window is available when you select **Edit Field-Specific Type** from the **Tools** menu of the Free Form Designer **Settings** pane.

Table 5-49: Edit Field-Specific Types Window

Element	Description
File menu	Creates a new field-specific type definition file, opens an existing one, saves any modifications required, opens the **Options** window and closes the **Edit Field-Specific Types** window. • **New**: Creates a new field-specific type definition file. • **Open**: Opens an existing field-specific type definition file. • **Save**: Saves new or modified field-specific type definition files. • **Save As**: Saves the definition file or field-specific type file (*TFT*) in *XML* format. You must save it to the following path: *<Recognition project directory>* `\Resources\OCR` • **Options**: Opens the **Options** window. • **Exit**: Closes the **Edit Field-Specific Types** window.
Test menu	To run a test based on the settings defined: • **Unit Test**: Opens the **Regular Expression Builder** or the **Search Word in Content** window depending on whether the **Definition** type selected in the **Expressions** tab is a **Regular expression**, **Fuzzy regular expression**, or **Constant**. This option enables you to find a regular expression or a string in a content. • **General Test**: Opens the **Field-Specific Type Test** window. This window enables you to find both regular expressions and constants in content. If regular expressions and constants are found, they are highlighted in the text, and the results are displayed at the bottom of the search area.

Element	Description
Tools menu	Builds an expression list for formats or for keywords. • **Build an Expression List for Formats**: Creates a list of formats in a text file from which you can build a list of regular expressions. Since the regular expressions syntax is complex, this option enables you to automatically convert the list of formats into regular expressions. • **Build an Expression List for Keywords**: Creates a list of keywords in a text file from which you can build a list of regular expressions. Since the regular expression syntax is complex, this option enables you to automatically convert the list of keywords into regular expressions. **Note:** Create separate files for formats and keywords. Lines can contain digit values, special characters and one or several words. You can enter as many lines as you want and lines have no size limits. However, the longer the file, the more time it takes to process and the longer the lines, the less relevant the regular expressions.
Help menu	Displays help options for the WinWrap Editor including *WinWrap Language Help*, *WinWrap Editor Help*, and information about the WinWrap Editor, such as the version and copyright.
Expressions tab	Lists defined expressions, enables expressions to be built and tested, allows expressions to be defined as a **Regular expression**, **Fuzzy regular expression**, or **Constant**, and enables specifying a **Hit threshold (%)** for constants, as well as the **Priority**, **Value** and **Description** for each expression.
Scripting tab	Use the script editor to control field-specific types with a script. For help understanding and working with field specific types, see "Understanding Field-Specific Types" on page 171.
Edit Field-Specific Types toolbar	The toolbar enables common functions such as creating or opening a field-specific type file, saving the file, testing expressions, and building expressions.

5.4.1.9 Expressions Tab

Table 5-50: Edit Field-Specific Types Expressions Tab

Element	Description
List of expressions	Displays the list of formats, keywords or regular expressions and their associated settings available in the **Definition** pane. The **List of expressions** table displays the following columns: • **Number** • **Value/Format** • **Type** • **Threshold (%)** • **Case** • **Priority** • **Example/Description** **Note:** When running keyword search, keywords are searched by orientation, then by priority and finally by distance.

Element	Description
Definition	• **Regular expression**: Use regular expressions to search for data patterns as target data formats. A regular expression is a string used to describe or match a set of characters, according to certain syntax rules. During recognition or validation steps, regular expressions can, for example, search for specific characters, the position of characters in a string, or specific grouping of characters. • **Fuzzy regular expression**: Unlike the **Regular expression** option, fuzzy regular expressions take advantage of the **Hit threshold** option, which enables a single regular expression to have a wider range of text matches. For more information, see "Fuzzy Regular Expressions" on page 100. • **Constant**: Use constants to search strings as target data formats. When selected, the **Hit threshold (%)** and **Case sensitive** properties are available. – **Hit threshold (%)**: Leave the default hit threshold (*<90>*%) or set a threshold from *<0>* to *<100>*. For example, if you search for the keyword "date", set the hit threshold to *<75>*% if you want to keep values such as "oate" or "dote". – **Case sensitive**: Select to treat uppercase and lowercase letters as distinct characters. • **Priority**: – **Regular expression** or **fuzzy regular expression**: The default priority is *<1>*. When the default is used for all regular expressions, they are searched in the order in which they are defined in the list of expressions. Otherwise, expressions with priority *<1>* are searched first, expressions with priority *<2>* are searched second, and so on. Regular expressions having the same priority level are processed together in groups of 100. When a constant is defined, regular expressions are processed in multiple groups.

Element	Description
	– **Constant**: The default priority is *<1>*. When the default is used for all constants, they are searched in the order in which they are defined in the list of expressions. Otherwise, constants with priority *<1>* are searched first, constants with priority *<1>* are searched second, and so on. • **Value**: Type in a string or a regular expression. • **Description**: Type in any comments required.

5.4.1.10 Scripting Tab

Use the scripting tab to write and edit scripts that control field-specific types. For help using the editor, see the *Winwrap Language Help* and *WinWrap Editor Help* files available from the **Help** menu in the **Edit Field-Specific Types** window.

Table 5-51: Edit Field-Specific Types Scripting Tab

Element	Description
Object	Select an object from the **Object** list box to add an event for the object.
Proc	Select an available event from the **Proc** listbox to insert the event script automatically. Use the `FieldSpecificTypeFormat` function to write a script formatting output data values. For example, to consistently format different date formats such as 01.02.2004, 01-02-04 and 01-02-2004, apply a unique output format such as <*DD\MM\YYYY*>, resulting in all dates formatted *<01\02\2004>*.
Scripting pane	Displays the scripting language and the content of the script. Write and edit scripts for field-specific types in this pane.
Scripting toolbar	Enables a variety of common scripting and Clipboard functions to facilitate building scripts. These buttons are similar to those in the project and index family script editors, and tool tips describe the function of each button. For more information about toolbar options, see the *WinWrap Editor Help* file displayed from the **Help** menu.

5.4.1.11 Position in Relation to Keyword

This window contains settings to define the position of the target data relative to the keyword and the position of the associated word relative to the keyword.

Table 5-52: Position in Relation to Keyword Window

Element	Description
Orientation	Sets the direction of search as one of the following: • **Full page** is recommended when both the position and the distance of the target with respect to the keyword are unknown so the target needs to be searched in the whole document. This slows down processing. In this case, the **Minimum** and **Maximum** distances are not taken into account. • **All directions** is recommended when the target is near the keyword but in no specific direction. In this case, the target is searched in all directions with respect to the keyword and within the specified **Minimum** and **Maximum** distances. **Note:** To reduce the number of candidates found, unselect the option **Accept characters between key and target** (see this option next). • **Specific direction** is recommended when both the direction and distance of the target location are known. Select one of the directional arrows. **Note:** When the selected direction is right or left (so horizontal), the target is found even if it is not perfectly aligned horizontally with the keyword. This enables correct detection even on skewed documents.

Element	Description
Distance	The maximum and minimum distances (in mm) are calculated from the nearest edge of the keyword text box to the nearest edge of the target data text box. For example, when the target data is located at the right of the keyword in the reference image, the distance is calculated between the left edge of the target data text box and the right edge of the keyword text box. Default values are **Minimum** = 0 and **Maximum** = 50. In the horizontal direction, a small overlap is authorized between the keyword text box and the target text box enabling the target to be retrieved even if it slightly overlaps the keyword text box. Another possible use of the distance setting is when the keyword and the target data overlap because they are searched using regular expressions that are not specific enough. In this example, the regular expression `(?i)\t?[^\r\n\t]{4,250}` is not specific enough and the keyword "N/REF" (in green) is found as part of the target data (in blue).  When such an overlapping cannot be avoided by setting a more specific regular expression, a solution to retrieve the target data is to set the orientation of the keyword to **All directions** and the **Maximum** distance to 1; this will ensure that the keyword is fully included in the overall target data. **Note:** Another solution in this case is to write a script that removes that part of the target data that corresponds to the keyword ("N/REF" in this example).

Element	Description
Classify target words by distance	This option is available to define the position of the target data relative to the keyword. This option is recommended when multiple targets can be found within the minimum and maximum distance interval. This option enables prioritizing between multiple targets as a function of their distance to the keyword. In some cases, the priority is given to the target that is closest to the keyword and in other cases the priority is given to the target that is the farthest from the keyword. In most projects, it is the target that is closest and in some very specific cases it is the one that is fastest. Select **Ascending** to keep the target that is closest to the keyword. Select **Descending** to keep the target that is farthest from the keyword. In the following example, there are three target candidates (in the blue frames) found for the keyword (in the yellow frame). The three candidates are 0.01, 817 and 813.28. If **Ascending** is selected, free form will prioritize 0.01 before 817 and before 813.28 to build the hypotheses and so 0.01 has the highest score. If **Descending** is selected, the candidates are kept in the order 813.28, 817 and 0.01 to build the list of hypotheses and so 813.28 has the highest score. The list of hypotheses and their scores display in the **Search Keywords** pane, in the **List of hypotheses** tab. 
Accept characters between key and target	This option is available if you have selected a **Specific direction** that is left, right, up or down. This option is selected by default. This option is recommended to detect the target when the distance from the keyword can be large. In this case it is recommended to set a distance from the keyword that is long enough for the target to be found. If this option is cleared, targets are ignored if there is no blank space between the keyword and the target. In the graphical example above, clearing this option means that the “0.01” target found is the only valid target.

Related Topics

"Creating and Testing Free Form Rules for Line Item Extraction" on page 163

"Testing Free Form Rules for Index Fields" on page 160

5.4.1.12 Order Relation Definition

Table 5-53: Free Form Designer Order Relation Definition Window

Element	Description
Relation name	Type in a name for the new order relation to create.
Available columns	Displays the list of available columns that appear in the document.
Order relation	Specifies the order in which the columns appear in the document.
Display order relation	Checks the order of the elements in the relation.

Related Topics

"Creating and Testing Free Form Rules for Line Item Extraction" on page 163

"Defining Full Text Relations" on page 184

5.4.1.13 Selection of a Field-Specific Type

The **Selection of a Field-Specific Type** window is displayed from the Free Form Designer **Settings** pane. When a specific field , keyword, or other element is selected, the **Parameters** pane for that field displays. Click **Select** from the **Parameters** pane to display the following elements in the **Selection of a Field-Specific Type** window:

Table 5-54: Selection of a Field-Specific Type

Element	Description
Number column	Enumerates the field-specific type definition files.
File column	Displays the list of the field-specific type definition files (TFT).
Version column	Displays the version number of the field-specific type definition file. This version number can be modified in the **Options** window from the **Edit Field-Specific Types** window.

Element	Description
Description column	Displays any description associated to a field-specific type definition file. This description can be modified in the **Options** window from the **Edit Field-Specific Types** window.
Description zone	Displays the description associated to the selected field-specific type definition file.
OK button	Validates your selection and exits the **Selection of a Field-Specific Type** window.
Cancel button	Cancels any operation and exits the **Selection of a Field-Specific Type** window.

5.4.1.14 Options

The **Options** window appears in Free Form Designer when you click **Options** on the **Settings** window toolbar or select **File** > **Options**.

Table 5-55: Free Form Designer Options Window

Element	Description
Version Information tab	Enables you to edit the version number of a field-specific type definition file. In the **Major** and **Minor** list box, select the new version number you want to associate to your field-specific type definition file.
Summary View tab	Enables you to edit a description associated to a field-specific type definition file. Type in any description required.
OK button	Validates the modifications and exits the **Options** window.
Cancel button	Cancels any operation and exits the **Options** window.

5.4.2 OCR Reading

After defining full text fields and relations, you can generate *OCR* data in the interface of Free Form Designer and test the free form settings. Test the free form settings on a test base of at least 1000 images. The test base should be different from the development base. At the end of the test, export the test results and use them as reference data to further compare them to other results

Table 5-56: Free Form Designer OCR Reading Pane

Element	Description
OCR menu	Opens a results file, saves results, opens a test base, runs recognition, aborts recognition process, reads the current image, coordinates the list and closes Free Form Designer window. • **Open Results File**: Imports OCR data and avoid carrying out OCR reading. If no OCR data is imported, OCR reading is carried out when you start the test. When you export OCR data, two options are available: – **Import all binary result files from a tree structure**: Use this option to select one directory and load all the OCR files contained in this directory – **Import binary result files**: Use this option to select one or several OCR files to be loaded. **Note:** This test is for keyword rules only. If you want to test keyword classification, you must run a classification test. • **Save Results**: Saves the OCR reading (so that they are available for next tests). OCR data is exported to a binary format. One OCR file is created per image. The `*.ocr` files are generated in the same directory location as the images. • **Open test base**: Loads the images for which you want to generate OCR data. Opens the Test Base Manager window. • **Run recognition**: Generates OCR data on all the loaded images. • **Abort recognition process**: Stops OCR and load other images, or to select another search zone. • **Read current image**: Generates OCR data for the current image. • **Coordinate list**: Displays the **Coordinates** pane that contain the coordinates (up, down, left, right) of each character (value). To reach a position, enter it in the **Position** area and click the **Search** button to go to this position in the **Coordinates** pane. • **Exit**: Exits Free Form Designer.

Element	Description
Display menu	The **Display** menu enables you to zoom in or zoom out on the image displayed.
Help menu	Opens the Help.
Images	Presents the list of images with its associated test result. The image base assumes the following default parameters: **Time (s)**. The **Time (s)** column in the **Images** base indicates the reading time in seconds.
Image pane	Displays the image selected in the **Images** base.
Summary View tab	When OCR data is generated, the **Summary View** tab indicates the total number of images processed and the average processing time per image.
Coordinates pane	Displays the **Coordinates** pane that contains the coordinates (up, down, left, right) of each character (value). To reach a position, enter it in the **Position** area and click the **Search** button to go to this position in the **Coordinates** pane.
Content tab	When OCR data is generated, the **Content** tab contains the OCR raw data. You may need to copy OCR data from the **Content** tab when you define target data formats and anchor findings. If an OCR data cache is being used, then **[Standard OCR]** is displayed in the **Content** pane.
Rubber Band tab	To use rubber band, hold down the right mouse button and draw a frame over the area of the image on which you want to run OCR. Select the **Rubber Band** tab to display the OCR results.

Element	Description
OCR options	• **OCR engine**: Select a full text OCR engine. You can use the following full pages: Western OCR, General-Use OCR, and Advanced OCR/ICR. For more information on selecting the OCR engine, see "Recognition Types Supported by Recognition Engines" on page 429. • **Full page**: OCR reading over the whole page. • **Select zone on the image**: Use the mouse to draw the reading zone on the current image. The selected zone appears in a red frame on the image.  **Caution** Make sure the field placed on the template covers the same zone as the zone you have selected as search zone for the test. • **Recognition quality**: Select **Accurate** or **Fast**. • **Enable OCR data cache from Standard OCR module**: If this option is set, then you can open PDF files as a test base, which also loads their associated OCR data caches.  **Note:** This is the same option as the one on the **Project Options** > **Standard OCR** tab. For more information, see "Setting Up Extraction" on page 232. If **Enable OCR data cache from Standard OCR module** is enabled in Recognition Designer and Free-Form Designer is opened from Recognition Designer, then this option is also set. • **Image filters**: Select the **Image filters** checkbox to apply filters to the images before OCR is run. Select the image filters to generate the OCR data for the test in Free Form Designer.  **Note:** These are not the filters that apply in production. To select image filters to be used in

Element	Description
	production, see "Recognition Tab" on page 287.
OCR Reading toolbar	The **OCR Reading** toolbar enables you to open a results file, save results, open a test base, run recognition, abort recognition process, zoom in and zoom out on the image.

