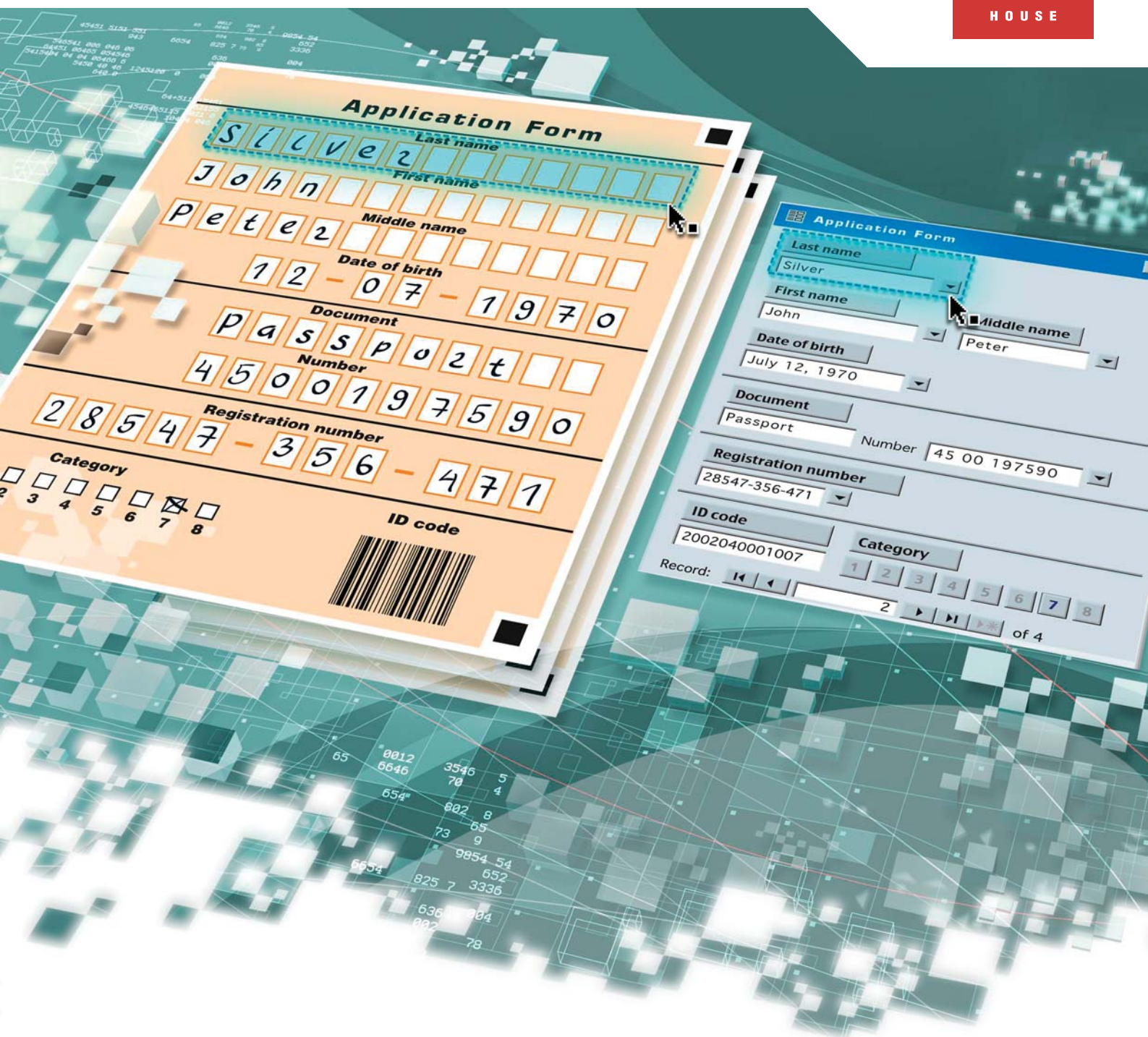


ABBYY

SOFTWARE
HOUSE



AUTOMATED FORMS PROCESSING

Table of Contents

Introduction	3
Form Types	3
What is a form?	3
Form structure	4
Form types and design elements	5
What is forms processing?	6
The cost of manual processing	7
Automated forms processing	8
OCR/ICR basics	9
Automated Forms Processing	10
Where data capture should be used?	10
Designing a form	11
Determining the form's logic	11
Selecting form type and design	11
Drawing a form	12
Setting up FormReader	13
Selecting a scanner	14
Personnel training	15
Processing cycles	15
Ensuring Data Quality	17
Defining data quality	17
Image pre-processing	17
Data type checks	18
Verification	19
Data format checks	20
Controlling logic	21
Processing multi-page forms	22
Operator stress as an important quality factor	22
Organizing Automated Forms Processing	23
Approaches to data capture	23
Front-office data capture	23
Back-office data capture	24
Data capture basics	25
Batch processing	25
Operator specialization	25
Scalability	25
Processing queues	25
Data flows	25
Production capture	26
Using ABBYY Technologies to Solve Untypical Tasks	27
What if FormReader does not support a required language?	27
Remote scanning and processing faxed forms	28
Distributed verification	28
Processing flexible forms	29
Capturing data from forms that are not machine-readable	29
Conclusion	31
Contacts	32



Introduction

In the course of our lives we fill in hundreds of forms - application forms, questionnaires, insurance claims, etc. At the same time computers have become indispensable for collecting and managing information, making the task of extracting data from printed documents even more pressing.

This White Paper presents an overview of the existing data capture technologies used to extract hand-printed text from completed forms and explains in detail the principles behind ABBYY FormReader, a data capture solution that is used to process forms in more than 30 countries.

Form Types

What is a form?

A form is a document with blank spaces to be filled in with particulars before it is executed. These blank spaces are called fields and are usually provided with explanations or captions that tell people what kind of information and in what format is to be entered into each particular field.

Forms are used whenever information must be collected from a large number of people. Government bodies in particular make wide use of all sorts of forms. In Russia, for example, forms are extensively used by the Tax Ministry and the Pension Fund. The former collects and processes **tax returns** filled in by hand and the latter collects **social security forms**.

Forms are also widely used in business. Insurance companies, for example, have to handle thousands of insurance applications and insurance claims, marketing agencies have to deal with opinion polls and customer surveys, and educational institutions make extensive use of forms in all sorts of examinations and formalized

tests. The banking industry also uses forms when issuing credit cards or handing out loans to their clients. There are also mail orders, coupons, medical forms, utility bills and many more - the list is practically endless.

The image displays a collage of various paper forms, illustrating different types of forms used in different contexts. The forms shown include:

- HGV LICENCE APPLICATION FORM**: A form for applying for a Heavy Goods Vehicle (HGV) licence, featuring sections for personal details, vehicle details, and tax information.
- West travel service order form**: A form for ordering travel services, including sections for personal details, travel preferences, and payment information.
- Parent Financial Aid Application Form School Year 2000-2001**: A form for applying for financial aid, featuring sections for student information, family income, and academic performance.
- ADAMS**: A form for the American Dental Association's (ADA) dental procedure coding system, featuring a grid for recording dental procedures.

The forms are arranged in a collage, showing their layout and structure. The forms are printed on white paper with black text and lines. The forms are designed to be filled out by hand, with various fields for text entry, checkboxes, and radio buttons.

Different types of paper form.

When completing a form one has to enter information into blank spaces or specially designed fields that make up the structure of the form. This information must then be extracted and processed. Forms from which data can be extracted, or "captured", automatically by computer are called machine-readable. Almost any form can be structured in such a way as to become machine-readable.

Form structure

Sometimes people filling in a form are too careless or sloppy. For this reason forms are designed in such a way as to make their completion intuitive and self-evident. The following **design elements** are used to tell people where to write what.

- **Entry (or data) fields.** These include
 - ✓ **Text fields.** Each text field consists of a certain number of character spaces supplied with an explanatory caption. Character spaces stand apart so that the entered letters do not merge.
 - ✓ **Check boxes.** These are fields of various shapes (usually squares, but in practice this can be any geometrical figure with a closed boundary). A person filling in the form makes a mark such as a check, a tick or a cross in this field to select a particular option. Or they may simply ink over the entire box.
 - ✓ **Groups of check boxes.** These are used for multiple choices. Usually check boxes within one group correspond to mutually exclusive options, i.e. only one of them must be selected.
- **Service fields.** Service fields contain so-called anchor or reference points that facilitate forms processing. Anchor points are used by a data capture program to detect the top and bottom of a form and to correct distortions introduced by scanning. Anchor points may also be used to identify different forms if mixed types of forms are processed within one batch. The following elements may be used as reference points on forms processed by ABBYY FormReader:
 - ✓ black squares, corners and crosses;
 - ✓ vertical or horizontal lines;
 - ✓ static text, i.e. field captions that remain unchanged from form to form.

Forms can be filled in:

- ✓ by hand (such forms are called hand-printed, because information is entered in separate block letters, each letter occupying one character space);
- ✓ using a typewriter or printer
- ✓ in a printing house;
- ✓ using a combination of all of the above.