Related Topics

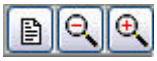
"Testing Free Form Rules for Index Fields" on page 160

"Creating and Testing Free Form Rules for Line Item Extraction" on page 163

5.4.2.1 Test Base Manager

This window enables loading of images to generate *OCR* data. Select **Open test base** from the **OCR** menu of the **OCR Reading** window in Free Form Designer.

Table 5-57: Test Base Manager Window

Element	Description
Left pane	Displays the list of the selected images.
Right pane	Displays the image selected in the image base.
Open a base button	Opens an existing test base.
Save button	Saves edits to an existing test base, or saves a new test base to a specified location.
Add images button	Adds images to the test base.
Add tree button	Adds a tree to the test base.
Batch images button	Loads batch images exclusively when the project is used with b-Wize. Access to a b-Wize database is compulsory.
Empty list button	Deletes the images listed in the image base.
OK button	Validates the selection, and exits the **Test Base Manager** window.
Cancel button	Cancels any operation and exits the **Test Base Manager** window.
	Zooms in, zooms out or restores the default size of the image.

5.4.3 Search Keywords

Table 5-58: Free Form Designer Search Keywords Window

Element	Description
Search menu	• **Open results file**: Imports *OCR* data and avoid carrying out OCR reading. If no OCR data is imported, OCR reading is carried out when you start the test. When you export OCR data, two options are available: – **Import all binary result files from a tree structure**: Use this option to select one directory and load all the OCR files contained in this directory – **Import binary result files**: Use this option to select one or several OCR files to be loaded. **Note:** This test is for keyword rules only. To test keyword classification, run a Classification Test. • **Start search**: Starts searching keywords on the images base. • **Search on current image**: Starts searching keywords on the current image. • **Exit**: Exits Free Form Designer.
Display menu	Enables zooming in and out on the selected image, display the **Content** tab and the results of the **Rubber Band** tab. • **Display the content**: Select this option to display the **Content** tab. If this option is cleared, the **Content** tab will not be available. • **Display the Result of the Rubber Band**: Select this option to display the **Rubber Band** tab. If this option is cleared, the **Rubber Band** tab will not be available.

Element	Description
Tools menu	Exports the results, compares test results to a reference data, quickly searches a text string and searches the next occurrence. • **Export Results**: Exports the results to a text file for comparing the current test results with previous test results. The exported text file includes a title line and <*N*> lines of data. It has the following columns: – <*N*>: Indicates the image number – **Image**: Indicates the image name – **Result**: Contains 1 if the line was recognized and 0 if it was not recognized, and columns which indicate the title of the field and the read value. To export the results to compare them with previous test results, make sure data is sorted on the <*N*> column (image number in descending order 1, 2, 3). **Example 5-1: Result file on index fields** N;Image;Result;InvoiceDate 1;ISP0001.tif;1;22.06.01 2;ISP0002.tif;1;August 29 2000 3;ISP0003.tif;1;30\03\00 4;ISP0004.tif;0;- **Example 5-2: Result file from a test on table data** <image1_name>, <number_of_row> <column1_name>, <column2_name>, . , <columnN_name> <value_co1_row1>, <value_col2_row1>, , <value_colN_row1> <value1_col1_row2>, <value_col2_row2>, , <value_colN_row1>

Element	Description
	<image2_name>, <number_of_row> <column1_name>, <column2_name>, . , <columnN_name>  • **Compare to a reference**: Select the text file containing the reference data, that is the results you have exported. Then the **Comparison of Results with Reference Data** window appears. • **Search quickly a text string**: This helps you check if a keyword has been correctly recognized. If the test results are not as expected, you may want to first test OCR functionality. • **Search next**: Searches the next occurrence in the **Content** tab.
Help menu	Opens the Help.
Images	Presents the list of images with its associated test result. The image base assumes the following default parameters: **Time (s)**. The **Time (s)** column in the **Images** base indicates the reading time in seconds. Also, the **Image** base displays another column.
Image pane	Displays the image selected in the **Images** base.
Summary View tab	When OCR data is generated, the **Summary View** tab indicates the total number of images processed and the average processing time per image.
List of Hypotheses tab	The **List of Hypotheses** tab indicates the relations applied to each hypothesis. Select a hypothesis from the list to display the area on the current image that matches the hypothesis. For the selected hypothesis, the relations applied and scores obtained are indicated in the frame to the right of the **List of Hypotheses** tab.

Element	Description
Full text table tabs	The following tabs only appear when running a **Unit Test** on the full text table node from the **Settings** window: • **Primary rows** • **Per row and per relation** • **Per row** • **Whole table** • **Final results** For more information, see "Full Text Table Tabs" on page 351
Detail of a Field tab	The **Detail of a Field** tab indicates the values found for the searched element. Icons are: •  for the full text field •  for the keyword •  for the target data format
Content tab	Displays the recognition results. You can search the **Content** tab using the menu **Tools** > **Search Quickly a Text String**. This helps you check if a keyword has been correctly recognized. If the test results are not as expected, you may want to first test OCR functionality.
Rubber Band tab	The **Rubber Band** tab shows the recognition results which you can obtain by selecting a specify portion of the image with the right-click of the mouse.
Search Keywords toolbar	Opens a results file, starts the search, exports results to a TXT file, compares results with a reference data, starts a quick search of a text string in the content of an image and zooms in/out the selected image. For Unicode support, the project designer must select the encoding format of the TXT file to load when importing data in Recognition Designer or Free Form Designer: • Autodetect • ANSI • UTF-7 • UTF-8 • Unicode (UTF-16 Little-Endian) • Big-Endian (UTF-16 Big-Endian)

Related Topics

"Testing Free Form Rules for Index Fields" on page 160

"Creating and Testing Free Form Rules for Line Item Extraction" on page 163

5.4.3.1 Comparison of Full Text Table Results with Reference Data

This window enables you to compare full text table results with reference data. Select **Compare to a reference** from the **Tools** menu of the **Search Keywords** window in Free Form Designer to display this window.

Table 5-59: Free Form Designer Search Keywords Comparison of Full Text Table Results with Reference Data Window

Element	Description
Select a column	To select a column in the list box. • If running a comparison of test results of a column node, select a column in the **Select a column** list. • If using reference data generated during a test on a column node, select only this column in the **Select a column** list. • If using reference data generated during a test on a full text table, select all the columns of the full text table in the **Select a column** list.
Results pane	This table displays the following values: • **Number** • **Image name** • **Reference values** • **Found values** • **Correct values** • **Wrong values** • **Unfound values** • **Wrongly found values** • **Right rate** • **Wrong rate** • **Missed rate** • **Wrongly found rate**

Element	Description
Summary pane	Statistics for the whole set of image appear in the **Summary** pane. • **Tested image**: Number of tested images • **Reference images**: Number of images in the reference data file • **Unfound images**: Number of tested images not found in the reference data file • **Referenced values**: Number of items of the column in the reference data • **Found values**: Number of items of the column returned by the *LIFFE* • **Correct values**: Number of correct values in the column returned by the LIFFE • **Wrong values**: Number of wrong values in the column returned by the LIFFE • **Unfound values**: Number of unfound values in the column returned by the LIFFE • **Wrongly found values**: Number of wrongly found values in the column returned by the LIFFE • **Correct rate**: Percentage of correct values • **Wrong rate**: Percentage of wrong values • **Unfound rate**: Percentage of unfound values • **Wrongly found rate**: Percentage of wrongly found values • **Correct mean rate**: Mean rate of correct values on all the images • **Wrong mean rate**: Mean rate of wrong values on all the images • **Unfound mean rate**: Mean rate of unfound values on all the images • **Wrongly found mean rate**: Mean rate of wrongly found values on all the images • **Tested image**: Number of tested images • **Reference images**: Number of images in the reference data file • **Unfound images**: Number of tested images not found in the reference data file • **Referenced values**: Number of items of the full text table in the reference data • **Found values**: Total number of items on all the columns returned by the LIFFE

Element	Description
	• **Correct values**: Total number of correct values on all the columns returned by the LIFFE • **Wrong values**: Total number of wrong values on all the columns defined in the definition file • **Unfound values**: Total number of unfound values on all the columns defined in the definition file • **Wrongly found values**: Total number of wrongly found values on all the columns defined in the definition file • **Correct rate**: Percentage of correct values • **Wrong rate**: Percentage of wrong values • **Unfound rate**: Percentage of unfound values • **Wrongly found rate**: Percentage of wrongly found values • **Correct mean rate**: Mean rate of correct values on all the images • **Wrong mean rate**: Mean rate of wrong values on all the images • **Unfound mean rate**: Mean rate of unfound values on all the images • **Wrongly found mean rate**: Mean rate of wrongly found values on all the images
Close button	Closes the **Comparison of Full Text Table Results with Reference Data** window.

5.4.3.2 Comparison of Full Text Field Results with Reference Data

This window enables comparison of full text field results with reference data. To display this window, select **Tools** > **Search Keywords** > **Compare to a reference** in Free Form Designer.

Table 5-60: Free Form Designer Search Keywords Comparison of Full Text Field Results with Reference Data Window

Element	Description
Select a Full text field	Selects a full text field in the list box.

Element	Description
Results pane	This table assumes the following parameters: **Number**, **Image**, **Search result**, **Reference data** and **Identical**. The **Identical** column indicates for each image if the results are consistent or inconsistent with the text file. You can examine each image to find out what causes the results to be different (recognition error or lack of an associated word to validate or to invalidate the keyword). Symbols are: ✔ indicates the results are consistent with the text file indicates the results are not consistent with the text file.

Element	Description
Summary pane	The **Summary** pane displays the following statistical values and rates: • **Correct values**: Number of images out of the total image number on which field values are identical in both search results and reference data. If the number of identical items is lower than expected, make sure that the images are sorted in the same order (sort by the <*N*> column) in both current project and reference data. When exporting the results (to the TXT file), data is exported in the order in which it is displayed. • **Including non empty**: Subset of preceding quantity, including the number of images on which field values are not empty and which are identical in both search results and reference data. • **Wrong and non empty values**: Number of images on which field values are found (not empty), but are different from the reference data. • **Empty references**: Number of images on which field values are empty (blank fields) in the reference data. • **Unfound values**: Number of images on which field values are empty in the search result (blank fields) and should be not empty. • **Wrongly found values**: Number of images for which the system has found some field values whereas they are not present in the reference data. • **Correct rate**: Percentage of correct fields (including blank fields). • **Wrong rate**: Percentage of incorrect and non blank fields. • **Unfound rate**: Percentage of missed fields. • **Wrongly found rate**: Percentage of wrongly found fields.
 button	Results are consistent with the text file.
 button	Results are not consistent with the text file.
OK button	Exits the **Comparison of Full Text Field Results with Reference Data** window.

5.4.3.3 Full Text Table Tabs

To test the relations between columns of a full text table, select the **Full text table** node in the tree structure of the **Settings** window. The **Search Keywords** window appears and displays the test results on the image in several tabs.

Table 5-61: Unit Test Results for Full Text Tables

Element	Description
Primary Row tab	This tab shows the rows found for each primary row definition. The **Primary Row** definition pane indicates the different primary row definitions with for each definition, the column names involved in the primary row definition. Those column names also appear in the first column of the **Primary Row** tab. The second and next columns are for the text boxes found in the row (one column per text box found). The **Lines** pane lists all the rows found for the primary row definition. The rows with a tick symbol are the rows that match the primary row definition. • When selecting a row that does not match the primary row definition, all the text boxes found for this row appear green-framed on the image and no field values appear in the **Primary Row** tab. • When selecting a row that matches the primary row definition, the **Primary Row** tab indicates the field values of the text boxes that match the primary row definition (they are red-framed on the image); no field value is displayed for text boxes that do not match the primary row definition (they are blue-framed on the image). The field value that appears in the column header is the raw *OCR* output. The field value that appears in each row is the value after the application of the field-specific type definition file (.tft) selected for the target data format in the full text table.
Per Row and Per Relation tab	Displays the column hypotheses that match the relations for each primary row. Select a relation in the **Relations** pane to view only the hypotheses corresponding to this relation. Select a hypothesis to view all the text boxes found for this hypothesis red-framed on the image.

Element	Description
Per Row tab	Displays for each primary row, the combined column hypotheses, in other words, the column hypotheses that result from a merge of all the column hypotheses found by applying the relations. The score in the **Score** column is the sum of all the scores of the combined column hypotheses. The default initial score is 10 (this initial value cannot be modified). Each time a relation is matched, its score is incremented another 10 points. The combined column hypotheses are displayed in a decreasing score order. Next to the **Score** column, there is one column per column in the primary row definition and finally one column per relation. The tick in the relation column indicates that the combined column hypothesis matches the relation. Select a hypothesis to view all the text boxes found for this hypothesis red-framed on the image.
Whole Table tab	Displays all the column hypotheses in the table. These column hypotheses result from a merge of the combined column hypotheses for each primary row. Select a hypothesis to view all the text boxes found for this hypothesis red-framed on the image. The score in the parentheses right to the caption of the hypothesis is the sum of all the scores of the combined column hypotheses plus a propagation score (the latter is automatically calculated and applied by Recognition Designer; it cannot be modified). Each row of the hypothesis has a row number displayed in the **Row** column. Next to the **Row** column, there is one column per column in the primary row definition.
Final Results tab	This tab displays per each primary row, the column values found and the secondary rows. On the image, the secondary rows that are above the primary row are blue-framed and those that are below the primary row are green-framed. Select a primary row or a secondary row in the **Final Results** tab to view all the text boxes found for this selection red-framed on the image.

5.4.4 Script Relation Definition

Table 5-62: Free Form Designer Script Relation Definition Window

Element	Description
Relation name	Type in a name for the new script relation to create. Each relation must have a different name. **Note:** If you rename a relation name, it is automatically renamed in the script.
Available columns	Displays the list of available columns that appear in the document.
Columns in relation	Lists the columns in the relation.
Add button	Adds an item from the **Available columns** table to the **Columns in relation** table.
Delete button	Deletes the selected item from the **Columns in relation** table.
Edit Script button	Opens the *VBA* project script file. For more information on the VBA script files, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.
Move buttons	Changes the order of the elements in the relation. They apply in the order in which they are listed in the table.
OK button	Validates the settings and exits the **Script Relation Definition** window.
Cancel button	Cancels any operation and exits the **Script Relation Definition** window.