- **ID fields or identifiers.** These fields serve to identify the form. Black squares, corners and crosses can also be used to identify forms, but identification is more reliable if forms are identified using such identifiers as numbers, bar codes or form titles.
- **Image areas.** These areas contain objects which are not to be recognized, e.g. seals or signatures which will be treated as pictures. FormReader can save such images into an ODBC database in the following formats: TIF, BMP, JPG, PCX, and WMF.
- **Optional design elements:** logos, headers, footers and other formatting elements. In data capture, data contained in these elements can also be used to identify forms, e.g. by analysing text in logos the program can find out which company has issued the invoice.

The diagram shows a 'NatWest travel service order form' with various sections. Annotations with arrows point to specific design elements:

- service fields:** Points to the 'PERSONAL DETAILS' section and the 'FOREIGN CURRENCY NOTES' section.
- identifier:** Points to the 'NatWest' logo and the 'travel service order form' title.
- check boxes:** Points to the 'I would like to pay by' section, which includes options like 'Debit to my NatWest account', 'Cash', 'NatWest credit card', 'Other credit card', 'Personal cheque', 'NatWest annual multi trip travel insurance policy', 'Policy #', 'Advantage Premier', 'Advantage Gold', 'NatWest private banking customer relationship manager', and 'Students/Graduate service'.
- text fields:** Points to the 'Currency name' and 'Code (*)' fields in the 'FOREIGN CURRENCY NOTES' section, and the 'Traveller's name' field in the 'AMERICAN EXPRESS TRAVELLERS CHEQUES' section.

Examples of form elements.

Form types and design elements

Forms can be divided into two major classes - structured forms, on which the locations and sizes of all fields are exactly the same for all forms in a batch, and flexible forms, on which the sizes and locations of fields may vary from form to form. In order to capture data from a structured form, a program has to know where to look for data. For this purpose a template is created which is essentially a skeleton of a form that contains information about the locations of fields and the kind of data the program may expect to find in each of them. The program will then match this template with a completed form and separate the entered data from the field borders and captions. Next, the entered data are "read" or recognized, i.e. converted into text and digits.

All the forms in a batch must conform to one and the same pattern. It is also essential that reference points and ID fields are preserved during scanning.

If a form is not structured, it cannot be processed automatically and requires a human operator to read the data from its fields and type them into a database. This is a slow and tedious process that can be avoided by designing a well-structured form that can then be read by computer.

Depending on their design, machine-readable forms can be divided into the following three **major types**:

- **Colour forms.** All data fields on such forms consist of white rectangles printed on a colour background. Backgrounds are usually light grey, pink, orange, or green. The colours and saturation are selected so that the background disappears during scanning (this is why they are also known as drop-out colours). Ideally, all elements must disappear during scanning with the exception of reference points and ID fields. Special scanners with red or green lamps are used to scan such forms.

Colour drop-out form.

Alternatively, the drivers of common scanners may be adjusted so that they become blind to the background. Colour forms provide the best recognition quality.

- **Raster forms.** Data fields on such forms consist of white rectangles printed on a colour background, but unlike on colour forms, backgrounds are made up of small dots located at regular intervals from one another. These dots do not disappear during scanning, but ABBYY recognition software can remove such dots without losing information entered into the data fields. There is also a subtype of raster form which has no background at all. The borders of data fields on such forms are made

Raster field borders.

up of separate dots which can then be filtered out by ABBYY software.

- **Black-and-white linear forms.** Field borders on such forms consist of solid black lines which do not disappear during scanning.

The following field designs are available for linear forms:

(a) solid lines

(b) frames for words

(c) isolated frames for characters

(d) conjoined frames for characters

(e) lines with "combs"

(f) frames with "combs"

The recognition engine separates the data from the field borders and then recognizes them. ABBYY FormReader uses information about the field design provided on the template and looks for specific design elements such as vertical lines or the number of character cells. The program then ignores the formatting and recognizes only the data contained within the fields. A form may also contain "garbage" or undesirable artefacts resembling field lines. The program will remember the shape of the fields and distinguish between the meaningful field borders and the arbitrary "noise" which will be removed so that it does not interfere with recognition.

Jersey Public Services Concessionary Travel Pa Application Form

Please PRINT CLEARLY IN BLOCK CAPITALS using black ballpoint pen, one letter per box or in between boxes will be ignored. If you make a mistake please start a new line to help you complete this form.

I am applying for: (Please tick one box)

☐ the new CONNEX travel pass ☐ a replacement CONNEX

I am applying for a Concessionary Travel Pass on the basis of: (Please tick only)

☐ Age ☐ Health Insurance Exemption - Valid to (Head of Household Only)

Social Security Number:

Title: ☐ Mr ☐ Mrs ☐ Miss ☐ Other

Gender: ☐ Male ☐ Female

First Name (s):

A black-and-white form on which characters are to be entered into separate frames.

What is form processing?

Forms processing is a process whereby information entered into data fields is converted into electronic form:

- entered data are "captured" from their respective fields
- forms themselves are digitised and saved as images.

In most cases forms processing is considered complete when the data from all the forms have been captured, verified and saved into a database. It is also essential that the integrity of the captured data be preserved.

Many people still prefer to process forms manually, even though this is not the most efficient and reliable method. Here is a list of typical actions that need to be performed in the case of manual data entry:

- Each **human operator (keyer)** must be provided with a working place. This entails the most expenses, since each operator must be provided with a computer connected to the local area network, and the average productivity of a qualified operator is no more than 200 forms per day.
- Forms pre-processing requires **sorting operators** and **input controllers**. Controllers make sure that no pages are lost if a form has more than one page and oversee the sorting process. The number of sorting operators and input controllers depends on the expected work load. On average, one sorting operator will sort up to 1,000 forms per day, and one input controller will handle up to 300 forms per day
- Once the data from forms have been entered into a computer, they must be checked by **verifiers**. Verifiers check the data entered by keyers and correct any errors that may have occurred.
- Finally, a manager is required to supervise the entire data entry team.

Now suppose you need to enter data from 1,000 forms per day. You will need five keyers, one input controller and one manager. This means seven desks, seven chairs, seven PCs and additional equipment - network adapters and UPS.

	Costs,USD	Qty	Total, USD
PC	1,000	7	7000
Office furniture	1,000	7	7000
Network and other equipment			1,000
			15000

Table 1. Lump-sum costs for manual processing at 1,000 pages per day.

As has been mentioned earlier, forms can be processed manually or using forms processing software. In the sections that follow we consider the advantages and disadvantages of each method.

The lump-sum costs stand at around USD 15,000. Now let's count your monthly costs for the same productivity. You will need an office of at least 50 sq. m. which may cost you around 1,000 per month. Labour costs will amount to USD 1200 for the operator and controller and another USD 2000 for the manager.

	Costs, USD	Qty	Total, USD
Operators' salary	1,200	5	6,000
Controller's salary	1,200	1	1,200
Manager's salary	2,000	1	2,000
Office space	20	50 sq. m.	1,000
			10,200

Table 2. Monthly costs for manual processing at 1,000 pages per day.

Note that these calculations do not include the cost of electricity, telephone, cleaning, fill-in staff, etc. But even this austere budget stands at around USD **10,200** per month

The cost of manual processing

In the previous section you saw that the lump-sum and running costs of manual forms processing add up to a pretty sum. And we have the first conclusion.

Manual processing is expensive.

But money is not the only problem associated with manual forms processing. You will need additional staff and another tier of management. Obviously it takes some time to set up a team of 8-10 employees and buy the necessary equipment. And some of the new staff may not like this tiresome job and leave.

Now suppose your client needs his forms processed by tomorrow or by the day after tomorrow. Obviously, high costs is not the only problem - you simply won't be able to kick-start the whole process within these two days. The second conclusion suggests itself.

Manual processing takes time to set up.

Another important point is that whatever the size of your processing team, you won't be able to increase their productivity quickly - hiring additional operators is useless unless you provide them with the right equipment. This equipment will require additional office space. Hiring additional staff entails costs which are comparable to the lump-sum costs of setting up the entire team. The third conclusion is:

There is a host of other problems. The most critical of them have to do with the human factor, and this is practically unsolvable. Manual data entry is a tedious job - try typing, for example, a newspaper article in your word processor. This means that even experienced keyers will make mistakes, and their number tends to increase towards the end of the working day. Some of these mistakes will be corrected by the output controller, but controllers are

also human, and the quality of the output data tends to deteriorate. And typing is a great strain for the eyes, so you are likely to get complaints from your staff as early as within the first two months.

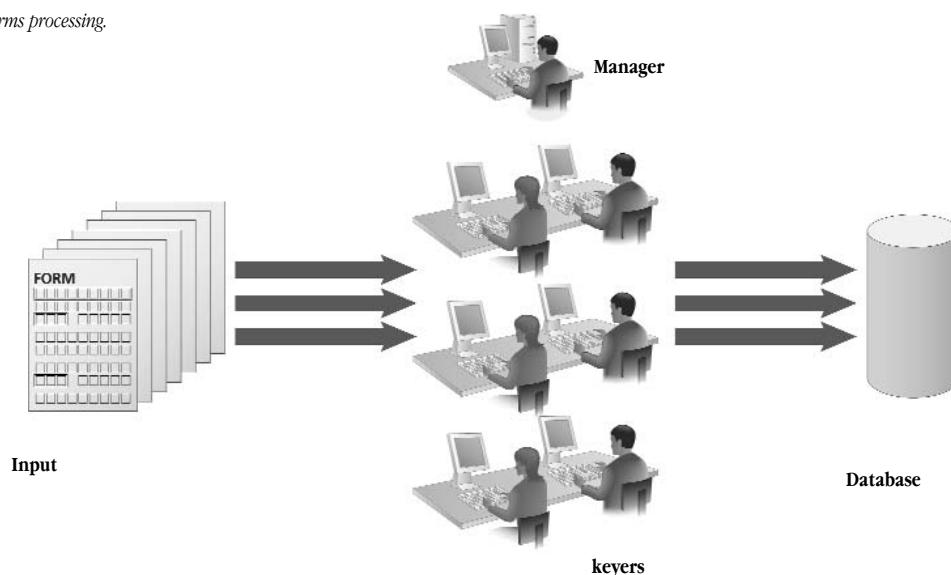
Manual processing is not easily scalable.

The quality of the output data is likely to be unacceptably low because a human operator cannot verify data character by character for hours. Your customer will never be happy with an error-ridden database which your team of operators took so long to create. Two other conclusions arise:

Your staff won't like the job. And you won't like the results of their work.

It follows, then, that manual forms processing is not the best solution, particularly for companies which need to process large number of forms regularly.

Scheme of manual forms processing.



Automated forms processing

An alternative is a data capture solution such as ABBYY FormReader. This is how FormReader works:

- A batch of completed forms is scanned using a high-speed scanner (usually scanners that scan at least 10 pages per minute are used);
- Most of the data are recognized automatically;
- A few characters about which the program is uncertain are passed on to a human operator;
- Verified data are saved into a database.

It is noteworthy that the entire process requires only one human operator since all of the stages, except verification, are fully automated.

The operator's workplace must be equipped with one scanner and one PC connected to the local area network. This workplace can be set up within one day and does not require a lot of office space.

Neither manual sorting nor checking for missing pages is required, since FormReader can identify forms and select the matching template.

With ABBYY FormReader 6.0 Desktop Edition, one operator will be able to process from 1,000 to 3,000 forms per day depending on the complexity of their layout.

Now let us estimate the possible one-time and monthly costs for processing the same 1,000 pages per day using ABBYY FormReader.

	Costs, USD	Qty	Total, USD
PC	1,000	1	1,000
Scanner	1,500	1	1,500
Office furniture	1,000	1	1,000
Software licence	1,695	1	1,695
Software installation and setup	1500	1	1,500
			6,695

Table 3. Lump-sum costs for FormReader at 1,000 pages per day.

	Costs, USD	Qty	Total, USD
Main operator	1,200	1 person	1,200
Fill-in operator	1,000	1 person	1,000
Office space	50	10 sq. m.	500
Scanner maintenance	-	-	50
			3,250

Table 4. Monthly costs for ABBYY FormReader at 1,000 forms per day.

The costs of manual and automated processing compared:

	Manual processing USD	Form processing with FormReader USD	Money saved USD
Lump-sum costs	15,000	6,695	8,305
Monthly costs	10,200	3,250	6,950

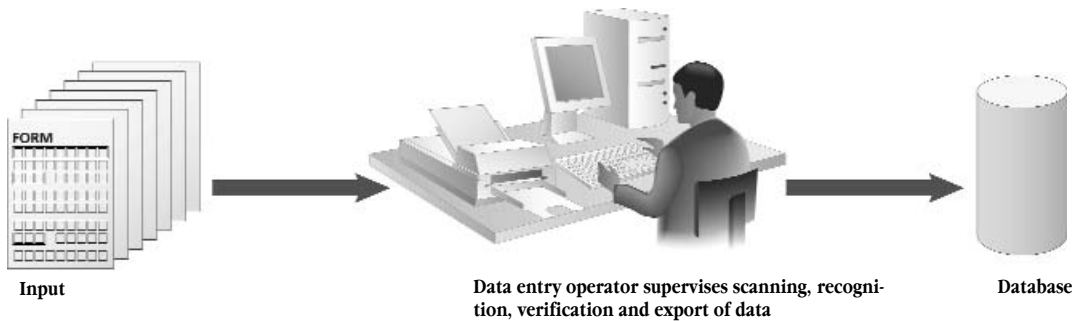
Table 5. Money you can save when processing 1,000 forms per day using ABBYY FormReader.

These figures talk for themselves, but, more importantly, FormReader will solve **all of the five problems** discussed above.

ABBYY FormReader is a highly scalable solution - you only need a few more FormReader modules and several additional operators (whom it will take just hours to train). **There is no other way to increase productivity tenfold within just one day.**

It goes without saying that the quality of output data will be much higher, because the role of the human factor will be reduced to a minimum. Most of the job will be done by computers which never get tired and never make typos. What's more, FormReader can use specially designed validation rules ensuring even higher data integrity and reliability.

Automated forms processing.



OCR/ICR basics

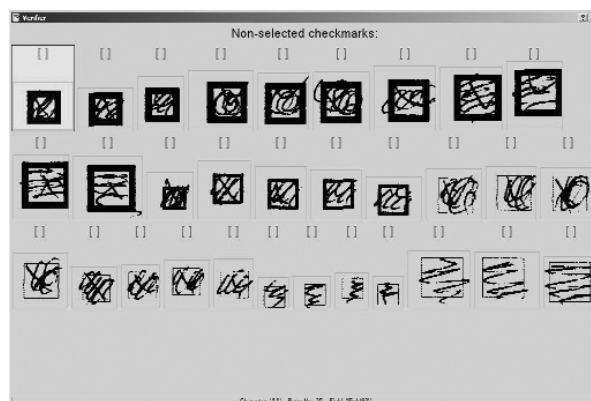
There are two major types of character recognition - Optical Character Recognition (OCR) and Intelligent Character Recognition (ICR). OCR programs recognize characters printed using a printer, a plotter or a typewriter. ICR programs read documents filled in by hand in block letters (so-called hand-print recognition). Let us consider the main differences between OCR programs and ICR programs.

An OCR program first analyses the image and divides it into zones which include text, tables, illustrations, etc. Next, it divides these zones into smaller objects: paragraphs, lines, words, and characters. Once the characters have been recognized by the character classifiers, the OCR program will assemble them back into words, lines, paragraphs, etc., until it gets an electronic version of the original paper document.

ICR programs, which are mainly used to process hand-filled forms, work differently. First, an ICR program detects zones that are expected to contain meaningful data entered by the user. These zones are then processed by the program's modules, including the character classifiers. ICR programs do not attempt to recreate the original document. Instead, they are extracting information from particular fields and save it into a database.

An important feature of an ICR program is mark sense recognition, or recognition of marks in check boxes. Check boxes are widely used on all sorts of forms, because they make their completion easier and can increase the reliability of output data up to 99.9%. ABBYY FormReader 6.0 can recognize all sorts of marks. Mark sense recognition is usually referred to as OMR (Optical Mark Recognition) and works as follows: when creating a template, the operator singles out a check-box zone where the program has to look for a mark; the program then analyses these zones on completed forms and calculates the black/white ratio in these areas. If the portion of black colour in a check box exceeds a certain threshold, FormReader will consider the check box selected. FormReader can even recognize corrected marks, i.e. boxes ticked by mistake and then inked over.

ABBYY FormReader 6.0 will reliably recognize not only conventional ticks/checks and crosses, but also completely inked over check boxes if the latter are rectangular in shape or have no borders.



Verification of inked-over check boxes in ABBYY FormReader Desktop Edition.

This feature of ABBYY FormReader has a very important practical application. Suppose someone filling in a form makes a mistake and ticks the wrong box. Instead of taking a new blank form and filling it from scratch, they can just blot out the mark in the check box selected by mistake and put a new mark in the right check box. FormReader will treat the inked-over check box as a mistake and consider it to be unchecked. This method may also be used when recognizing text fields.

Automated forms processing: step by step

Where data capture should be used?

There are numerous situations when automated forms processing is the only right solution. Here are some possible scenarios.

- **Forms processing is not the main speciality of a company.** Manufacturing or trading companies in most cases don't even have a department responsible for forms processing. Forms, such as order bills, are usually processed manually by secretaries or office assistants. Everything runs smoothly if the company needs to process no more than several dozen forms. But processing hundreds of forms requires additional staff, otherwise there will be long waiting lists and the personnel will be distracted from their main job of communication with customers.

Solution: installing a forms processing application, such as ABBYY FormReader Desktop Edition. The entire system can be placed on one desk and does not require additional staff or maintenance.

- **Processing questionnaires is one of the major business processes of a company.** A good example is a marketing agency that collects and analyses data. Sometimes such agencies need to process tens of thousands of forms per day. In this case data capture is part of the entire technological process and selecting a data capture solution has its own specifics.

Firstly, the volume of incoming information varies greatly and depends on the customer's needs and the scale of a particular survey. Therefore **scalability** is crucial, so that the agency can easily increase its throughput.