5.5 HPA Template Editor

The HPA Template Editor is useful for achieving higher classification rates. Right-click the templates list and select **New HPA Template**, select the images to use, and click **OK** to display the **HPA Template Editor** window.

Note: Expand the window if all the parameters do not appear.

Table 5-63: HPA Template Editor

Element	Description
Template menu	Closes the HPA Template Editor.
Edit menu	Cancels or repeats creating or deleting anchors.

Element	Description
Image menu	Zooms in, zooms out or centers on the current selection.
Test menu	The **Evaluation on Image Base** evaluates the settings of the pre-classification threshold and high precision anchors. The images from the template image base are loaded automatically.
HPA Template Editor toolbar	Provides buttons to add anchors and to evaluate the image base.
Left pane	Displays the selected reference image. Draw typical anchors on the image to discriminate non corresponding images and to obtain a higher classification rate.
Right pane	Displays all the following options needed to set and clean up the selected anchors.
Pre-classification threshold	An expanding bar that represents the matching rate required for a document to be a potential candidate. A high pre-classification threshold is recommended. Default threshold is set to *<70>*%.
High precision anchors	The number of anchors placed on the image.
Minimum hit number	Represents the value required for the document to be classified. This parameter is set to **All** by default meaning that the condition for the document to match the *HPA* template is that all the anchors hit the mark.
Anchor size	The size of the anchors placed on the image. Small anchors are recommended. Anchor size must not exceed *<10>* x *<10>* mm. If the size exceeds the recommended size, a warning message displays at the bottom right of the **HPA Template Editor** window.
Search zone	The boundary zone for searching the anchor. Smaller zones are recommended. Default value can only be set in **File** > **Project Options** > **Classification** > **HPA default search zone**. A minimum value of *<30>* x *<30>* is recommended.
Reverse answer	If this option is checked, the anchor is considered valid when the matching rate is inferior to the anchoring threshold.
Strengthen	Thickens the anchor to facilitate searching when the anchors are printed in thin characters.

Element	Description
Mandatory	For a document to be classified, the anchor in the document must match the selected template anchor.
Binary conversion threshold	This option exclusively applies to color documents, otherwise it is not displayed in the interface. When anchors are placed on colored images and if this option is selected then anchors are converted into black and white.
Anchoring threshold	Represents the threshold for an anchor to be detected or not. Default value is *<70>*%.

5.6 HPA Template Test

Assesses the pre-classification threshold and the high precision anchors specified for *HPA* templates.

Table 5-64: HPA Template Test Window

Element	Description
File menu	Opens or reopens a tree structure or a template base, to export images, to use the selected image as a referenced image and to close the **HPA Template Test** window. • **Open**: – **Tree Structure**: loads directories containing image base. – **Both-Sided Tree Structure**: loads project images in duplex mode. – **Batch Images**: Option only available if the project is used with b-Wize. Access to the b-Wize database is compulsory. – **Images**: loads an image at a time. – **Template Base**: loads the images of the HPA template selected in the template list. • **Re-Open**: Reloads an image tree structure among the previously opened image tree structure. • **Export Images**: Exports images in the template bank or in a folder. **In the Template Bank** copies the selected images to the template image base (the images are copied without any confirmation message). **In a Folder** copies the selected images to an existing or a new directory. • **Close**: Closes the **HPA Template Test** window.
Edit menu	Selects or deletes documents from the file list of the **HPA Template Test** window.
Display menu	Zooms in, zooms out or restore the default zoom of the image.
Test menu	Runs the test on the image base.

Element	Description
Left pane	Displays the evaluation on image base results: • The **File** and the **Back page file** columns list the images and the paths where they are saved. • The **Classified** column displays a green tick if the image is classified and a red cross if it is not. By default, the classified images appear at the top of the list. Select the column headers to display the images according to the data in each column. •  and  detect rotated and inverted images (including side flipping produced by scanning double-sided documents). When detection for rotations and double-sided inversions is activated, the **Template Test** interface displays a rotation column and an inversion column. Files that are detected as rotated or inverted contain a symbol in the appropriate column. •  indicates a double-sided inversion. •  and  indicate a rotation and the direction of the rotation. The images are rotated without any correction and the anchors are detected regardless of the direction in which the image was rotated. • For each documents, the anchor column displays the matching rate value corresponding to the anchoring threshold. Results can appear up or down to the anchoring threshold. • The table displays each calculated value in pre-classification and if this value is greater than the pre-classification threshold then the value is calculated for each anchor.
Right pane	Displays the document selected from the file list. It enables you to visualize the anchors found in each document. **Show detail for** lists all the anchors placed in the **HPA Template Editor**. If selecting a double-sided document, the HPA Template Test displays a **Front Page** tab and a **Back Page** tab at the top of the referenced image.

5.7 Image Analyzer

Checks the image resolution and converts images to the project resolution.

Table 5-65: Image Analyzer

Element	Description
File menu	Loads images, analyzes them, changes their resolution and closes the **Image Analyzer**. • **Load Images**: Loads selected images from a directory. • **Load Tree**: Loads all the images from a directory and subdirectories. • **Analyze**: Runs the image analysis to check the resolution and the format of each image. When you run the image analysis, an expanding bar appears to show you the evolution of the process. • **Change Resolution**: Enables you to change the resolution of the images if it differs from the project resolution. • **Close**: Closes the **Image Analyzer**.
Display menu	Five displays are available in the **Image Analyzer**: . • **General**: Default value. Displays all the images regardless of their dimension, resolution, size or format. • **Group by Dimension**: Groups the images according to their length and width. • **Group by Resolution**: Groups the images according to their screen resolution. • **Group by Bits/Pixel**: Groups the images according to their format. • **Group by Status**: Groups all the images that have been successfully analyzed.
Image Analyzer toolbar	The toolbar offers you exactly the same options as the **File** and **Display** menus.

Element	Description
Left pane	Displays the image analysis results: • The **File name** column lists the loaded images. All the images are preceded by a number. • The **Dimension column** displays the length and the width of each document. • The **Resolution** column displays the screen resolution of each document. You can modify the resolution of an image by right-clicking on an image from the **File name** list and by selecting **Change resolution** from the menu that opens from the list. An expanding bar appears while the operation processes. • The **Pixel count** column displays the size of each document. • The **Bits/pixel** column displays the format of each document. • The **Status** column indicates whether the image analysis has been completed: **OK** means that the image has been analyzed, **Error** means that the image could not be analyzed (for example if the image format is not supported). In each display, click **+** to see the detailed composition of a group. To delete an image from the **File name** list, right-click an image and select **Remove**.
Right pane	Displays the image selected in the **File Name** list.

5.7.1 Type of Resolution

This window enables you to check a new image resolution for your project and is available when you select **Change Resolution** from the **File** menu of the **Image Analyzer** window.

Table 5-66: Image Analyzer Type of Resolution

Element	Description
On X axis	Scroll and select a new *DPI* value for the X axis.
On Y axis	Scroll and select a new DPI value for the Y axis.

Element	Description
Project resolution	Select a pre-defined resolution displayed in the combo box.

5.8 Image Export

This window displays when selecting **File** > **Export the Following Images** from the **Classification Test** window. This window exports images from the **Classified** tab, the **To Confirm** tab or the **Not Classified** tab. If no images are selected, all images in the current tab will be exported.

Table 5-67: Image Export Window

Element	Description
Directory	Selects the destination directory for exporting the templates. Click  to navigate to the destination directory. If exporting images from the **To Confirm** tab or the **Not Classified** tab, images are exported directly to the selected destination directory. If exporting from the **Classified** tab, select a **Grouping**.

Element	Description
Grouping	The **Grouping** list works only when exporting images from the **Classified** tab. You can group: • **Per code**: to create subdirectories that take the name of the template codes. Each classified image is copied to the subdirectory that corresponds to its template code. • **Per template**: to create subdirectories that take the name of the template. Each classified image is copied to the subdirectory that corresponds to its template name. • **Per code then template**: to create subdirectories that take the name of the template code and then the template name. During the export, a progress window appears. Select **Cancel** to cancel the export and the images are neither exported nor deleted from the project. After the export has completed, a message displays the number of images exported and indicates if any special characters were renamed. The special characters of template names and codes are replaced by <_>. Duplicate images are renamed: `image.tif` = `image(1).tif` then `image(2).tif`.
... (browse) button	Browse and select the directory for exporting the templates.
OK button	Validates the settings, and exits the **Image Export** window.
Cancel button	Cancels any operation and closes the **Image Export** window.

5.9 New Project Wizard (Dispatcher Manager Only)

Use this wizard for analyzing images and creating classification templates automatically.

Table 5-68: New Project Wizard

Element	Description
Start	Describes the use of the **New Project Wizard** and begins creation of a new project.
Project Parameters	Sets identification parameters for the project by indicating the **Project name**, **Author**, **Company** and by adding some notes about the project you are creating. • **Maximum number of images linked to an image reference**: specifies the number of images you want to keep in the template image base, regardless of the number of images used to run automatic learning. For example, if you specify 10 images and use 100 during automatic learning, Dispatcher Manager will automatically keep in the image base the 10 images that are most representative of the template. The image base is used to run template tests and classification tests. The default value is 50. At least 10 images are necessary. It is recommended to provide 10 to 20 images to enable efficient automatic learning. • **Remove black edges from documents**: instructs Dispatcher Manager to ignore black edges present on the images during classification (the black edges are just ignored, the image files are not modified). This option is recommended as images with black edges are not typical of the template and may not be correctly classified. • **Empty project**: Enables creation of an empty project without creating templates. For example, select this option to create the templates manually because the images are already sorted and you know which templates to create.
Select a Base of Images	Displays all the images loaded from the image base.

Element	Description
Automatic Learning	The automatic learning process builds all your documents templates for you. Before creating the template, the automatic learning process analyzes the images, loads them, compares them two at a time, identifies images groups, locates center images and finally creates the template. Each time a step is completed, a green tick appears before the related step.
End	Cancels, ends the creation of the templates or closes the **New Project Wizard**. Summarizes the result of the automatic learning process. When closing the wizard, Dispatcher Manager opens and displays the new templates just created.
Left pane	Enumerates the different steps to create a recognition project.
Right pane	Summarizes the steps to complete the wizard and enables you to go to the following step or to cancel and close the **New Project Wizard**.
Cancel button	Cancels any operation and closes the **New Project Wizard** window.
Previous button	Goes to the previous step.
Next button	Goes to the following step. However, as soon as the Automatic learning process ends, it goes automatically to the next step.
Empty project button	To create templates from Dispatcher Manager.
Learn button	Runs the Automatic learning process. The **Image Analyzer** runs and automatic learning is performed.
New Project Wizard toolbar	Enables you to add images, directories, trees, delete images and learn the images.

5.10 New Template Wizard

Enables manual creation of one standard template at a time, instead of using automatic learning.

Table 5-69: New Template Wizard

Element	Description
Start	Briefly describes the **New Template Wizard** features and begins the process of creating a template.
Select a Base of Images	Displays all the images loaded from the image base. The number of the selected images for the template base is defined above the left pane.
Automatic Template Creation	• **Automatic template creation**: Before creating a template, the automatic template creation analyzes the images, loads them and identifies the different images group if any. Each time a step is completed, a green tick appears before the related step. • **Creating template**: For each step, an expanding bar appears during the **Automatic template creation**. Below this bar, a timer indicates the time the process will last and the remaining time before it ends. • **Setting the invariant zones**: During the template creation phase, the process stops at the **Setting the invariant zones** step. A reference image is displayed. This option enables you to define invariant zones on the reference image and constraints to use only these zones to identify a document. Below the image, the number of defined zones is indicated.
End	Summarizes the results of the automatic template creation process. It also displays the number of created and validated templates. Select **New template** to keep the wizard open and create another template, or select **End** to close the **New Template Wizard** window. **Cancel** closes the wizard without adding the template to the project. After closing the wizard, the new template is displayed.

Element	Description
Create template button	Runs the **Create template** process. Before the template is created, the **Image Analyzer** runs automatically.
Setting zones button	Select to define invariant zones. The **Setting the Invariant Zones** window appears. Invariant zones are useful when too few typical images are available. Invariant zones apply to very specific cases. In most cases, the system automatically defines those zones.
Cancel button	Cancels any operation and closes the **New Template Wizard** window.
Previous button	Backs up one step.
New Template Wizard toolbar	Enables you to add images, directories, trees, delete images and learn the images. • **Add images**: Loads selected images from a directory. • **Add directory**: Loads all the images from a directory. • **Add tree**: Loads all the images from a directory and subdirectories. • **Remove**: Unloads the selected images. • **Empty list**: Unloads all the images. • **Image analysis**: Runs the **Image Analyzer** to check the image format and resolution. Otherwise, go to the next step where the **Image Analyzer** will run automatically.

Related Topics

"Creating Standard Templates Manually" on page 60

"Setting Invariant Zones" on page 365

5.10.1 Setting Invariant Zones

The **Setting Invariant Zones** window displays after an image base is added in the **New Template Wizard** and presents the following options:

Table 5-70: Setting Invariant Zones Window

Element	Description
Setting Invariant Zones pane	Displays the reference image where invariant zones will be defined.

Element	Description
Setting Invariant Zones toolbar	Provides buttons to delineate the invariant zones. • **Selection** mode enables selection of the invariant zones and to move them through the reference image. • **New zone** delineates the invariant zones which are typical of the template. Those graphical zones must be wide enough to cover all the invariant zones of the document. If graphical zones are too small, the template cannot be created and a warning message appears. After setting the invariant zones, select **Create template**. • **Zoom in**, **Zoom out**, or **Default zoom** enables zooming in or out or to restore the initial display size of the image.

5.11 OCR Engine Edition

The *OCR* engines include many of the most powerful available on the market. OCR engines function through configuration files. Select an existing configuration file, edit an OCR configuration file, or create one based on an existing file.

Note: In Core Capture, only the Advanced OCR/ICR and Western OCR engines are supported.

Table 5-71: OCR Engines

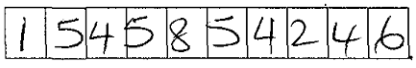
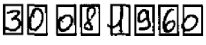
Element	Description
Advanced OCR/ICR	Carries out full page form recognition, *ICR* and multi-line OCR. It is used mainly for recognition of handprinted documents.
Barcode Recognition	Carries out *1D* and *2D* barcode recognition.
Barcode 39 Recognition	Reads barcodes of the type Code 39. This engine can read a barcode at field level (i.e., zonal recognition), in other words cannot detect the barcode in the whole document.
Basic French ICR	This engine is dedicated to the French language only. For other languages, use either General-Use OCR or Advanced OCR/ICR engines.
Basic OCR	Recognizes machine printed alphanumeric characters that have consistent, predictable shapes and fonts.