Secondly, in the case of a marketing agency, investing into automated forms processing means investing into means of production. Consequently, **the ROI must be easy to calculate.**

Thirdly, questionnaires will change significantly from survey to survey, and the marketing agency will need a **tool for designing new forms.**

In this case, ABBYY FormReader Enterprise Edition would be the ideal choice. This is a highly scalable solution whose productivity can be increased by organizing distributed forms processing and adding new modules.

- **Converting archives into electronic form.** In most cases this job has to be done only once, but the amount of information to be processed is considerable - a paper archive may take up several rooms. At the same time archive owners often do not have the sufficient administrative or financial resources to hire additional personnel.

In this case time required to install and set up a data capture solution is not so crucial. What is important is its ease of use and efficiency. ABBYY FormReader Desktop Edition would be the ideal solution for archives.

ABBYY offers a very attractive licensing scheme which takes into account the number of pages processed by FormReader. The licence allows you to process a certain number of pages, and once this allowed number has been used up, FormReader becomes inoperable. Next time you need to process a known number of pages you will simply need to renew your licence. This approach is particularly suitable for situations similar to the one described above.

It is quite possible that your company needs to solve a similar task. But how do you set about choosing the right solution? Where do you start?

Designing a form

First of all you have to design a form. You need a form that is both easy to fill in and to process. The design is crucial because any mistakes made at this stage may drastically reduce the speed of processing. Be sure to follow the recommendations of the supplier of your data capture application.

To create a form, you first need to think out its logical structure, then design it, and, finally, draw your form. A detailed treatment of each stage follows.

Determining the form's logic

Forms with carefully thought-out logical structures are easier to fill in and process. You need to decide what data you will need to gather and draw up a list of required data fields. Be sure to discuss these fields with your clients.

Next, you have to determine such parameters as the size of the form and the number of pages in the form. If you decide to change these parameters later, you may need to make considerable changes in the setup of your data capture program. This is why we recommend drawing all your sketches on sheets of paper the size of your future form so that you have enough space for all the elements you wish to place on the form.

ID fields (identifiers). If you need a form which consists of several pages, be sure to introduce elements which will help you avoid confusion. Usually each page is provided with an ID field which is the same for all the pages of the same kind. The nature of the ID field depends on the nature of the form. This can be an SSN, a customer's ID, the code of the project, etc.

Simple and complex fields. Try making your fields as simple as possible. This will make for fewer errors when filling in and processing the form. The more predictable the words or numbers entered into particular fields, the higher the recognition rate. It is best to split such fields as Name, Date, Telephone (area code + city code), Address (country+city+street) into several subfields.

Free space is a rare commodity on any form, therefore if you know the maximum length of a field do not make it larger than necessary. This will prevent the person filling in the form from entering redundant information and will make the whole completion process more self-evident. Examples of data fields with a known number of character spaces: SSN, postal code, abbreviations for US states, local telephone numbers, standard names of currencies.

Field length. The length of words in such fields as Street, Second Name or E-mail is difficult to predict, therefore you need to provide some extra character spaces "just in case". If you think one line may prove insufficient, allocate two or more lines to such fields. FormReader can logically merge such lines into one field without diminishing recognition quality.

Separators. The form must encourage people who fill it in to enter only meaningful information into its fields. For example, it would be wise to design a Date field in such a way that users do not have to enter separators (e.g. slashes, hyphens or dots) themselves, because they will be printed on their forms. This will greatly increase recognition accuracy. Similar examples: pre-printed hyphens in SSN and ISO fields, the first three digits of the current year, etc.

Check boxes. If possible answers are known in advance, it is best to use check boxes instead of text fields, as OMR algorithms are much more reliable than ICR. For example, do not ask users to fill in their marital status in text fields by writing such words as

"married", "single", "divorced" or "widower". Instead, print the possible answers on the form and ask users to tick the appropriate box.

Captions and photos. If your form will include such fields as Signature, Seal, Photo or Fingerprint, be sure to provide enough space for these fields. This will reduce the number of corrections and increase recognition quality. Don't forget that affixing a stamp or putting a seal on a form may result in blots on the reverse side which may impede the recognition of the text there.

Selecting form type and design

One of the major recognition tasks is to separate the contents of the fields from the field boundaries. The success of this task largely depends on selecting the right type of form. Remember that colour drop-out forms provide the best results. Users will enter information into white rectangles and the scanner will later remove the background. The general rule of thumb: use grey forms when-ever you cannot print colour drop-out forms.

When designing your forms, pay particular attention to reference points and ID fields - this will help you get the most out of automated forms processing.

What is a reference point? FormReader uses reference points to match forms with their templates. Reference points are also used to correct linear distortions introduced by scanning and to detect the location of the fields on the form. Sometimes reference points are referred to as anchors. Examples of reference points: black squares, corners, crosses, captions that do not disappear during scanning, vertical and horizontal lines. We recommend placing three or four reference points in the corners of the page. This will enable FormReader to match forms with their templates and to process similar forms printed on different printers or sent in by fax.

What is an identifier? Identifiers are form elements that do not disappear during scanning and that are used to match a form with its template. If multiple-page forms are processed within one batch, you need to provide a unique element on each page which will be used to identify pages as belonging to a particular form. We recommend using bar codes, form titles or additional black squares as form identifiers.

Drawing a form

Once the logical structure of the form has been arrived at, you need to draw your form. What drawing tool should you use? Currently there are several tools available on the market.

If your designer is familiar with CorelDRAW or Adobe Illustrator, they may draw the form in one of these applications. These are good professional design tools but they have their drawbacks - both programs are a bit too "heavy" and expensive. They are too difficult for a novice to learn, and learning all their features will take considerable time.

Microsoft Visio is more common and less difficult to use. Even though it is mainly intended for drawing charts and graphics, it can also be used to draw quite attractive forms. The easiest way to design a form in MS Visio is to use its so-called template galleries. You can obtain a template gallery containing various form elements form ABBYY. MS Visio can be used to create professionally-looking colour drop-out forms which can then be printed on a laser printer.

As a last resort, you can create forms in Microsoft Word. Since MS Word was not originally intended as a tool for designing forms, drawing a form in this text editor may prove a real challenge.

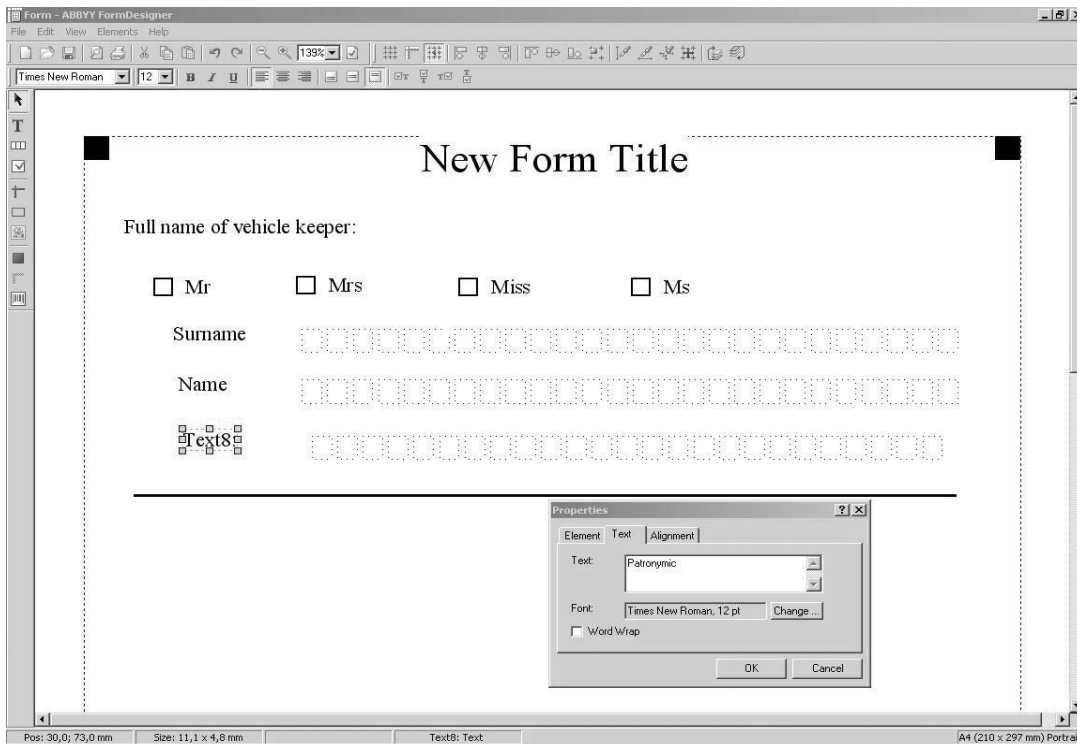
FormReader also includes a very handy from-drawing tool.

FormDesigner is a form-creation application provided with each copy of FormReader. This is a simple and efficient form drawer that will help you draw even the most sophisticated forms.

All forms include certain typical design elements: titles, black squares, text fields, check boxes, etc. FormDesigner is a WYSIWYG form editor that provides you with a set of ready-made elements which you can edit and adjust to suit your needs. Just click on one of the elements and place it where you want to see it on the form. You will be able to start creating your forms right away because you won't need to find graphic primitives first. Once you have designed

your form, FormDesigner will create an *.xfd file which will include all the relevant information for your template. When setting up FormReader to process your form, simply import this file and you will get a template almost ready for use. All you will need to do is specify the properties of the already marked fields, adjust reference points and add validation rules if required.

The next step is to set up FormReader so that it can capture data from your forms.



Drawing a form in ABBYY FormDesigner.

Setting up FormReader

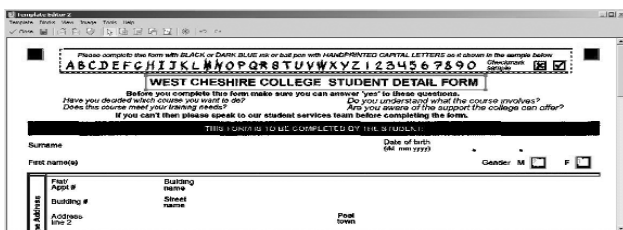
When you are setting up FormReader to capture data from a particular kind of form, you are "telling" the program where to look for data fields and what "hints" are available on printed forms. Setting up the program correctly is just as important as designing the form.

Creating a form template. Below follows a brief treatment of all the steps you need to perform in order to create a form template.

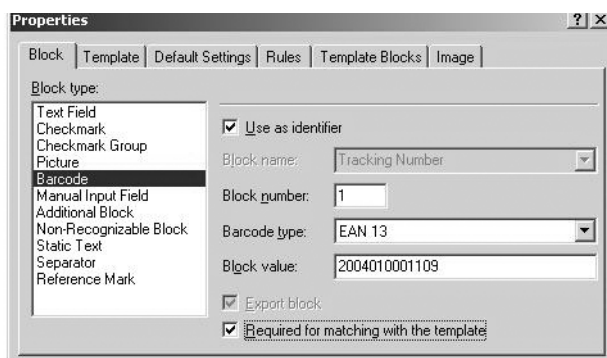
1. First, you must obtain an image of a blank form. You can either scan a blank form or use an image file obtained from any other source. If the form was created in ABBYY FormDesigner, simply import the *.xfd file created in FormDesigner. This template already includes all the required blocks.

If you do not have an *.xfd file, you will need to scan a blank form and follow steps 2 to 8.

2. Mark out reference points and identifiers. These blocks can be marked out either manually or automatically. Sometimes unchangeable text or bar codes are used as identifiers.

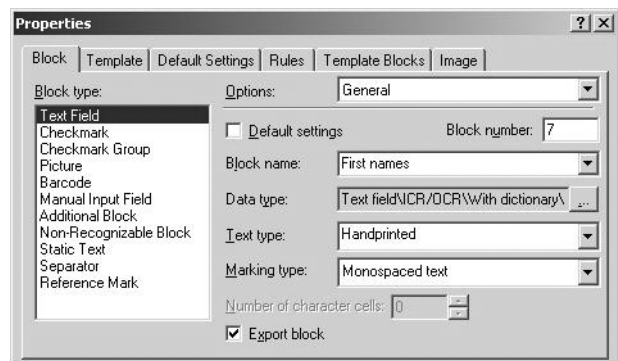


Specifying reference points and identifiers in form template designer.



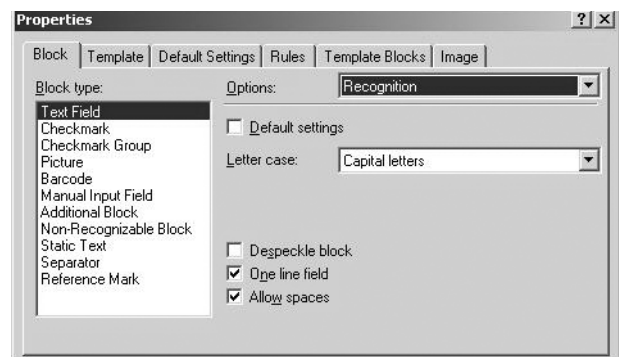
Specifying barcode as a form identifier.

3. The next step is to test your template to make sure that it matches the original form.. Try placing the template on the form to see whether the reference points and identifiers enable the program to match the form and its template correctly.
4. Mark out data fields. Use your mouse to draw blocks around those fields from which data must be captured.
5. Now you need to specify the properties of the fields, i.e. tell the program what kind of field borders are used and what kind of information will be entered into these fields. To optimise this process, we recommend first analysing the template and determining the properties that are common to most of the fields. You can then specify them as default properties for all blocks.



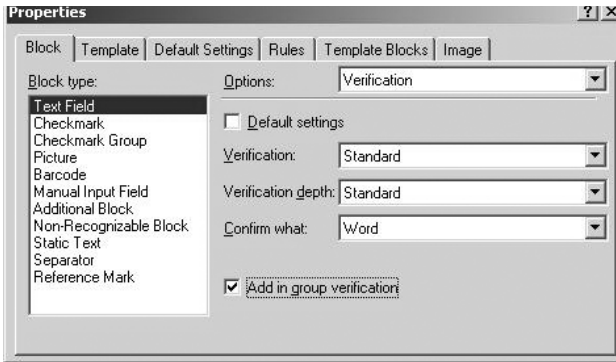
Specifying field properties in form template designer.

6. Next you need to add validation rules. These are the conditions that the data entered in the fields must satisfy. The program will use these rules to validate the format of the data and to normalize them if required (e.g. the program may convert the dates into a specific format). Rules can also be used to compare entered data with similar data in a database to make sure, for example, that sums written in figures correspond to the same sums written in words.
7. The right parameters under Recognition and Verification may also have a great impact on the quality of recognition..



Specifying recognition options.

8. If the captured data are to be exported to a database using an ODBC connection, the latter must also be set up in the Template editor.
- Before you can start capturing data you must also select a scanner.



Specifying verification options.

Selecting a scanner

Choosing the right scanner is important because scanners have a direct impact on the speed and quality of processing. It should be noted that if you need to process more than 100 forms per day common flat-bed scanners will not do. Such scanners are widely used in offices throughout the world to digitise photos and documents, but they are not suitable for industry data capture, because they are too slow and have a small in-service lifetime. After scanning 1,000 pages the lid of a flat-bed scanner may just fall off!

To scan large numbers of forms quickly and reliably you need a special scanner. Here is a list of scanner features to look for:

- **Paper format.** Usually forms are scanned using A3, A4 and A5 scanners.
- **Resolution.** Forms must be scanned at 200-300 dpi and all scanners support this resolution. Higher resolutions will inevitably slow down the entire process.
- **Duplex scanning.** Many projects require scanners that can run in both duplex and simplex modes and scan either in black-and-white or in colour. The latter is required when removing colour seals from images or when saving colour photos from questionnaires.
- **Automatic Document Feeder (ADF).** This device allows you to load batches of 25, 50, or 100 documents into the scanner. This is a must have, otherwise the operator will spend 90% of his time feeding paper documents into the scanner.

- **Throughput.** Very often the overall speed of processing depends on the speed of scanning. In terms of throughput, scanners can be divided into the following three groups: office low-throughput scanners, office medium-throughput scanners, and production scanners with very high throughput. Low-end office scanners have a throughput of up to 500 pages per day while production scanners may scan more than 20,000 pages per day.
- **Page feeder.** If the scanner accidentally takes in two form pages at a time, this may result in some of the pages not processed at all. To prevent this from happening, many scanners have a special control mechanism which weighs paper sheets, measures their thickness or measures the light that may penetrate them. But these methods do not work if the forms are not homogeneous (i.e. printed on different kinds of paper, have different colours and paper thickness, etc.). The solution is to use ultrasonic sensors which make sure that the signal has been reflected only from one surface, i.e. from one page.
- **Additional features.** Some scanners may have a number of additional features which may also come in useful
 - An endorser or a built-in printer that prints on scanned documents an index that is then used to identify them;
 - A hardware image enhancement module;
 - A hardware image compression module;
 - Colour lamps that can remove certain backgrounds (so-called drop-out colours - red, blue or green);
 - Caching images in the scanner's onboard memory, which makes for faster scanning.

Personnel training

Working with ABBYY FormReader requires minimum special knowledge and training. The data capture system is usually run by operators responsible for entering data from forms and an administrator who sets up and monitors the system.

- Depending on how data capture is organized, the operator's job can be of two kinds
 - all operations are performed on one computer, the operator loads forms into the scanner, and oversees the scanning, recognition, and verification processes;
 - in the case of ABBYY FormReader Enterprise Edition, different operators are responsible for specific processes - scanning, document assembly, verification, and export.
- The program is set up by an administrator. In the case of ABBYY FormReader Enterprise Edition, the administrator deploys the system, allocates processing roles to operators, creates templates and descriptions of multi-page documents. The

operator also oversees the flow of information within the system

It takes from several hours to 2-3 days to train the operators and the administrator*.

All the practical knowledge required to operate FormReader can be acquired within this time period thanks to a carefully thought-out training course

* The administrator's training course includes the following topics :

- 1) production capture;
- 2) designing new forms;
- 3) creating form templates;
- 4) installing ABBYY products (including network installations);
- 5) setting up scanning, recognition and verification options;
- 6) allocating processing roles to operators;
- 7) creating validation rules and document assembly rules;
- 8) monitoring the operation of the program and creating reports.