Element	Description
Check Reading	Recognizes business and personal checks, deposits slips, cash-in and cash-out documents.
Box Field	Used to remove boxes surrounding characters.
General-Use OCR	Carries out full page form recognition, OCR, multi-line and omni-font characters recognition and barcode recognition.
Modification Detection	Used in the following two cases: • To detect the presence of specific information in an empty field, for example the presence of a signature at the bottom of the document. • To detect a manual modification (handwritten notes) made to pre-printed characters such as a pre-printed address in a document.
Multi-Engine Voting	Enables combining several engines to improve recognition.
OCR/ICR Voting	Features a voting engine that is able to detect whether characters are handprinted or machine-printed to apply the appropriate recognition engine to the field.
Optical Mark Recognition	Detects whether a checkbox is selected or not.
Western OCR	Used for zonal and full text recognition. Recognizes machine printed characters.

5.11.1 Box Field

In zonal extraction, this engine is used to remove boxes that surround characters ("Box Field engine" on page 367).

Table 5-72: Box Field engine

Before running the Box Field engine	After running the Box Field engine
	1545854246
	30081966

Note: Apply this engine before running another OCR engine.

Table 5-73: Box Field Window

Element	Description
Name	The name for the new engine.
Crop Height Margin	The number of pixels to remove as measured from the inside border of a box's top and bottom lines outwards. Valid values are 0 to 10. If the boxes are skewed, have thicker lines, or have lines of variable thickness, then specifying a larger number of pixels can improve the removal of top and bottom lines.
Crop Width Margin	The number of pixels to remove as measured from the inside border of a box's left and right sides outwards. Valid values are 0 to 10. If the boxes are skewed, have thicker lines, or have lines of variable thickness, then specifying a larger number of pixels can improve the removal of right and left sides.
Vertical Bar Detection Threshold	In the current release, if the value is set to 100, then all content outside the frame is deleted. If the value is less than 100, then only the frame is removed.
Width Tolerance	The minimum size of the frame containing the content in relation to the selected area. The recommended relative size of the frame containing the content to the selected area is 30%.
Engine	The OCR engine to perform the extraction.

5.11.2 Barcode 39 Recognition

This engine can only recognize code 39 barcodes. The document must have a minimum resolution of 200 *DPI*. A lower resolution could make the barcodes appear as being stuck to the scanned document and thus rendering them unreadable by the engine. A resolution of 300 DPI is recommended for smaller-sized barcodes.

Table 5-74: Barcode 39 Recognition Window

Element	Description
Name	Type in a name for the new engine you are about to create.

Element	Description
Main parameters	**Type**: This *OCR* engine can only read Code 39 barcodes. **Minimum number of characters**: Select the number of characters in output. The default setting is 9.
OK button	Validates the settings and exits the **Barcode 39 Recognition** window.
Cancel button	Cancels any operation and exits the **Barcode 39 Recognition** window.

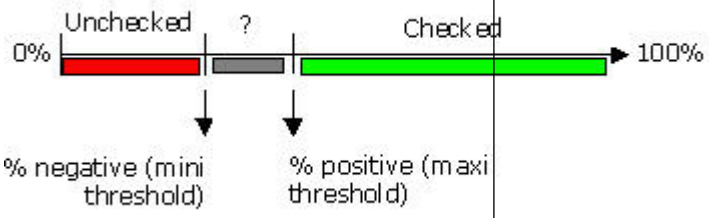
5.11.3 Optical Mark Recognition

This engine can detect whether a checkbox is selected or not. To do so, the engine calculates the number of pixels in the original image (for example, ☐) and on the finished image (☑) to detect the presence of any pixels added by selecting the checkbox (✓).

In the template, you need to place a field on each checkbox to be read (place the field at a maximum of 2 millimeters from the checkbox border). If the form has not been printed with inactinic ink, you should apply an adaptive filter.

Table 5-75: Optical Mark Recognition Window

Element	Description
Name	Type in a name for the new engine you are about to create.

Element	Description
Percentages	**Positive percentage** and **Negative percentage** values are used to fix the acceptation threshold of the engine. Recommended values are 1% for **Negative percentage** and 5% for **Positive percentage**. The engine analyzes the percentage of black pixels in the checkbox, that is, the fill-rate of the checkbox. If the percentage is greater than the **Positive percentage** value (maximum threshold) then the checkbox is checked. If the percentage is less than the **Negative percentage** value (minimum threshold) then the checkbox is not checked. If the percentage is between the maximum and minimum values, then the value ? is sent and displays in the template test. 
Option	Leave the **Without skeleton** option checked. This option is preserved for compatibility with former versions.
OK button	Validates the settings and exits the **Optical Mark Recognition** window.
Cancel button	Cancels any operation and exits the **Optical Mark Recognition** window.

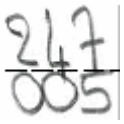
5.11.4 Basic French ICR

The Basic French ICR engine is appropriate for zonal recognition of hand printed characters using French.

Table 5-76: Basic French ICR Window

Element	Description
Name	Type in a name for the new engine to create.

Element	Description
OCR/ICR engine	Options are: • **Font**: refers to the specific font set to use. For Basic French ICR the **HandPrint** font (selected by default) is the standard learning base. `HandPrint` is also the name of the directory in the global resources that contains the learning base supplied in the standard package. The path is `\Resources\Fonts\` • **Mode**: Select **Alphabetic** or **Numeric**; alphanumeric mode does not exist. **NumericSlash** mode is not managed. **Note:** In the same directory as the `HandPrint` directory is a directory named `TypePrint` which is used to process machine printed documents. Use the Basic OCR engine for machine printed documents.
Average size	The average size is useful for fill-in form entry boxes, which is usually **Height** = *<5>* **mm**, **Width** = *<4>* **mm**. Reading is difficult for smaller sizes. The average size is used especially for **Segmentation** and **Characters association** options.
Minimum height	Specifies the minimum height that is accepted. Fro example, if using *<1>* mm, all characters less than *<1>* mm are eliminated. **Note:** For amounts do not use a minimum height of less than 1.3 mm. Zeros in certain writing are often reduced more than other characters and can be filtered if the **Minimum height** option is selected.
Segmentation	Select **Active** to automatically separate characters according to the average size defined. This option is useful when the characters are very close together as in the following example where the two zeros are joined together: . 
Raising	Select **Active** to dilate characters when the writing is very fine. Dilation makes characters thicker and improves recognition.

Element	Description
Detection of /	Select **Active** so that "/" characters are detected and replaced by a "?" in data validation. **Note:** A field value cannot contain a question mark ("?") because the question mark indicates a field in error. The question mark cannot be entered again during validation so they must be removed or replaced by another character.
Minus detection	Select **Active** so that "-" characters are detected and replaced by a "?" in data validation. **Note:** A field value cannot contain a question mark ("?") because the question mark indicates a field in error. The question mark cannot be entered again during validation so they must be removed or replaced by another character.
Characters association	Option active by default. Enables an engine to rebuild characters from connected shapes. Increase the threshold to associate connected shapes that are more spread out: In the example below, the character "5" is composed of two connected shapes.  In the next example, "247" and "005" are too close to each other. In this case, reduce the threshold to reduce overlapping: 
Other parameters	The **Words detection threshold** (in mm) is used to group characters together to build words. For example, if the threshold is *<3>* mm, the engine forms two words from two groups of characters if the two groups are separated by at least *<3>* mm: 

Element	Description
OK button	Validates the settings and exits the **Basic French ICR** window.
Cancel button	Cancels any operation and exits the **Basic French ICR** window.

5.11.5 Basic OCR

The Basic OCR engine is appropriate for zonal recognition of machine printed characters using French.

Table 5-77: Basic OCR Window

Element	Description
Name	Type in a name for the new engine to create.
OCR/ICR engine	**Font**: refers to the specific font set to use. For Basic OCR the **TypePrint** font (selected by default) is the standard learning base. **Mode**: Select the appropriate character type. **Note:** In the same directory as the `TypePrint` directory is a directory named `Handprint` which is used to process hand printed documents. Use the Basic French ICR engine for hand printed documents.
Machine printed characters average size	The average size is used in conjunction with the **Segmentation**, **Association** and **Despeckle** options. The **Predetermined** option **Predetermined**: Specifies the expected **Height** and **Width** for characters. This option is suitable for most applications. **Dynamically Measured**: Specify the minimum and the maximum height for the character to be read.
Association	Option active by default. Enables an engine to rebuild characters from connected shapes. For example, in the next image, the character “5” is rebuilt using two connected shapes: 

Element	Description
Despeckle	When Active, sets a range for accepted character size relative to the average character size. The minimum and maximum height is set as a factor of the average character height, and the width is set as a fixed limit. Characters which do not match the **Despeckle** limits are eliminated. Using the default values as an example, image components are eliminated if their height is less than 0.5 times the average height of characters (**AverH**) or more than 1.5 times the average height and if their width (**MinW**) is less than 0.2 mm.
Segmentation	Separates characters automatically according to the defined average size. This option is useful when characters are very close together as in the following example: 
Italic font	Straightens italic characters before recognition.
Inter-word	**Words detection**: Defines words by deleting spaces between characters. Select the **Words detection** option then specify the values **Min space** and **Max space**. These two values are then used to calculate the threshold dynamically. **Alphabetic/Numeric vote**: This option enables identification of characters that can be interpreted as being a letter or a number (examples: I and 1, 0 and O, 8 and B, Z and 2). The engine uses all the characters present to "vote". For example: • If the characters are letters, then the form will be voted as a letter (PAR1S = PARIS). • If the characters are numbers, then the form will be voted as a number (Z584 = 2584).

Element	Description
Pre-locate lines	Enable this option when fields are very close together, either horizontally or vertically, and if there is a risk of one field overlapping another, as in the following example:  The engine automatically determines characters and then eliminates all stray marks that are not part of the component shape. In most situations it is appropriate to leave this option active.
OK button	Validates the settings and exits the **Basic OCR** window.
Cancel button	Cancels any changes and exits the **Basic OCR** window.

5.11.6 Modification Detection

The Modification Detection engine is used in the following two cases:

- To detect the presence of specific information in an empty field, for example the presence of a signature at the bottom of the document.
- To detect a manual modification (handwritten notes) made to pre-printed characters such as a pre-printed address in a document.

For more information, see "Modification Detection for Handwritten Notes or Signatures" on page 139.

Table 5-78: Modification Detection Window

Element	Description
Name	Type in a name for the new engine you are about to create.

Element	Description
Large connected shapes detection criteria	The criteria is effective for signatures and hand written notes. • To detect a modification in a machine printed zone, define only the **Minimum height**. It should be slightly higher than the size of pre-printed characters in the machine printed zone. If the handwritten notes found in the zone are greater in height than the field placed on the pre-printed characters, then the engine considers that handwritten notes have been added onto the pre-printed characters. • To detect a signature, a height or length of *<10>* mm is advised in most cases. If the shape found in the zone is of a greater size than either the **Minimum height** or the **Minimum width**, then the engine indicates the presence of a signature.
Minimum connected shapes required	The engine can determine the number of connected shapes that make up the signature or note. The default of *<1>* is recommended for nearly all situations.
OK button	Validates the settings and exits the **Modification Detection** window.
Cancel button	Cancels any operation and exits the **Modification Detection** window.

5.11.7 Multi-Engine Voting

Multi-engine voting improves recognition by allowing combination of several engine configuration files from the same or different engines. A voting mechanism is applied to keep the results with highest confidence level. Voting requires more processing time, so combine engines that are similar in terms of performance (e.g., do not combine a slow engine with a fast engine or an accurate engine with an engine that reads less accurately).

Table 5-79: Multi-Engine Voting Window

Element	Description
Name	Type in a name for the new engine you are about to create.

Element	Description
Voting method	There are different voting types that will provide various results: • **Pessimistic vote**: This method compares all of the lowest scores from all configuration files, and selects the best return. For example, if the lowest score is 41% for A is and 47% for B, then B is selected. • **Optimistic vote**: This method compares all of the highest scores from all configuration files, and selects the best return. For example, if the , the highest score is 52% for A and 48% for B, the selected character is A. • **Average score vote**: This method compares the average scores from all configuration files and selects the character with the highest average score. For example, if the average score is 46.5% for A and 47.5% for B, the selected character is B. • **Global score vote**: This method calculates a global score from all the results for each engine configuration. The global score is the average of all the scores. The result retained will be the best global score. This calculation mode applies even if the number of characters differs from one engine to another. In fact, the vote is done at the level of the result and not at the level of the character.
Takes into account segmentation errors	Enables recognition of segmentation errors and taking them into account during voting. This option applies to **Pessimistic**, **Optimistic** and **Average score** votes and not to Easy Basic Control and global score votes. For examples of segmentation errors and information on the weighting coefficients, see "Understanding Segmentation Errors" on page 113 and"Defining Multi-Engine Voting Recognition" on page 114.

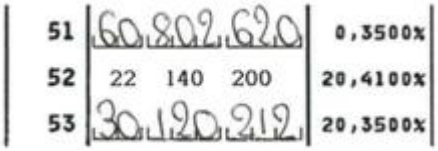
Element	Description
Select engines	Click the icons below the engine table to add, delete, or change the order of engines in the list. When adding engines, the **Define an Engine** window enables selection of the engine and filter to use, as well as the engine weight coefficient. The coefficients apply more or less relative weight to the specific engine compared to the other engines used for multi voting. More information about engines and weighting coefficients is available from "Defining Multi-Engine Voting Recognition" on page 114.
OK button	Validates the settings and exits the **Multi-Engine Voting** window.
Cancel button	Cancels any operation and exits the **Multi-Engine Voting** window.

5.11.8 OCR/ICR Voting

OCR/ICR Voting detects whether characters are hand printed or machine printed using both *ICR* and *OCR* reading. For information on setting up this engine, see "Recognition Types Supported by Recognition Engines" on page 429.

Table 5-80: OCR/ICR Voting Window

Element	Description
Name	Type in a name for the new engine to create.
Engine 1 and **Engine 2**	Select an ICR engine for **Engine 1** to detect hand printed characters. Select an OCR engine for **Engine 2** to recognize machine printed characters. Type a confidence threshold in the field to the right (between *<1>* and *<100>*%).

Element	Description
Hand/Type-printed discrimination criteria	To discriminate between hand printed and machine printed characters, the voter uses **Max Type Height** and **Max Type Width** to determine the presence of machine printed characters, since machine printed characters fit a limited profile in terms of size. Also, the **% minimum of machine printed characters** is used to determine which engine to use. For example, consider a field that returns 40% of characters with a height or length that is less than or equal to **Max Type Height** and **Max Type Width**. If the **% minimum of machine printed characters** is set at 75%, this result does not meet the criteria for machine printed characters, and the voter would apply **Engine 1** for hand printed characters..   In the hand printed form, line 52 is pre-filled with machine printed characters whereas lines 51 and 53 are handwritten. At the template level, the voter is used to determine whether the line contains machine printed or handwritten characters. An adaptive filter is applied to delete pre-filling. The voter compares the size of the characters in lines 51, 52 and 53 with the thresholds set in **Max Type Height** and **Max Type Width**. Since character size in lines 51 and 53 is greater than the thresholds, the voter considers that these characters are handwritten and it applies the handwritten recognition engine, **Engine 1**. Here, the **% minimum of machine printed characters** option can be set to *<100>* since there is no risk to have mixed machine

Element	Description
	printed or handwritten characters in the same line.
Browse (...) button	Browse and select an OCR engine `.reco` file.
OK button	Validates the settings and exits the **OCR/ICR Voting** window.
Cancel button	Cancels any operation and exits the **OCR/ICR Voting** window.