Processing cycles

For a better understanding of how ABBYY FormReader works, let us take a closer look at the main processing cycles.

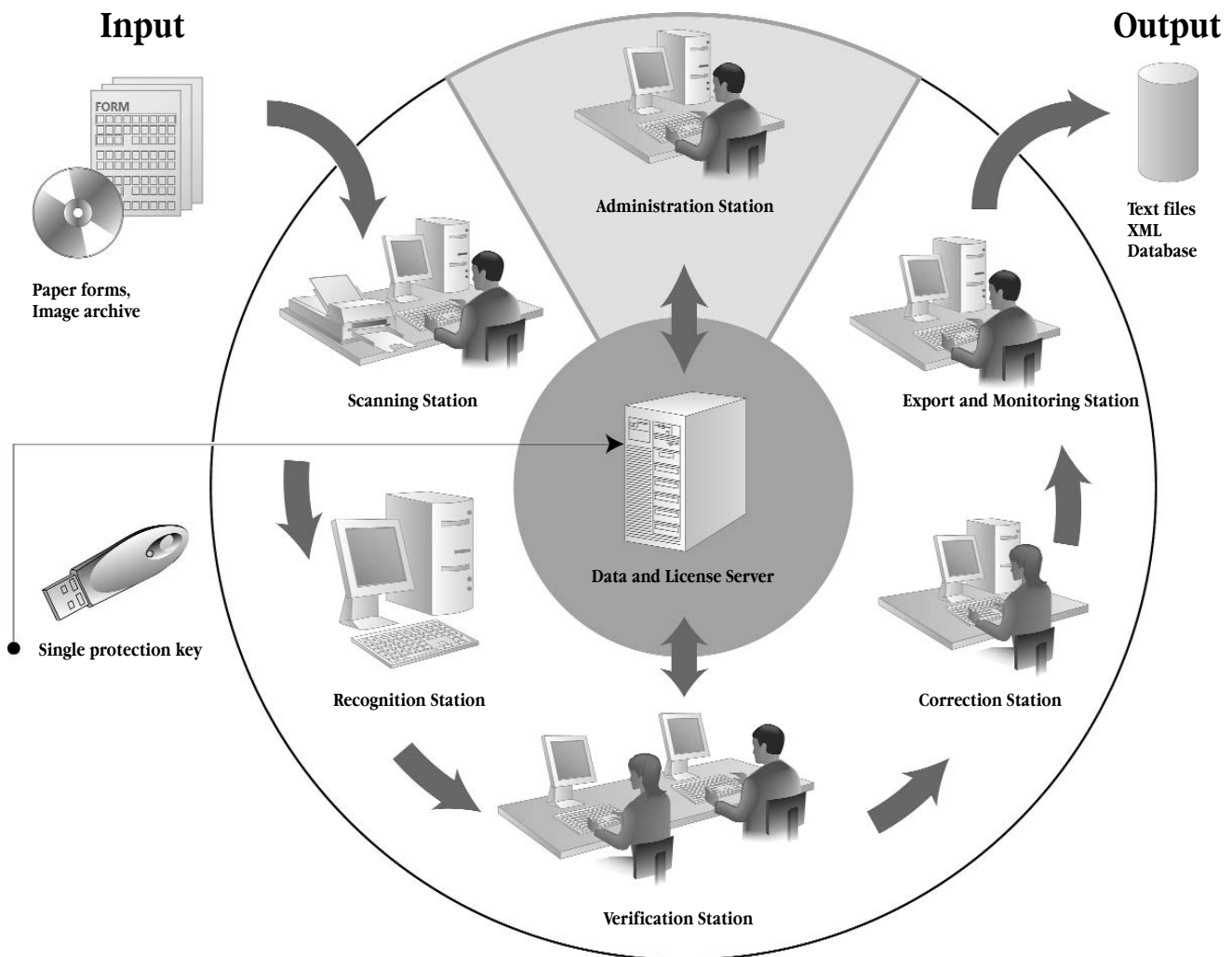
1. **Creating a batch.** A batch is a collection of similar documents which must be processed and saved both as images and as text data captured from the fields. Batches can be opened either by an operator or automatically by the program.
2. **Adding images to a batch.** Images of forms that need to be processed may be added to a batch in one of the following ways:
 - by scanning paper forms;
 - by adding pre-scanned images from a special dialog box;
 - by dragging and dropping document icons in
 - Windows Explorer.
3. **Recognition.** Recognition is an automated process whereby the text in the data fields is "read" by the program and converted into electronic form. First, the program selects the right template for the form and detects blocks from which data have to be captured. Next, the block images are converted into electronic text.

4. **Validation and verification.** Once all the images in a batch have been recognized, some pages may contain characters about which the program is unsure. These pages are passed on to the operator for verification. The verifier either confirms the characters or corrects them. Similarly the operator corrects any errors detected by validation rules (the program marks pages with errors with special colour flags).
5. **Export.** Finally, verified and validated data are saved to a file or exported to a database. All the operator needs to do is to click the "Export" button.

Throughout the entire data capture process the involvement of the operator is kept to a minimum. More importantly, the operator's actions are strictly circumscribed, which greatly reduces the chance of errors. Therefore, automated forms processing is not only much faster than manual data entry but produces much more accurate results. The quality of resulting data is paramount and the following section describes various mechanisms used in ABBYY FormReader to ensure the high quality of captured data.

Below you can see a chart showing how forms are processed in ABBYY FormReader Enterprise Edition.

- There are two streams of data - an input and an output stream.
- Each operator is responsible only for one processing stage, e.g. scanning and registering images in the system.
- The operators handle data as if they were working at an assembly line.
- If there are not enough operators responsible for a particular stage, their number can be easily increased
- All the data and settings for all the modules are stored in one place.
- A protection key is plugged into the server and is used to protect the entire system..



Processing forms in ABBYY FormReader 6.0 Enterprise Edition.

Ensuring the quality of data

Defining data quality

In the previous sections we have often used the phrase "quality of data". By the quality of data we mean the completeness and accuracy of captured information. The higher the correspondence between the data exported into the database and the data entered into the fields of the paper forms, the higher the quality of data.

The quality of data is the correspondence of the data entered into the target system to the data entered into the fields of the paper forms. The quality of data is one of the most important parameters of a forms processing application.

The following factors may have an adverse effect on the quality of data:

- **Sloppy writing.** If someone writes carelessly, makes corrections or merges some letters, the chances of recognition errors will increase. There is an obvious remedy: when designing a form make sure that there is a separate character space for each letter and digit on the form and that complex fields are broken down into simpler ones, which are easier for the program to handle. Follow the recommendations given in "Developing the Logical Structure of the Form", and sloppy writing will have a minimal impact on recognition accuracy.
- **Typos.** When entering data from forms manually, typos are an important factor. Keyers will inevitably get tired and make more mistakes towards the end of the day. The only solution is to give up manual processing altogether. Operators of automated data capture systems experience much less strain, and even if they do get tired this will have almost no impact on the quality of the resulting data - ABBYY FormReader will use validation rules to ensure data integrity. Even if an operator

makes a mistake, the program will easily detect it and alert the operator.

- **Recognition errors.** When reading information from the fields, the program will mark some characters as "uncertainly recognized". These will be passed on to the operator for verification. But suppose the program is too self-confident about some characters, even though they have been recognized wrongly. They would not be submitted for manual verification and incorrect data would be exported into the database. This is the bane of all data capture applications, but ABBYY developers have successfully tackled this problem of "hidden" errors. Tests show that chances of error are as low as 0.5% for letters and 0.1% for marks in check boxes.

To sum up: FormReader has special methods and techniques to ensure the high quality of data. These include

- image pre-processing;
 - data type checks;
 - data verification;
 - data format checks;
 - validation rules;
 - document assembly rules
- (in ABBYY FormReader 6.0 Enterprise Edition).

Image pre-processing

Very often form images will contain "garbage" in the form of excess dots introduced by scanning. Sometimes an image may be skewed or rotated by 90 degrees from its normal orientation. It is very important that the influence of such external factors be minimized. ABBYY FormReader can do the following:

- despeckle images, i.e. remove excess dots that hamper recognition (the size of the dots to be removed can be adjusted);
- deskew images that have a skew angle of up to 10 degrees;
- rotate images by 90 degrees;
- invert images, i.e. turn black pixels into white and vice versa.

The program can also detect textured backgrounds consisting of dots or lines that are much thinner than the characters to be recognized. FormReader will remove such textures before it starts analysing and recognizing the text. Excess dots will be removed during pre-processing, and grids of hair-width lines will be detected and removed when analysing the structure of the document.

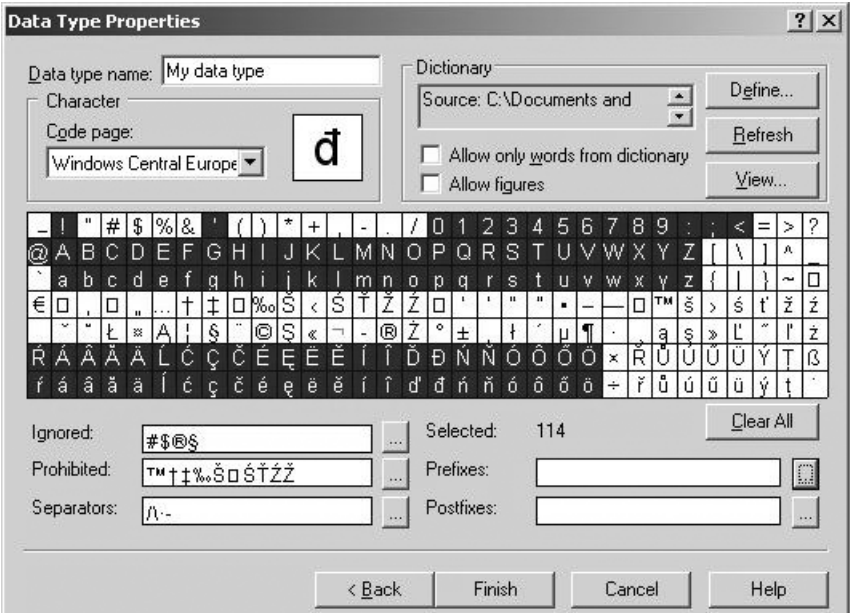


An image with a textured background.

Data type checks

Even before submitting data for verification, ABBYY FormReader 6.0 will **check the recognized data against dictionaries and user databases**. Suppose your questionnaire has a field captioned "Your favourite brand of cheese". You can create a dictionary of cheese brands and use it to facilitate recognition. Dictionaries can be created for any data types to help the program more readily recognize the information entered into the fields.

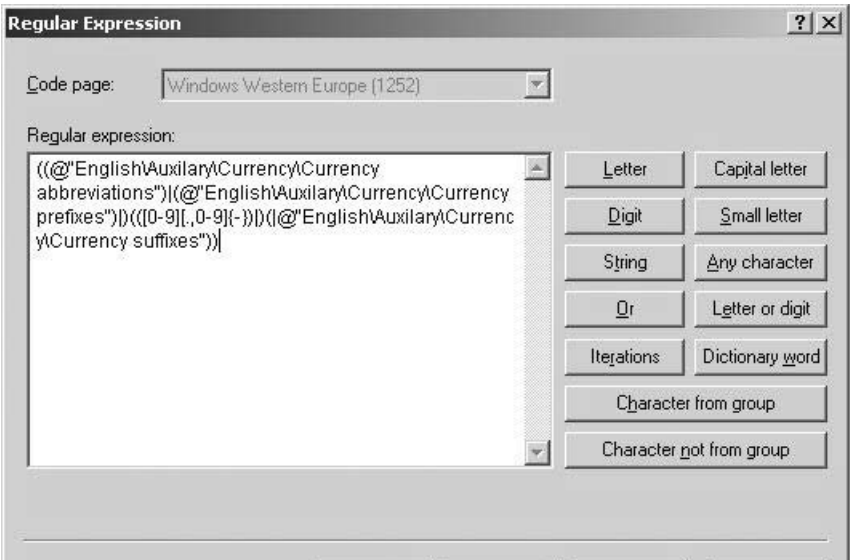
ABBYY FormReader 6.0 already includes multilingual dictionaries for standard data types ranging from proper names and cities to currencies and postal codes. Of course it is impossible to cover all possible areas of human activity, but users can create their own dictionaries and associate them with the corresponding fields.



Creating a user's data type and connecting a user's dictionary in ABBYY FormReader 6.0 Desktop Edition 6.0.

Together with dictionary-defined data types, FormReader makes extensive use of regular expressions. Regular expressions describe the possible combinations of characters and their mutual positions. For

example "c*t" describes all three-letter words staring with a "c" and ending with a "t" - cat, cut, cot, etc.



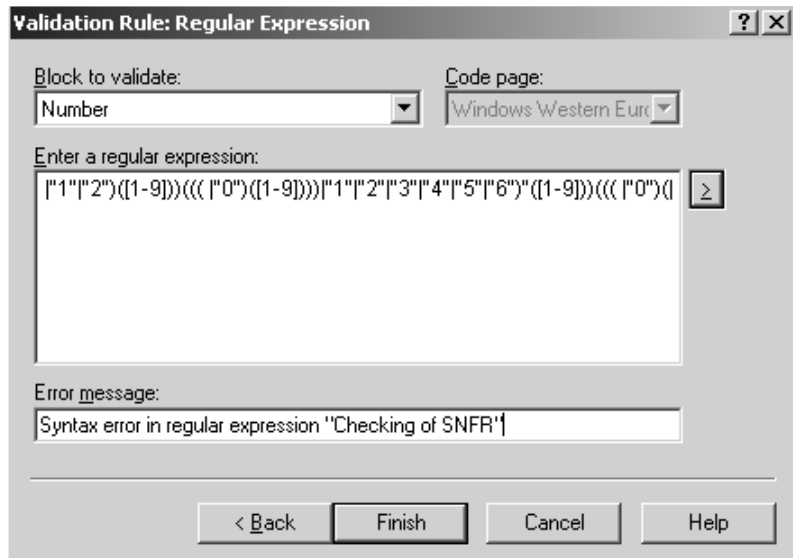
Adding a data type defined by a regular expression in ABBYY FormReader 6.0 Desktop Edition.

Data format checks

Once FormReader finishes recognizing the data, it will check whether the results conform to the format specified in the template. Let us take a closer look at this type of check using a Serial Number field as an example. Suppose the serial number must have the following format:

SNFRn-nn-nnn-nn, where n is a digit

When creating the template, the user can write a **regular expression** that describes this format:



Validation Rule: Regular Expression

Block to validate: Code page:

Enter a regular expression:

```
"1" "2" ([1-9]) ((( "0" ) ([1-9]) )) "1" "2" "3" "4" "5" "6" ([1-9]) ((( "0" )
```

Error message:

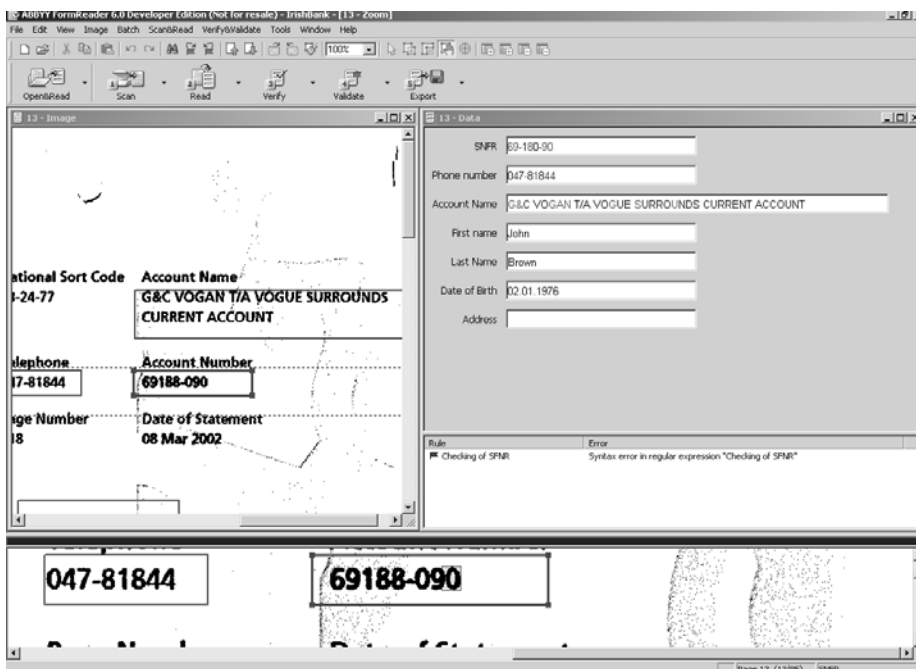
Syntax error in regular expression "Checking of SNFR"

< Back Finish Cancel Help

A regular expression used to check the format of data.

FormReader will use this regular expression to make sure that the data entered into the Serial Number field conform to the set format. Those pages on which the regular expression reports an error will be marked with a warning flag. This approach helps to detect such tricky errors as "o" instead of "0", "l" instead of "1", etc. There are

several gradations of flags - some serve to alert the operator, others will prevent the form from further processing until the operator has made the necessary corrections.



FormReader 6.0 Developer Edition (not for resale) - IrishBank - [13 - Zoom]

File Edit View Image Batch ScanRead VerifyValidate Tools Window Help

OpenRead Scan Read Verify Validate Export

1.2 - Image

1.2 - Data

SNFR: 69-180-90

Phone number: 047-81844

Account Name: G&C VOGAN TIA VOGUE SURROUNDS CURRENT ACCOUNT

First name: John

Last Name: Brown

Date of Birth: 02.01.1976

Address:

ational Sort Code: 1-24-77

Account Name: G&C VOGAN TIA VOGUE SURROUNDS CURRENT ACCOUNT

Telephone: 047-81844

Account Number: 69188-090

Page Number: 18

Date of Statement: 08 Mar 2002

Rule: Checking of SNFR

Error: Syntax error in regular expression "Checking of SNFR"

047-81844 69188-090

Page 13 (13/99) SNFR

An error has been detected on this page - the date has the wrong format.