5.11.9 General-Use OCR

General-Use OCR can carry out full page form recognition, *OCR*, multi-line and omni-font characters recognition and barcode recognition, and over 100 languages are available. This engine requires a license.

General-Use OCR settings support two recognition types:

- Character recognition
- Barcode recognition

Table 5-81: General-Use OCR Window

Element	Description
Name	Type in a name for the new engine you are about to create.
Recognition mode	Displays the General-Use OCR settings that support two recognition types: • **Character** mode • **Barcode** mode The selection changes the nature of the settings available in the window.

5.11.9.1 Customization of the Character Type

When **Customized** is the selected **Character type**, click **Customize** in the **General-Use OCR** window to select a specific set of characters to take into account when recognition is performed.

Table 5-82: Customization of the Character Type Window

Element	Description
Custom characters set	Lists the characters to take into account when recognition is performed. Select the characters to be recognized (all other characters will be ignored).
OK button	Validates the selection and exits the **Customization of the Character Type** window.
Cancel button	Cancels any operation and exits the **Customization of the Character Type** window.
All button	Selects all the characters in the list.
Mixed Case button	Selects both upper and lowercase characters.
Upper Case button	Selects uppercase characters.
Lower Case button	Selects lowercase characters.
Digit button	Selects digit characters, that is, all the numbers from 0 to 9.
None button	Clears all selections.
Invert button	Reverses the selection, that is, selects all the checkboxes that were clear and clears the selected ones.
Additional characters	Enables entering non-ASCII characters, for example a specific set of Asian characters. Enter as many characters as you want without separator.

5.11.9.2 Character Recognition using General-Use OCR

Select **Character** as the **Recognition mode** for these parameters to be available.

Table 5-83: General-Use OCR Window

Element	Description
Recognition parameters	

Element	Description
Text type	Select a **Text type** from the listbox: • **Machine printed** • **Precise machine printed** • **Dot/matrix (24 points)** • **Dot/matrix (9 points)** • **Asian**: displays Asian languages in the **Languages** list. For more information on recognition types, see "Recognition Types Supported by Recognition Engines" on page 429.
Character type	Select a **Character type** from the listbox. The **Character type** only displays the set of characters corresponding to the selected **Recognition languages** to improve accuracy. For Asian languages, the characters size should be smaller than 30x30 pixels for the recognition to be as much accurate as possible.
Case type	Specifies the case expected for the characters: **Upper**, **Mixed**, or **Lower**. **Note: Case type** is only available with the some character types.
Engine mode	**Engine mode** affects the quality of the output values and the speed at which results are obtained. • **Accurate** to provide the best results, but takes more time. • **Balanced**, which is faster than **Accurate** and more accurate than **Fast**. • **Fast** gives up some accuracy in favor of quicker results. Default value is **Accurate**.
Enable recognition of rotated image	Enables correct interpretation of images that are rotated before recognition. **Note:** This option is only available for the **Machine printed**, **Precise Machine printed** and **Dot /matrix (24 points)** text types. The image output is not modified.

Element	Description
Customize button	Select the **Customized** option for **Character type** and click **Customize** to open the "Customization of the Character Type" on page 380 window for selecting characters specific to the recognition language. With this selection, accentuated letters for languages selected in the **Recognition languages** list are ignored.
Language Parameters	
Languages	The engine can process documents corresponding to one or more languages at a time. By default, a subset of languages appears. If the language is not in the list, select **Show all installed languages** to display the whole list of languages.
Recognition languages	Each language owns a specific set of characters. The engine only returns characters that belong to the selected recognition language. One or more languages can be selected. The longer the list of recognition languages, the less confident and competitive the output values. If a character does not belong to the selected recognition language, the engine returns either a character with similar shape or a ? (unrecognized character). **Note:** Chinese Simplified, Chinese Traditional, Japanese and Korean cannot be selected with any other language.
Show all installed languages	Displays the whole list of languages supported by General-Use OCR.
OK button	Accepts any changes and closes the **General-Use OCR** window.
Cancel button	Closes the **General-Use OCR** window without saving any settings.

5.11.9.3 Barcode Recognition using General-Use OCR

Select **Barcode** as the **Recognition mode** for these parameters to be available.

Table 5-84: General-Use OCR window

Element	Description
Barcode parameters	Specify the settings for barcode recognition by selecting the barcodes from the **Barcodes** list.
Barcodes	Select the barcode to use for recognition and click **Add** to use this recognition type. More than one barcode type can be added to the **Recognition barcodes**.General-Use OCR can detect several barcodes of different types in a document.
Recognition barcodes	At least one recognition barcode must be selected, although a number of barcodes can be selected. Some barcodes are incompatible. When selecting incompatible barcodes, an error message appears. The following incompatibilities are known: • Postnet code is incompatible with all other barcodes. • Code 128 is incompatible with *UCC* and *UPC-A*. • *EAN* 8/13 is incompatible with UPC-A. • Code 39 is incompatible with Code 39 full ASCII mode, Code 39 with check digit control and transmit and Code 39 with start-stop char transmit.
OK button	Accepts any changes and closes the **General-Use OCR** window.
Cancel button	Closes the **General-Use OCR** window without saving any settings.

5.11.10 Advanced OCR/ICR

The Advanced OCR/ICR engine can carry out full page and zonal recognition, *ICR* and multi-line *OCR*, although it is used mainly for recognition of handprinted documents. It supports many languages. Italic fonts, bold, script and underlined characters are not supported.

The Advanced OCR/ICR engine requires a license to work in development and production environments. Licensing with two processing types is available, with the possibility of having both at the same time:

- Zonal: limited to 25, 50, 100 or 250 characters/second

- Full page: 360, 450, 600, 900, 1800 and 3600 documents per hour

The Advanced OCR/ICR engine has two modes that you can select in the **Recognition mode** field:

- **Character field recognition**
- **Handprinted text detection**

Table 5-85: Advanced OCR/ICR Window: Character field recognition

Element	Description
Name	Type in a name for the new engine you are about to create.
Recognition mode	Select one of the following modes: • **Character field recognition** • **Handprinted text detection**
Country	Select the countries to specify the country/language-specific characteristics of the documents to be processed. The **Syntax**, **Font** and **Reader** options vary depending on the selected countries. Futhermore, specifying a country enables the engine to more effectively recognize that country's accented characters. For better performance, you should only select the countries of the documents are to be processed. The first country is the principal country and is used to define the some settings, such as handwriting recognition. **Notes** • For the AEG reader, some countries have different code pages and therefore are incompatible. • The **International** value is valid only for backward compatibility. • The Intelligent Capture Asian Language Add-on enables the following Asian languages for Advanced OCR/ICR: – Chinese (Simplified) – Chinese (Traditional) – Chinese (Traditional, Hong Kong) – Japanese – Korean – Thai

Element	Description
Orientation	Select the orientation of text recognition, which also enables the extraction of separate vertical or reverse lines.
Reader	Select an engine from the **Reader** list box. • The Voter engine combines results of the Advanced OCR/ICR and AEG engines and validates the results using a voting system. • The **Voter** option is only available when both Advanced OCR/ICR and AEG readers are available. This list is filtered according to **Country**, **Syntax** and **Font** settings.
Syntax	Select the expected character type in a field during recognition. This list is filtered according to the **Country** settings.
Font	Search for the classifier that corresponds to the **Font** and the syntax, from the following: • **CMC7** • **E13B** • **Farrington 7B** • **Fixed** • **Hand Printed** • **Machine Printed** • **OCR A** • **OCR B** • **Unknown** (default value) If several classifiers are defined, respect the order of classifiers. If two classifiers are specified, the engine will deliver the most likely of the two generated replies. **Notes** • To recognize both handprinted and machine-printed writing simultaneously: select a classifier that can process machine-printed writing and another classifier that can process handprinted writing and then select **Unknown** in the **Font** listbox.

Element	Description
Machine printed parameters	The available options are: • **Known character height**: Select this option and use the controls to define the expected height of characters. The character height is defined by the height of capital letters, and is generally in the range of 2.5 to 3.5 mm (1.8 to 4 mm is allowed) for machine printed characters.  To measure character height, in Recognition Designer, select the **Index View**, hold down the right mouse button and draw a bounding box around the character to measure. Keep the mouse button down and read the height (**H**) and length (**L**) (in mm) from the status bar. • **Character pitch**: The pitch is the number of characters per unit length. Select this option and use the controls to define the expected number of machine printed characters per unit length. You can set the pitch to **Variable** (default value), **Unknown**, or **Fixed**.
Hand printed parameters	The available options are: • **Character height**: The character height is defined by the height of capital letters. Character height should be 5.5 mm for hand printed characters (3 to 7 mm is authorized). • **Character pitch**: The pitch is the number of characters per unit length. Select this option and use the controls to define the expected number of hand printed characters per unit length. The pitch value is in millimeters and has a valid range of 2 to 10, with a default value of 5. You can set the pitch to **Variable** (default value), **Unknown**, or **Fixed**.
Advanced parameters	
Logical Context	Makes an attempt to distinguish between number and letter alternatives according to the text environment. For example, distinguishing between the letter Z and the number 2.

Element	Description
Trigram Mode	Trigrams are sequences of three consecutive characters. A trigram table contains the occurrence probabilities of all trigrams for the countries and readers that are listed within.
Formal Context	Specify the characters authorized in a field with the help of a regular expression. This option controls character segmentation and the selection of recognition alternatives. For more information, see "Regular Expressions" on page 96.
Classifiers	Classifiers are required. They are used during reading for decision-making. For example, the decision that a character belongs to a class depends on the selected classifiers.

Table 5-86: Advanced OCR/ICR Window: Handprinted text detection

Element	Description
Name	Type in a name for the new engine you are about to create.
Recognition mode	Select one of the following modes: • **Character field recognition** • **Handprinted text detection**
Positive result	Enter a value to override the default return value 1, if handprinted text is detected in the selected zone.
Negative result	Enter a value to override the default return value 0, if the selected zone is empty or contains machine-printed characters.

5.11.10.1 Select a Classifier

This window is available when you select the **List** button in the **Classifiers** of the **Advanced OCR/ICR** window.

Table 5-87: Select a Classifier Window

Element	Description
Classifier	List of available classifiers.
Country	Countries vary depending on the selected classifier.
Description	Information on the classifier type.

Element	Description
Version	Version number of the Advanced OCR/ICR revision on classifiers.
OK button	Validates your selection and exits the **Select a Classifier** window.
Cancel button	Cancels any operation and exits the **Select a Classifier** window.

5.11.11 Check Reading

Check Reading engine is used for automatic reading of business and personal checks, deposits slips, cash-in and cash-out documents. It can read hand printed, handwritten and machine printed documents and performs entire check recognition efficiently. It has been successfully used for check and remittance processing.

Check Reading engine requires a license to work in development and production environments. Check Reading licenses need to be registered using the Check Reading license management system.

In this window, **Data type** settings depend on the selected language: **English (United-States)** (default value) or **French**.

Table 5-88: Check Reading Window

Element	Description
Name	Type in a name for the new engine to create.
Language	There are only two languages available with this engine: **English (United-States)** or **French**.
Data type	The **Data type** settings depend on the selected language (**English** has more settings available).
CAR acceptance threshold	By default, this option appears when you open the **Check Reading** window. It is only available when **English (United-States)** and **Amount** are selected.

5.11.11.1 English (United-States)

These settings are available when selecting **English (United-States)** as **Language**.

Table 5-89: Check Reading English (US) Data Type Setting

Element	Description
Amount	Displays the amount read in a check without a decimal separator and currency symbol. For example, if the recognition engine reads "$61.07" it returns the following value: "6107". If the recognition engine reads an integer amount such as "$123" it automatically adds two zeros to replace the decimals. The value returned is then "12300". Check Reading is able to read the courtesy amount (amount in figures) and the legal amount (amount in letters) and to compare both values to ensure a better recognition quality. The **CAR acceptance threshold** enables setting recognition of legal and courtesy amounts. The option **Define CAR acceptance threshold** is cleared by default. When the option is cleared, a 84% default acceptance threshold is applied with the following default behavior: Check Reading first tries to recognize the courtesy amount. If the confidence threshold is strictly higher than 84%, the engine does not try to recognize the legal amount. If the confidence threshold is lower than 84%, the engine tries to recognize the legal amount and compares both the legal and the courtesy amounts. Selecting **Define CAR acceptance threshold** enables you to set a *CAR* acceptance threshold to a value between 0% - 100%. If you set the threshold to 0%, the engine does not try to recognize the legal amount and returns the courtesy amount. If you set the threshold to 100%, the engine tries to recognize the legal amount and to compare it to the courtesy amount. The *CAR/LAR* values and the confidence values are available by scripting through the object `DpAdditionalInformation` for example to populate the amount with the *LAR* value or detect fraudulent checks. For help on using `DpAdditionalInformation`, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

Element	Description
MICR/CMC7 Codeline	The *CMC-7*font is a special *MICR* barcode font that is used to print characters for magnetic recognition and optical character recognition systems.
Signature detection	Check Reading detects signatures on checks. If Check Reading detects one or more signatures, the value returned is “1” or “2”, otherwise it is “0”.
Payee line	By default, the payee list is empty. You must create a payee list and then import it into Check Reading. We recommend importing the list from a TXT file. These options are available: • Add (**+**), delete (**-**) or edit (**ab**) a payee. If you try to add a payee that already exists in the **Payee list**, a warning message appears and the action is aborted. We recommend not changing the name of the payee to include it in the list, otherwise the engine is not able to recognize it. • Type in the **Payee list** the name of the payee you are looking for (space characters are not allowed). It is automatically highlighted in the payee list. • Import or export a payee list. If you import a list including payees with the same name, a warning message appears and only one of the payee is kept. The number of payees is displayed at the bottom right of the **Payee list**. Before closing the **Check Reading** window, check if the payee list is not too long by clicking either **Check payee list** (manual action) or **OK** (automatic action). For Unicode support, the Project Designer must select the encoding format of the TXT file to load when importing data in Recognition Designer or Free Form Designer: • Autodetect • ANSI • UTF-7 • UTF-8 • Unicode (UTF-16 Little-Endian) • Big-Endian (UTF-16 Big-Endian)

Element	Description
Check number	Returns an alphanumeric value. For example: “1046” or “456A”.
Date	The value returned respects the following structure: mm/dd/yyyy. For example, if the separator used in the check is “01-23-07”, the value returned is “01/23/2007”. Select a date interval. Options are: • **Fixed** date: Select the minimum or maximum date limit. You must always keep within a date interval of 4 months. By default, the minimum date is 3 months and one day prior to the current date and the maximum date is one month after the current date. If you modify the minimum date, the maximum date automatically keeps a date interval of 4 months. Fixed dates are permanent dates: they are not automatically updated depending on the current date. • **Floating** date: Select the minimum or maximum date limit. You must always keep within a date interval of 4 months. By default, the minimum date is 3 months and one day prior to the current date and the maximum date is one month after the current date. If you modify the minimum date, the maximum date automatically keeps a date interval of 4 months. Floating dates are automatically updated depending on the current date.

5.11.11.2 French

This topic is available when you select **French** as **Language**.

Table 5-90: Check Reading Data Type Setting French

Element	Description
Amount	Displays the amount read in the check without a decimal separator and currency symbol. French language does not recognize US currency. For example, if the recognition engine reads “61.07” it returns the following value: “6107”. If the recognition engine reads an integer amount such as “123” it automatically adds two zeros to replace the decimals. The value returned is then “12300”.

Element	Description
MICR/CMC7 Codeline	The *CMC-7*font is a special *MICR* barcode font that is used to print characters for magnetic recognition and optical character recognition systems.

5.11.12 Barcode Recognition

Barcode Recognition carries out barcode recognition. Barcode Recognition settings supports two barcode types:

- 1D barcode parameters
- 2D barcode parameters

Table 5-91: Barcode Recognition Window

Element	Description
Name	Type in a name for the new engine you are about to create.
Barcode type	Displays the Barcode Recognition settings that support two barcode types: 1D and 2D barcodes.

5.11.12.1 1D Barcode Parameters

Table 5-92: Barcode Recognition 1D Barcode Type

Element	Description
Barcodes	Select the barcode to use for recognition. Multiselection is not allowed.
Recognition barcodes	Barcodes used for recognition. At least one recognition barcode must be selected. Some barcodes are incompatible between them. For help on incompatible barcodes see "1D and 2D Barcodes with Barcode Recognition" on page 151.
Barcode detection mode	Frequency of barcodes detection. Move the cursor on the progress bar depending on whether you want greater accuracy or speed: • **Accurate**: Recognition is slower, and fewer barcodes are detected, but results are more accurate. • **Fast**: Recognition is faster, and more barcodes are detected, but results are generally less accurate.

Element	Description
Barcode orientation	Select the barcode orientation to recognize: **Horizontal**, **Vertical**, **Horizontal and vertical** and **All**. Default value is **Horizontal**. If you select **All**, the engine reads the barcodes rotated by 45, 90, 135, 180, 225, 270, 315 and 360C. Even if the engine detects the rotation of barcodes, Recognition Designer does not detect the rotation of the document.
Append checksum	Checks the integrity of the barcode. A checksum character is appended at the end of the output value. This option is available for the following barcodes: Code 128, Code 93, Code 93 extended, Airline 2 of 5, Intelligent Mail, PostNet, Royal Post. Checksum characters are always appended at the end of the following barcodes: *EAN* 13, EAN 8, *UPC-A*, *UPC-E*, *UCC/EAN* 128. **Note:** If a checksum error is detected (meaning that the barcode output value does not match with the checksum character), the confidence threshold is set to 50. For more information on setting confidence values, see "Recognition Engine Confidence Threshold" on page 110 .
Include control characters	Adds a control character at the beginning and the end of the barcode. This option is only available for Codabar.

Element	Description
Decode type	This option is only available for the Australian Post barcode. The Australian Post barcode supports different formats, including custom formats. Custom formats are composed of encoding data. To decode custom formats, use the following decode types: • **Bar states**: Custom formats are not decoded. The engine returns the bars status: – **A**: ascending bars – **D**: descending bars – **F**: full bars – **T**: tracking bars • **Table N**: Custom formats are decoded using numeric values. • **Table C**: Custom formats are decoded using alphanumeric values. • **None**: Custom formats are ignored. Default value is **Bar states**.
Reverse video image	This is an invert image filter. It creates negative versions of images, replacing white pixels with black and black pixels with white. Select **Auto detect** to automatically invert the image if needed. However, it slows the recognition speed.
Maximum barcode per image	Select the number of barcodes per image. Default value is set to *<1>*.
Output value separator	If the zone contains several barcodes, the output value returned is concatenated. This option enables you to create a separator between each barcode output value. Define the separator (only one character) in the **Output value separator** field.
OK button	Accepts any changes and closes the **Barcode Recognition** window.
Cancel button	Closes the **Barcode Recognition** window without saving any settings.

5.11.12.2 2D Barcode Parameters

Table 5-93: Barcode Recognition 2D Barcode Type

Element	Description
Export as binary data	Select this option to export the output value in a hexadecimal data format. By default this option is not selected.
Reverse video image	This is an invert image filter. It creates negative versions of images, replacing white pixels with black and black pixels with white. Select **Auto detect** to automatically invert the image if needed. However, it slows the recognition speed.
Maximum barcode per image	Select the number of barcodes per image. Default value is set to *<1>*.
OK button	Accepts any changes and closes the **Barcode Recognition** window.
Cancel button	Closes the **Barcode Recognition** window without saving any settings.

5.11.13 Western OCR

The Western OCR engine enables zonal and full text recognition. It includes a modifiable recognition threshold and indication of character details (character position and associated recognition score). It can automatically detect and identify the language used in documents (English, German, Danish, Spanish, Finish, French, Dutch, Italian, Norwegian, Portuguese). In addition, an included English lexicon of 30,000 words can improve English word recognition.

Western OCR reads only the characters indicated shown in Figure 5-1.

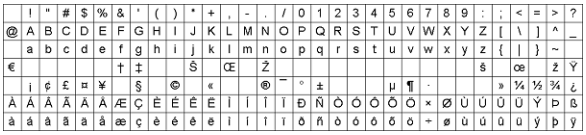


Figure 5-1: Western OCR

Table 5-94: Western OCR window

Element	Description
Name	Type in a name for the new engine you are about to create.

Element	Description
Mode	Select a mode from the list box. The difference between **Numeric** and **Amount** is that for **Amount** the following special characters are recognized: “-”, “+”, “.”, and“,”.
Case	Upper and lower cases are available only in **Alphanumeric** mode. The case is automatically **Indifferent** in **Amount**, **Numeric**, **Customized** and **All characters** modes.
Rotation	Rotates the image before processing. Select **Automatic detection** to detect the orientation of a page. Also select the rotation options and clear the **Text box reading** option in **Project Options**. For information on the **Text box reading** option, see “Text Matching Tab” on page 281. **Caution** Do not select this option when using Western OCR with free form recognition (both index and table fields); images rotated by Western OCR cannot be processed by the free form engine.
Tradeoff	Balances performance and accuracy. The default setting is **Fast**. This setting may result in reduced accuracy results but higher speed or performance. Choose the **Fastest** setting for best accuracy results. The **Very Accurate** setting does not provide any significant improvement in accuracy or performance.
Filter spaces	Removes spaces between words.
Speckle erasing	Select this option and specify, in pixels, the horizontal (**Lmin**) and vertical (**Hmin**) distance from a character to eliminate noise.
Advanced button	Opens the **Advanced Options** window.
Customize button	Opens the **Customization of the Mode** window.
OK button	Validates the settings and exits the **Western OCR** window.
Cancel button	Cancels any operation and exits the **Western OCR** window.

5.11.13.1 Western OCR Advanced Options

The options listed here enable additional processing information to be applied to Western OCR.

Table 5-95: Western OCR Advanced Options window

Element	Description
Reverse video image	Applies reverse video mode for the whole page before performing OCR of this page.
Deskew image	Straightens oblique or slanted images before processing.
Fine removal of shading patterns	Removes noise before processing.
Delete the lines	Deletes lines and underlines before processing.
Filter shading	Deletes shadows before processing.
Reverse video zones	Tries to recognize a snippet both with reverse and non-reverse video in it. By default, this option is set true for full text Western OCR global resources and false for non full Western OCR global resources.
Section the image and return the results in read order	Detects sections of text in a page in an attempt to reproduce read order (for example, newspapers are read down column before reading the next column). Sectioning can improve accuracy when there are lines of text that align between separate columns or sections.
Merge incomplete or partially recognized characters	Merges connected shapes to rebuild characters and separates merged characters.
Delete rejected lines	Deletes rejected lines after processing.
Reject badly recognized characters	Rejects badly recognized characters after processing.
Use the lexicon	Uses a 30,000 word lexicon to improve English word recognition. The lexicon is pre-activated in all modes. Deactivate **Alphanumeric** mode so that digits are not replaced with letters in alphanumeric fields.
OK button	Validates the options and exits the **Advanced Options** window.
Cancel button	Cancels any operation and exits the **Advanced Options** window.
Default button	Resets the options to their default values.

5.11.13.2 Customization of the Mode

This window enables selection of the characters to take into account. In the **Customization of the Mode** window, only those characters that the engine can recognize are selected; those which are not selected are not recognized.

The **Customization of the Mode** window presents the following elements:

Table 5-96: Western OCR Customization of the Mode window

Element	Description
Characters to be used	Lists the characters to take into account. The characters that can be recognized are selected and those which are not selected are not recognized.
OK button	Validates the options and exits the **Customization of the Mode** window.
Cancel button	Cancels any operation and exits the **Customization of the Mode** window.
All button	Selects all the characters in the list.
Mixed Case button	Selects both upper and lowercase characters.
Upper Case button	Selects uppercase characters.
Lower Case button	Selects lowercase characters.
Digit button	Selects digit characters, that is, all the numbers from 0 to 9.
None button	Clears all the selected checkboxes.
Invert button	When opening this window, the characters to be recognized are already selected. The **Invert** button reverses the selection.

5.12 Auto-Learning Supervisor-Settings

This window presents options for scheduling the learning process. It is displayed by right-clicking on the **Supervisor** icon in the system tray and selecting **Settings**. Additional options for *PAL* must be set from the **Project Options** window.

Table 5-97: Supervisor Settings Window

Element	Description
Days	The days on which automatic learning should run against those documents the Collector has placed in **Documents Storage**. Automatic learning can be performed on any selected day, or on every day of the week. To automatically run automatic learning, at least one day must be selected. The default is no days selected.
Start Time:	The hour and minute of the selected day when automatic learning should run against those documents the Collector has placed in **Documents Storage**. This setting can be edited and must be a valid time value in the format formatted as *<00>*h*<00>*min to *<23>*h*<59>*min. The value cannot be empty.
Dispatcher Projects	The setup paths of all recognition projects that **Supervisor** will manage. The **Supervisor** will manage a maximum of 255 projects. Use the **+** and **-** buttons to add or remove projects from the list. The arrow buttons to the right of the projects pane enable reordering the project list. Both the **Dispatcher Project Path** and the **Collector Location** display for each project.
Add	Add a new recognition project to the **Dispatcher Projects** list. Clicking this button displays a browse window for selecting the project to add.
Delete	Deletes the selected project from the **Dispatcher Projects** list.
OK button	Saves the **Supervisor** settings and closes the window.
Cancel button	Closes the settings window and discards any changes.

5.13 Project Analyzer

Each classification template has a dedicated directory that contains a `classifier.tif` (the reference image) and a `classifier.dpm` (the classification anchors).

The **Project Analyzer** checks the integrity of the templates and ensures that these two files are present in the template directories. If one of those templates is missing the project will be considered inconsistent.

Table 5-98: Project Analyzer Window

Element	Description
Project	Click the **Browse** button and select the project file to analyze.
Analyze project	This pane displays two messages: • If the project is consistent a confirmation message appears, meaning that the `classifier.tif` and the `classifier.dpm` are present in the template directory. • If the project is not consistent, the pane displays the list of inconsistent templates with a warning message. This might also occur if the reference image or the classification anchors are missing.

5.14 Project Update Wizard

The **Project Update Wizard** creates a new standard template in an existing project.

Table 5-99: Project Update Wizard

Element	Description
Start	• Left pane: Represents a tree structure enumerating the different steps to create a standard template in an existing project. Each step is preceded by a colored square. When the colored square is dimmed it means that a step has been successfully completed. • Right pane: Offers a brief presentation of the **Project Update Wizard** features. Also, it enables you to go to the following step or to cancel and close the **Project Update Wizard** window.

Element	Description
Select a Base of Images	• **Select a base of images**: Displays all the images loaded from the image base. When you select an image in this pane, a thumbnail of this image is displayed in the right pane. The number of the selected images for the template base is defined above the left pane. • **Delete source images from their directory when they are included in Recognition Designer**: Deletes the images used to update the project.
Select Settings	• **Template creation method**: Select the classification method to be used for template creation. – **Textual Templates**: These templates are based on a full-text analysis of the collected data. They provide good accuracy while keeping the project size relatively small; you can also update them more easily than graphical templates. – **Graphical Templates**: These templates are based on a graphical analysis of the collected data. They enable high-speed processing and are best used when images are reasonably stable. – **Textual and Graphical Templates**: These templates are based on both full-text and graphical analyses of the collected data. Although the project gains better accuracy by using both types of templates, the project can grow in size and performance might be slower. • **OCR engine**: Displays the OCR engine used for table fields detection. **Note:** You must set the OCR engine in the **Project Options** > **Recognition** tab. • **Minimum number of documents required to create a template**: The minimum number of documents that must be analyzed to create templates. The maximum is 999.

Element	Description
Automatic Update	The automatic update creates automatically the templates and indicates the total time the process will last and the remaining time before it ends. Before creating the template, the automatic update process performs the following tasks: • Generates OCR files. • Analyzes the images, loads them, compares them two at a time, identifies images groups, locates center images, merges the images with the existing templates and finally creates the template. Each time a step is completed, a green checkmark appears before the related step.
End	Summarizes the result of the automatic update process. Select **End** to close the **Project Update Wizard** or **Cancel** to close without updating the project. As soon as you close the wizard, the new templates are displayed.
Cancel button	Cancels any operation and closes the **Project Update Wizard** window.
Previous button	Enables you to go to the previous step.
Update button	Runs the project update process. Before the project update process starts, the **Image Analyzer** runs automatically. If the resolution of the image is not homogeneous then a warning message appears asking you if you want to select a unique resolution. Select **Yes** otherwise the **Image Analyzer** will automatically convert the images to the project resolution.
Project Update Wizard toolbar	Provides toolbar buttons for performing common functions.

5.15 Template Import Wizard

Assists in importing automatically a set of new templates from another recognition project.

Table 5-100: Template Import Wizard

Element	Description
Start	• Left pane: The left pane represents a tree structure enumerating the different steps to import a set of templates from another project. Each step is preceded by a colored square. When the colored square is dimmed it means that a step has been successfully completed. • Right pane: The right pane offers you a brief presentation of the **Template Import Wizard** features. Also, it enables you to go to the following step or to cancel and close the **Template Import Wizard** window.

Element	Description
Choice of Templates to Import	• **Import Project** window: Enables you to select the source project. • **Project**: This box indicates the path and the source project name of the selected project. If you want to select another project, click the **Browse** button. • **Directory**: Indicates the root directory where the selected templates will be imported. • **Displays sub-directories content**: When selecting a source project, this option is checked by default, and all the project templates are displayed and selected for import to the root directory. If you want to import templates from different subdirectories, then check the **Display subdirectories content** option. Browse through the list of templates and clear the selection boxes for those templates you do not want to import. • **Templates**: Displays two panes: the left pane lists all the templates belonging to the source project, and the right pane displays a thumbnail of the template selected in the template list. Check or clear the selection boxes to import the templates or not. The number of the selected templates and the number of the templates to import are displayed at the bottom of the left pane.
Target Directory	The default destination directory is the directory of the current project. You can add subdirectories to the new project by typing a subdirectory name in the text box at the bottom of the window. This subdirectory is created in the current project directory and will contain the imported templates. If the target directory does not already exist, it will be created. **Target directory will be created if it does not exist**: This message means that if the templates reside in subdirectories in the source project, corresponding subdirectories will be created automatically in the destination subdirectory.

Element	Description
Import Templates	• **Imported project**: Name of the imported project. • **Target directory**: Name of the subdirectory created in the current project and that will contain the imported templates. • **Imported files**: Displays all the elements that have been imported. • **Templates**: If a template name already exists in the destination directory, the `_new(<index>)` suffix is added to the name of the imported template. **Index families**: The index family of a template is only imported if the family does not exist in the destination project. If an index family name already exists in the destination directory, then it is not updated. • **Resources**: The content of the source project directory `/Resources/` is copied into the `/Resources/` directory of the destination project. • **Keyword classification rules**: If a template to be imported contains keyword classification rules, they are added to the current project together with their configuration. However, the configuration of the reading zone and the *OCR* engine both remain as defined in the current project. • **Next**: Enables you to go to the following step.
End	Select **End** to close the **Template Import Wizard** and display a summary of the import results. It also displays the number of imported templates. As soon as you close the wizard, the templates are displayed.
Cancel button	Cancels any operation and closes the **Template Import Wizard** window.
Previous button	Enables you to go to the previous step.
Next button	Enables you to go to the following step.
Template Import Wizard toolbar	Provides toolbar buttons for performing common functions.

Related Topics

"Importing Templates" on page 117

"Placing Fields Automatically During Setup Using the Template Wizard" on page 224

5.16 Template Wizard

To place fields and anchors, use the **Template Wizard**. The wizard is able, based on free form settings defined in Free Form Designer, to run a full text search to place fields automatically on a selected set of templates.

Note: Before using the **Template Wizard**, configure Free Form Designer. The **Template Wizard** menu is dimmed if the project does not contain at least one definition file *DFT* and one template.

Table 5-101: Template Wizard

Element	Description
Start	• Left pane: The left pane represents a tree structure enumerating the different steps to help you place index fields in the templates of your choice. Each step is preceded by a colored square. When the colored square is dimmed it means that a step has been successfully completed. • Right pane: The right pane offers a brief presentation of the **Template Wizard** features. Also, it enables proceeding to the next step or to cancelling and closing the **Template Wizard** window.
Free Form Parametering	• **Definition file**: Selects the definition file from the list displaying all the DFT files for the project. All definition files must be saved to `<Project directory>\Resources\OCR\`. If you save these files to another path, they will not be available in Recognition Designer. This path is the default path when you open Free Form Designer from Recognition Designer through the menu **Tools** > **Free Form Designer**. If you open Free Form Designer from the Windows **Start** menu, you will need to select the correct path. • **OCR engine**: Selects a full text engine: Western OCR, General-Use OCR, or Advanced OCR/ICR. For more information on full text engines, see "Recognition Types Supported by Recognition Engines" on page 429.

Element	Description
Choice of templates	• **Current directory**: Displays the directories of the current project. Select templates either from one or more subdirectories. • **Display sub-directories content**: Select this option to select templates from various sub-directories. All project templates (in all sub-directories) are displayed in the list of compatible templates. A template is compatible if its index family and the DFT file both contain at least one field with the same name. The name of the field name in the index family must be exactly the same as it is in the free form (DFT) definition file. • **Compatible templates**: **Template name** pane: To select templates from one sub-directory, select a sub-directory. The list of templates from that sub-directory is displayed in the **Template name** pane. If you select another sub-directory at this point, the templates selected in the current sub-directory will no longer be selected. To select templates from different sub-directories, follow the directions for selecting templates from various sub-directories. The number of selected templates is indicated at the bottom left of the **Template name** pane. Image pane: Displays the image selected in the **Template name** pane.

Element	Description
Automatic editing	To be able to run the **Automatic editing** window, no fields must be placed on the selected templates. If you have already placed the fields on the template, then this window will not appear. In the **Automatic Template** window, editing starts automatically. For each template processed, the template image appears and the following information is also displayed: • **Current Template** Displays the template name, the template code, the index family name and the number of fields to be placed on the image. • **Automatic editing in progress** A progress bar appears while the automatic editing is in progress. As soon as automatic editing has finished, the **End** window appears.
End	Select **End** to close the **Template Wizard** window. The current template is refreshed. This window indicates the result of the automatic positioning of fields for the selected templates. It also displays the number of positioned fields. A field can be considered as being placed with success even if the associated anchor has not been placed.
Cancel button	Cancels any operation and exits the **Template Wizard** window.
Start button	Proceeds to the following step.
Previous button	Returns to the previous step.
Next button	Proceeds to the following step.
End button	Closes the **Template Wizard** window.

Related Topics

"Testing Free Form Rules for Index Fields" on page 160

"Creating and Testing Free Form Rules for Line Item Extraction" on page 163

"1D Barcodes with General-Use OCR" on page 150

5.17 Text Matching Designer

Text Matching Designer enables creation of the classification rules relating to text matching templates and text matching references. These rules are created using an automatic learning feature that uses images of document pages that are typical of the pages that are to be classified using text matching templates in the Classification production module.

Table 5-102: Text-Matching Designer

Element	Description
File menu	This menu enables you to load images from a directory, to apply or modify text matching options and to migrate previous versions of Recognition Designer to the current version. • **Learn** opens the directory selection window. Select a directory. At least one image is present in the selected directory or in one subdirectory of the selected directory. The image format must be a supported image format. A text matching template is created for each subdirectory contained within the selected directory. If there are no subdirectories, a text matching template is created for the selected directory. The template name is the same as the name of the corresponding directory or subdirectory. The automatic learning results are displayed in the **Learning details** pane, in the following tabs: **Summary**, **Skipped images**, and **Images in conflict**. • **Project Options** Opens the **Text Matching** tab of the **Project Options** window. • **Migrate previous version project** applies to users migrating from version 3.6 of the Text Classification management tool to the current version of Recognition Designer. The migration process migrates text matching templates, text matching reference signatures, and any related images if they exist. Recognition Designer updates actual settings with imported setting values.

Element	Description
	– If an imported text matching template has the same name as an existing generic template in the current recognition project, a message appears asking if you want to replace the existing generic template with the text matching template. The new text matching template keeps the same ID as the old generic template. Otherwise, each imported template that has an existing template in the recognition project with the same name is renamed. The same naming convention as for Recognition Designer import functionality is adopted: the `_new(<index>)` suffix is added to the template name. – If images associated to text matching references are not accessible, a selection window appears in which you can select a default image. An image must be selected, otherwise the migration process is cancelled. • **Close** closes the **Text Matching Designer** window.

Element	Description
Edit menu	Enables you to create, rename, select or delete text matching templates and to add, cut or paste text matching references. • **Create new TM template** creates a new text matching template. Select one or more images in the **Open** window. The **Creation of a new TM template** window appears. Type in a name for the template. An empty template is created. • **Rename TM template** displays the **Rename TM template** window to rename a *TM* template. Type in a new name for the template in the **Type in the name of the new TM template** field. Click **OK** to validate the modification or **Cancel** to close the **Rename TM Template** window without saving the modifications. • **Select all** selects all the templates in the current project. • **Delete** deletes the selected templates. When deleting text matching templates, you can select one or several templates, or use the **Edit** > **Select All** menu to select all the templates in the current project. When a text matching template is deleted, all text matching references associated to it are also deleted. • **Add TM references** A text matching reference can be added (or associated) to a text matching template. The reference image can be any image, but its image format must be a supported image format. Select this option, and select one or more image files. Recognition Designer tries first to classify the image with each of the text matching references already defined in other text matching templates in the project. • **Cut TM references** associates text matching references that are associated to other text matching templates to a different template using the cut and paste functionality. Select one or more text matching references in the list of templates. Then cut the TM references. • **Paste TM references** selects a text matching template and paste the TM references. The text matching template references are added to the text matching

Element	Description
	template and are no longer associated to the original text matching templates.

Element	Description
Display menu	Zooms in, zooms out or restores the initial display size of the image. It also enables you to navigate between the elements in conflict in the list of elements and to show/hide the **Learning details** pane. • **Previous Conflict** returns to the previous element that has a conflict in the list of templates. • **Current Conflict** returns to the last conflict that has been selected before having navigated to another template in the list. The first time you select this option, the first element in conflict in the list is selected. • **Next Conflict** proceeds to the next element that has a conflict in the list of templates. • **Refresh Conflicts** refreshes the conflicts in the list. With large projects, this operation can last several minutes. • **Zoom In** • **Zoom Out** • **Default Zoom** restores the initial display size of the reference image. • **Expand All** expands all the reference images grouped in the templates. • **Collapse All** collapses all the reference images grouped in the templates. • **Display Learning Details** contains three tabs that display information relating to the learning process. – **Summary**: Displays the learning process results. – **Skipped Files**: A reference image is skipped when it is too similar with a reference image already defined in a text matching template. This tab contains the list of skipped image files with their original directory name, original file name, and each text matching template to which they match but to which they have not been associated since they have been skipped. – **Images in conflict**: When a reference image matches at least two reference images already in conflict, it is automatically skipped because

Element	Description
	conflicts between two images only are managed. In the **Images in Conflict** tab, these skipped images in conflict are listed, displaying their original directory name, original file name, and the text matching template in conflict. **Note:** The text matching **Learning details** pane no longer appears when you close and reopen the Text Matching Designer.
Test menu	Runs the text matching test.
Tools menu	Exports images. You may want to keep skipped images for use at a later date or for comparison purposes. Use the export image feature that saves the image files, keeping the original tree structure of the images. • **Export Skipped Well Classified Images**: Exports skipped images that are classified. • **Export Skipped Images in Conflict**: Exports skipped images that are in conflict.
Help menu	Opens the **Help** and the **About** window.
Left pane	Loaded images appear in the left pane. They are grouped by templates in a table. Click to see the detailed composition of a group. If the symbol appears in the **Conflict** column of the table it means that the image in the **TM Reference** column is in conflict with another image defined in the **In conflict with** column. Images are in conflict when a reference image matches two templates.
Right pane	Displays the selected image.
Learning details pane	Contains three tabs that display information relating to the learning process: **Summary**, **Skipped Images**, and **Images in conflict**.
Text Matching Designer toolbar	Provides toolbar buttons for performing common functions.

Related Topics

"Creating Text Matching Templates Automatically" on page 83

"Creating Text Matching Templates Manually" on page 84

5.17.1 Creation of New TM Template

You can create text matching templates. At least one text matching template must exist in the project to be able to create a new text matching template.

Table 5-103: Creation of New *TM* Template Window

Element	Description
Type in the name of the new TM template	Select this option to create a text matching template. Then select one or more images in the **Open** window. The **Creation of a New TM Template** window appears. Type in a name for the template in the **Type in the name of the new TM template** field. Click **OK** to validate the creation or **Cancel** to cancel it. An empty template is created.

Chapter 6

Reference

Reference information includes content on keyboard shortcuts for using the application and outlines the support for image formats, languages, recognition types, and barcode types.

6.1 Keyboard shortcuts

Shortcuts followed by "NumPad" indicate that the second key in the sequence must be pressed on the numeric keypad. For example, "**CTRL++** (NumPad)" means press and hold down the **CTRL** key while pressing the **+** key on the numeric keypad.

6.1.1 Anchor Unit Test Keyboard Shortcuts

Table 6-1: Anchor Unit Test Shortcuts

Shortcut	Action
CTRL+O	Load images
CTRL+Q	Close
CTRL++ (NumPad)	Zoom in
CTRL+- (NumPad)	Zoom out
CTRL+* (NumPad)	Center image

6.1.2 Table Field Unit Test Keyboard Shortcuts

Table 6-2: Table Field Unit Test Shortcuts

Shortcut	Action
CTRL+O	Load images
CTRL+Q	Close
CTRL++ (NumPad)	Zoom in
CTRL+- (NumPad)	Zoom out
CTRL+* (NumPad)	Center image
CTRL+R	Refresh resources

6.1.3 Main Interface Keyboard Shortcuts

Table 6-3: Main Interface Shortcuts

Shortcut	Action
CTRL+E	Project Update
CTRL+N	New Project
CTRL+O	Open Project
CTRL+P	Move to Production
CTRL+S	Save
CTRL+F	Search Templates
CTRL+I	Show / Hide Images Base
CTRL+F9	Compile Project
ARROW KEYS	Navigate in the template list
CTRL+M	Template Test
CTRL+T	Test field
F9	Classification test
F1	Help

6.1.4 <Project Name> Keyboard Shortcuts

Table 6-4: <<Project Name>> Keyboard Shortcuts

Shortcut	Action
CTRL+O	Tree structure
CTRL+A	Select All
CTRL++ (NumPad)	Zoom in
CTRL+- (NumPad)	Zoom out
CTRL+* (NumPad)	Default zoom

6.1.5 Edit Field-Specific Types Keyboard Shortcuts

Table 6-5: Edit Field-Specific Types Keyboard Shortcuts

Shortcut	Action
CTRL+N	New
CTRL+O	Open
CTRL+S	Save
CTRL+Q	Exit
CTRL+T	Unit test
CTRL+M	General test

6.1.6 Edit Index Families Keyboard Shortcuts

Table 6-6: Index Family Editor Keyboard Shortcuts

Shortcut	Action
CTRL+X	**Cut**
CTRL+C	**Copy**
CTRL+V	**Paste**
DELETE	**Delete**

Table 6-7: Script Editor Keyboard Shortcuts

Shortcut	Action
CTRL+Z	**Undo**
CTRL+Y	**Redo**
CTRL+F5	Performs a syntax check on the project script and highlights the first error in red if one exists.
TAB	**Indent**: Move text to the left.
SHIFT+TAB	**Outdent**: Move text to the right.
CTRL+ALT+C	**Comment**: Turn the current selection/line in the code into a comment.
CTRL+ALT+U	**Uncomment**: Delete the comment attribute for the current selection/line in the code.
CTRL+F	**Find**
CTRL+R	**Replace**
F3	**Again**: Repeat last find or replace.

Shortcut	Action
CTRL+SPACE	**Complete Word**: Automatic word completion.
CTRL+I	**Parameter Info**: Display parameter information.
CTRL+A	**Macro**: Open the macro editing window.
CTRL+E	**Immediate**: Show the immediate output window.
CTRL+P	**Print**
CTRL+W	**Watch**: Display the **Watch Expressions** window.
CTRL+T	**Stack**: Show the **Call Stack** window.
CTRL+L	**Loaded**: Display macros and modules currently loaded.
F5	**Run**: Start a macro, or run the loaded macro to completion.
CTRL+ALT+ESC	**Pause**: Stop the macro or module. Execution can be resumed by clicking on **Run**.
SHIFT+F5	**End**: End a macro or a module.
F8	**Step into**: Run the current line. If it is a function call or subroutine, stop on the first line of the function call or subroutine. Start inactive macros.
SHIFT+F8	**Step over**: Run to the next line. If the current line is a function call or subroutine, execute to the end.
CTRL+F8	**Step out**: Move out of the current subroutine or function.
F7	**Step to cursor**: Execute until the cursor is reached. Start inactive macros.
F9	**Toggle break**: Toggle any breakpoints on the current line on or off.
CTRL+SHIFT+F9	**Clear all breaks**: Clear all breakpoints.
SHIFT+F9	**Quick Watch**: Display the value of an expression under the cursor in the **Immediate** window.
CTRL+F9	**Add Watch**: Add any expressions beneath the cursor to the watch list.

Table 6-8: Edit Index Families Keyboard Shortcuts

Shortcut	Action
DEL	Delete the selected box
CTRL+INSERT	Add field
CTRL+A	Select All
CTRL+D	Delete a field
CTRL+E	Save as
CTRL+N	New
CTRL+O	Open
CTRL+F4	Close
CTRL+S	Save
CTRL+SHIFT+S	Save All
F9	Test index fields
CTRL+F9	Test table fields

6.1.7 Field Unit Test Keyboard Shortcuts

Table 6-9: Field Unit Test Keyboard Shortcuts

Shortcut	Action
CTRL+O	Load images
CTRL+R	Refresh resources
CTRL++ (NumPad)	Zoom in
CTRL+- (NumPad)	Zoom out
CTRL+* (NumPad)	Center image

6.1.8 Free Form Designer Search Keywords Keyboard Shortcuts

Table 6-10: Free Form Designer Search Keywords Keyboard Shortcuts

Shortcut	Action
CTRL++ (NumPad)	Zoom in
CTRL+- (NumPad)	Zoom out
CTRL+F	Search quickly a text string
F3	Search next
F1	Help

6.1.8.1 Free Form Designer Settings Keyboard Shortcuts

Table 6-11: Free Form Designer Settings Pane Keyboard Shortcuts

Shortcut	Action
CTRL+N	New
CTRL+O	Open
CTRL+S	Save
DELETE	Delete
SHIFT+UP ARROW	Move the selected item up
SHIFT+DOWN ARROW	Move the selected item down
CTRL+C	Copy
CTRL+V	Paste
F1	Help
* (NumPad)	Expands the tree structure

6.1.8.2 Free Form Designer OCR Reading Keyboard Shortcuts

Table 6-12: Free Form Designer OCR Reading Pane Keyboard Shortcuts

Shortcut	Action
CTRL++ (NumPad)	Zoom in
CTR+- (NumPad)	Zoom out
F1	Help

6.1.9 Image Analyzer Keyboard Shortcuts

Table 6-13: Image Analyzer Keyboard Shortcuts

Shortcut	Action
F9	Analyze

6.1.10 Template Test Keyboard Shortcuts

Table 6-14: Template Test Keyboard Shortcuts

Shortcut	Action
CTRL+O	Load images
CTRL+T	**Table Wizard**
CTRL+R	Refresh resources
CTRL++ (NumPad)	Zoom in
CTRL+- (NumPad)	Zoom out

6.1.11 Text Matching Designer Keyboard Shortcuts

Table 6-15: Text Matching Designer Keyboard Shortcuts

Shortcut	Action
CTRL+L	Learn
CTRL+N	Create new *TM* template
F2	Rename a TM template
CTRL+A	Select All
DELETE	Delete
CTRL+X	Cut TM reference(s)
CTRL+V	Paste TM reference(s)
CTRL++ (NumPad)	Zoom in
CTRL+- (NumPad)	Zoom out
F9	Test
F1	Help

6.2 Advanced Recognition Supported Image Formats

All the advanced recognition modules support the formats outlined in "Supported Image Formats" on page 426.

Table 6-16: Supported Image Formats

File Format	Color Format	Compression
JPEG (`*.JPG`)	24 bit Color	Progressive JPEG Sequential JPEG
	8 bit Gray	Sequential JPEG
TIFF (`*.TIF`)	24 bit Color	No Compression JPEG in TIFF Packbits Progressive JPEG Sequential JPEG Wang JPEG ZIP
	8 bit Gray	No Compression JPEG in TIFF Packbits *LZW* ZIP Sequential JPEG Wang JPEG

File Format	Color Format	Compression
	Binary	*ITU* Group 3 ITU Group 4 JBIG Enhanced *JBIG* LZW Modified *G3* No Compression Packbits ZIP

6.3 Languages Supported by Recognition Engines

"Languages Support by Recognition Engine" on page 428 shows some of the languages or countries supported for recognition engines. Where appropriate, a complete list of languages is provided.

Table 6-17: Languages Support by Recognition Engine

Engine Name	Supported Languages/Countries
Advanced OCR/ICR	Supports many languages, including English. Supported countries include Australia, Austria, Azerbaijan, Belgium, Brazil, Bulgaria, Canada, Central America, Central Europe, Croatia, Czech Republic, Denmark, Estonia, Faroese, Finland, France, Germany, Great Britain, Greece, Hungary, International (for *OCR* and *MICR* classifiers only), Ireland, Italy, Liechtenstein, Lithuania, Luxembourg, Malaysia, Netherlands, New Zealand, Norway, Poland, Portugal, Romania, Russia, Rwanda, Scandinavia, Slovakia, Slovenia, Somali, South Africa, South America, Spain, Sweden, Switzerland, Turkey, United States, and Western Europe. The Advanced OCR/ICR engine cannot read Asian characters, except when the Intelligent Capture Asian Language Add-on is installed. The Intelligent Capture Asian Language Add-on enables the following Asian languages for Advanced OCR/ICR: • Chinese (Simplified) • Chinese (Traditional) • Chinese (Traditional, Hong Kong) • Japanese • Korean • Thai **Note:** With or without the Intelligent Capture Asian Language Add-on installed, the Advanced OCR/ICR engine can still be used on Asian operating systems to read Western characters.
General-Use OCR	Twenty languages installed by default and up to 123 languages available, including English, Simplified Chinese, Japanese and Korean.

Engine Name	Supported Languages/Countries
Western OCR	English, German, Danish, Spanish, Finish, French, Dutch, Italian, Norwegian, Portuguese, and Swedish. The Western OCR engine cannot read Asian characters, but can be used on Asian operating systems to read Western characters.
Basic French ICR	French
Basic OCR	French
Check Reading	English (United-States) and French. The Check Reading engine cannot read Asian characters, but can be used on Asian operating systems to read Western characters.

6.4 Recognition Types Supported by Recognition Engines

Intelligent Capture supports the following recognition types:

- **Machine printed**: Recognizes machine printed alphanumeric characters that have consistent, predictable shapes including fixed pitch, variable pitched, and kerned fonts. This recognition type is also referred to as Optical Character Recognition (*OCR*).
- **Precise machine printed**: Recognizes machine printed alphanumeric characters that have consistent, predictable shapes including fixed pitch, variable pitched, and kerned fonts. This recognition type is also referred to as Optical Character Recognition (OCR). This method favors recognition quality over speed.
- **Hand printed**: Recognizes alphanumeric characters that vary in shape, such as hand printed characters. This recognition type is also referred to as Intelligent Character Recognition (*ICR*).
- **Mark sense**: Recognizes checkmarks, Xs, or other marks placed in checkboxes. This recognition type is referred to as Optical Mark Recognition (*OMR*).
- **Barcode**: Recognizes industry-standard barcodes. Barcodes are comprised of self-contained information encoded in the widths of printed bars and spaces.
- **Automatic**: Determines whether the characters are machine printed or hand printed, then applies that recognition type.
- **Cursive**: Recognizes handwritten characters.
- **9 pins or 24 pins dot matrix**: Recognizes alphanumeric characters generated on a 9 pin or 24 pin dot-matrix printer.
- *MICR/CMC***7**: Recognizes the code line for checks.

- **CAR/LAR**: Recognizes courtesy amount (amount in figures) and legal amount (amount in letters) in *US* checks.

"Supported Recognition Type by Engine" on page 430 shows the characteristics of the recognition engines available.

Table 6-18: Supported Recognition Type by Engine

Engine Name	Processing Type	Recognition Type	License Type	Barcode Type
Optical Mark Recognition	Zonal	Mark sense	Delivered as standard	None
Barcode 39 Recognition	Zonal	Code 39 barcodes	Delivered as standard	Code 39
Basic French ICR	Zonal	Hand printed	Delivered as standard	None
Basic OCR	Zonal	Machine printed	Delivered as standard	None
Modification Detection	Zonal	Hand printed Machine printed	Delivered as standard	None
Multi-Engine Voting	Full text Zonal	Hand printed Machine printed	Delivered as standard	None
OCR/ICR Voting	Zonal	Automatic	Delivered as standard	None

Engine Name	Processing Type	Recognition Type	License Type	Barcode Type
General-Use OCR	Full text Zonal	Hand printed Machine printed 1D barcodes	Machine printed – delivered as standard. 1D barcodes – delivered as standard. Hand printed requires an additional license **Note:** The three processing types can be run at the same time.	**1D Barcodes**: • Codabar • Codabar with start-stop char transmit • Code 128 • Code 128 with check digit transmit • Code 39 • Code 39 full *ASCII* mode • Code 39 with check digit control and transmit • Code 39 with start-stop char transmit • *EAN*8/13 • EAN/*UPC* with 2 and 5 digit supplement • *ITF* (2 of 5 interleaved) • ITF with check digit control and transmit • Postnet code • *UCC* Code 128 • UPC-A • UPC-E (6-digit)

Engine Name	Processing Type	Recognition Type	License Type	Barcode Type
Advanced OCR/ ICR	Full text Zonal	Hand printed Machine printed	License with two processing types and possibility of having both at the same time: • Zonal: limited to 25, 50, 100 or 250 characters/ second • Full text: 360, 450, 600, 900, 1800 and 3600 documents per hour	None
Check Reading	Zonal	Handwritten Machine printed MICR/CMC7 - *CAR/LAR*	License with two processing types that can be activated separately with the possibility of running both at the same time: • Check Reading US • Check Reading France	None

Engine Name	Processing Type	Recognition Type	License Type	Barcode Type
Barcode Recognition	Zonal	1D barcodes 2D barcodes	Delivered as standard	**1D Barcodes**: • Add 2 • Add 5 • Airline 2 of 5 (*IATA* 2 of 5) • Australian Post • *BCD* Matrix • Codabar • Code 2 of 5 (Industry 2 of 5) • Code 32 • Code 39 • Code 39 Extended • Code 93 • Code 93 Extended • Code 128 • *UCC/EAN* 128 • DataLogic 2 of 5 • *EAN* 8 • EAN13 • Intelligent Mail • Interleaved 2 of 5 • Invert 2 of 5 • Matrix 2 of 5 • Patch Code • Royal Post • UPC-A • UPC-E • PostNet **2D Barcodes**: • *PDF*-417 • *QR*

Engine Name	Processing Type	Recognition Type	License Type	Barcode Type
				• Data Matrix
Western OCR	Full text Zonal	Machine printed	Delivered as standard	None

6.5 Supported Barcode Types

"Supported Barcode Types" on page 434 outlines the supported barcode types.

Table 6-19: Supported Barcode Types

Barcode Type	Description	Examples
Code 39	Code 39 is an alphanumeric barcode. This 1D barcode is used in industrial application and has both original and extended versions. The original version enables to encode 43 characters including digits 0 to 9, letters A to Z, 6 symbols, plus one special character (*) that marks the beginning and end of the barcode. This character is not read during recognition. The extended version enables encoding of all *ASCII* table characters (128 characters). The 39 barcode has a variable, bidirectional length. Its name comes from its structure, 3 of 9 and is sometimes called Code 3 of 9 code or USD-3. Each character is encoded by 9 elements (5 bars, 4 spaces), of which 3 are large (1 binary) and 6 straight (0 binary). All characters are separated by a space, which are not counted as characters.	

Barcode Type	Description	Examples
1D barcode	1D barcodes store a short message, such as a room number, customer number or serial number.	Code 93  Airline 2 of 5  CODABAR  UPC-A  Code 128  UPC-E  Interleaved 2 of 5 

Barcode Type	Description	Examples
		Postnet (Message: 94501-3511)  EAN-8  EAN-13  IATA 2 of 5 (Airline 2 of 5)  Industry 2 of 5  Matrix 2 of 5 

Barcode Type	Description	Examples
		

Barcode Type	Description	Examples
2D barcodes	2D barcodes such as PDF417, QR, and DataMatrix store large amounts of data such as the description of the content of a package for example. The *PDF417* barcode is the most widely used 2D barcode. Its capacity is 1850 ASCII characters.	PDF417  QR  DataMatrix 

Glossary

AN

Alphanumeric

ASCII

American Standard Code for Information Interchange

BCD

Bar Code Detection

CAR

Courtesy Amount Recognition

CCITT

Comité consultatif international téléphonique et télégraphique (French for International Telegraph and Telephone Consultative Committee, became the ITU Telecommunication Standardization Sector, ITU-T, in 1992)

CPU

Central processing unit

CSV

Comma Separated Variable

DFT

Dispatcher definition file extension

DOM

Dispatcher Object Model

dpi

Dots Per Inch

DPP

Dispatcher project file extension

EAN

European Article Number

G3

CCITT Group 3

GB

Gigabyte

HPA

High Precision Anchor

IATA

International Air Transport Association

ICR

Intelligent Character Recognition

ITF

Interleaved Two of Five

ITU

Type of image compression

JBIG

Joint Bi-level Image Experts Group

JPEG

Joint Photographic Experts Group

KFI

Key from Image

LAR

Legal amount recognition

LCK

Lock file extension

LIFFE

Line Item Free Form Engine

LZW

Lempel-Ziv-Welch compression algorithm

MB

megabyte

MFP

Multi-function Peripheral

MICR

Magnetic Ink Character Recognition

MSDN

Microsoft Developer Network

OCR

Optical Character Recognition

OMR

Optical Mark Recognition

PAL

Production Auto-Learning

PDF417

Portable Data File 417 barcode format

PDF

Portable Document Format

QR

Quick Response

RAM

Random Access Memory

SSN

Social Security Number

TFT

Dispatcher field-specific type file extension.

TIF

Tagged Image File file extension

TIFF

Tagged Image File Format

TM

Text Matching

UCC:EAN

Uniform Commercial Code : European Article Number

UCC

Uniform Commercial Code

UNC

Universal Naming Convention

UPC-A

Universal Product Code, 12-digit common version

UPC-E

Universal Product Code, 6-digit zero compressed version

UPC

Universal Product Code

US

United States

VAT

Value Added Tax

VB

Microsoft Visual Basic

VBA

Microsoft Visual Basic for Applications

XML

Extensible Markup Language