Controlling logic

Very often certain restrictions apply the data to be entered into the fields. For example, numbers may have to belong to a certain interval. This can also be checked by validation rules, and if the recognized data do not meet the imposed requirements, the rule will report an error. Here are some examples of such validation rules:

- **Normalize and check dates.** This rule

- (a) validates the date,
- (b) converts the date into a predefined format,
- (c) checks that the date falls within a predefined time period.

The first step will discard such dates as 32/45/199, the second step will arrange the figures in the required order, e.g. dd/mm/yyyy, and the third step will make sure that the date falls within the specified interval.

Validation Rule Properties

Rule type:

Rule name:

Rule status: ☒ Error ☐ Warning

☒ Don't verify blocks if validation has been successful

☒ Verify blocks if validation has been unsuccessful

Apply rule

☐ Always

☒ Always, but not to edited blocks

☐ After recognition, when saving the verification results

☐ Don't apply rule if all blocks to validate are empty

Validation Rule Properties

Rule type:

Rule name:

Rule status: ☒ Error ☐ Warning

☒ Don't verify blocks if validation has been successful

☐ Verify blocks if validation has been unsuccessful

Apply rule

☒ Always

☐ Always, but not to edited blocks

☐ After recognition, when saving the verification results

☐ Don't apply rule if all blocks to validate are empty

Date validation rule.

- **Check sums.** If there are several digits on the form and a field contains their sum total, we may add an additional validation rule that will make sure that the recognized figures add up.

Validation Rule Properties

Rule type:

Rule name:

Rule status: ☒ Error ☐ Warning

☒ Don't verify blocks if validation has been successful

☐ Verify blocks if validation has been unsuccessful

Apply rule

☒ Always

☐ Always, but not to edited blocks

☐ After recognition, when saving the verification results

☐ Don't apply rule if all blocks to validate are empty

Validation Rule: Check Sums

Blocks to add up:

- Article 1
- Article 2
- Article 3
- Delivery costs
- Service

Available blocks:

- Article 1
- Article 2
- Article 3
- Company
- Delivery costs
- Name
- Other

Type of comparison:

Comparison block:

☒ At least two non-empty items to add up

Check sum validation in ABBYY FormReader.

- **Normalize prices.** The price function automatically converts the price into the required format, e.g. 12.90 or 12,90 (Russian style). If the recognized data cannot be converted into the required format the program will report an error.
- **Conditional checks.** The user can use a special language, which is very much similar to programming languages, to specify certain conditions and actions to be performed by the program if these conditions have or have not been met. Below follows a sample condition that reports an error if the fields Country and City are empty.

```
If [City]. IsEmpty() then Error ([City],  
No City Name is signed  
else  
  
If [Country]. IsEmpty() then Error  
([Country],  
No Country Name is signed.  
else TRUE
```

- **Sum in figures - sum in words.** This rule compares the recognized number written in digits with the recognized number written in words. This rule works only for whole numbers in Russian-language documents.
- **Automation checks.** OLE Automation checks enable the user to create their own validation rules, however complex.
FormReader provides various means to control the logic of recognized data. Any errors that might slip in will be detected and reported to the operator. Critical data will not be exported until reviewed and corrected by the operator.

Processing multi-page forms

When processing multiple-page forms, pages of one form may get mixed with the pages of another form. This is a serious problem which may compromise the quality of the resulting data. To eliminate this sort of errors, multiple-page forms may be provided with a special field that contains a unique ID. All the pages of one form must be marked with the same ID. The program will use this ID to identify the pages as belonging to a particular form during document assembly.

ABBYY FormReader Enterprise Edition can process forms containing multiple pages. In ABBYY FormReader Enterprise Edition, document assembly means collecting data from separate pages of the same form and uniting them into an informational entity whose structure corresponds to the structure of the original paper form.

A good example of identifier used in processing large numbers of multi-page forms is INN (individual tax payer's number issued to all Russian tax payers). INNs are placed on each page of a tax return, which makes it possible to identify all the pages and assemble them into one document.

If, for some reason, the document assembly rule reports an error, the document will be passed on to the Verification Station. The operator will replace the pages, add the missing pages, asks for a re-scan of the document, etc., and then run the validation rules once again.

Operator stress as an important quality factor

One of the main advantages of a data capture application is that it relieves operators from the repetitive and monotonous task of typing data manually. ABBYY FormReader operators spend 90% of the time verifying data. They do not have to divide their attention between the keyboard, the screen and the paper forms. Nor do they have to keep in mind endless database columns into which this or that figure is to be entered. All the logical relationships between the data are established and checked at the preparatory stage.

But the most striking effect is that operators become less stressed, they do not strain their eyes weeding out errors, and they do not have to worry about the deadlines. This makes for better working atmosphere and has a direct impact on the operators' efficiency.

Organizing automated forms processing

If you take into account such factors as the quality of entered data, the speed of processing and the working conditions of the operators, data capture applications are hard to beat. Automated data capture becomes economically viable whenever you need to process 100 forms per day or more.

But even relatively small processing volumes will require certain changes in how the working process is organized. Processing thousands of forms per day is a huge task that will need significant changes in organization and administration.

Approaches to data capture

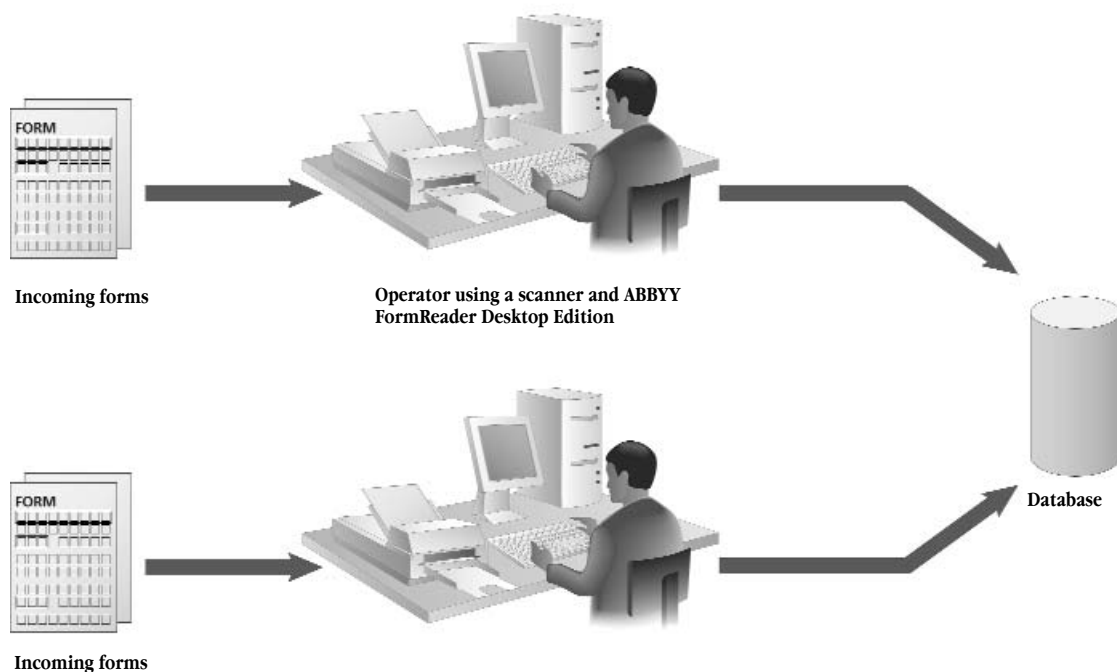
Data capture can be organized in one of the following ways: forms may be processed as soon as they are submitted, or forms may be first collected and then processed in batches. In the former case the data capture solution will be implemented in the front office, and in the latter case it will be implemented in the back office.

Front-office data capture

A good example of the first approach is a data capture solution installed in a warehouse of a trading company. A client wishing to place an order fills in an order form where he states the name and number of the goods he wishes to purchase. This form must be processed as soon as it gets to the staff of the warehouse, who will issue an invoice. The invoice will be paid by the client while the warehouse staff pack the ordered goods. This means that the data capture application must be installed directly in the warehouse. This kind of application will have a number of distinctive features:

1. The speed of scanning in this particular case is not very crucial, because packing and loading goods takes a lot more time than scanning orders. This means that even a flat-bed scanner will do. (It takes modern scanners 30-40 minutes to scan an A4 page, which is quite acceptable in this case.)

All data capture is performed at one desk where the client submits his order form. Even additional operations which are not directly related to data capture may also be performed at this desk, e.g. printing out an invoice, archiving the paper order, etc. In situations like the one above ABBYY FormReader Desktop Edition will be the ideal choice.



Back-office data capture

A good example of the second approach is the processing of tax returns. The Russian Ministry of Taxes has adopted the following system of processing tax returns: for a period of several months tax returns are to be gathered from citizens by local tax officials, after which the collected documents are shipped to a central site for processing. A very powerful data capture solution is required to process huge numbers of tax returns. This solution will have the following distinctive features:

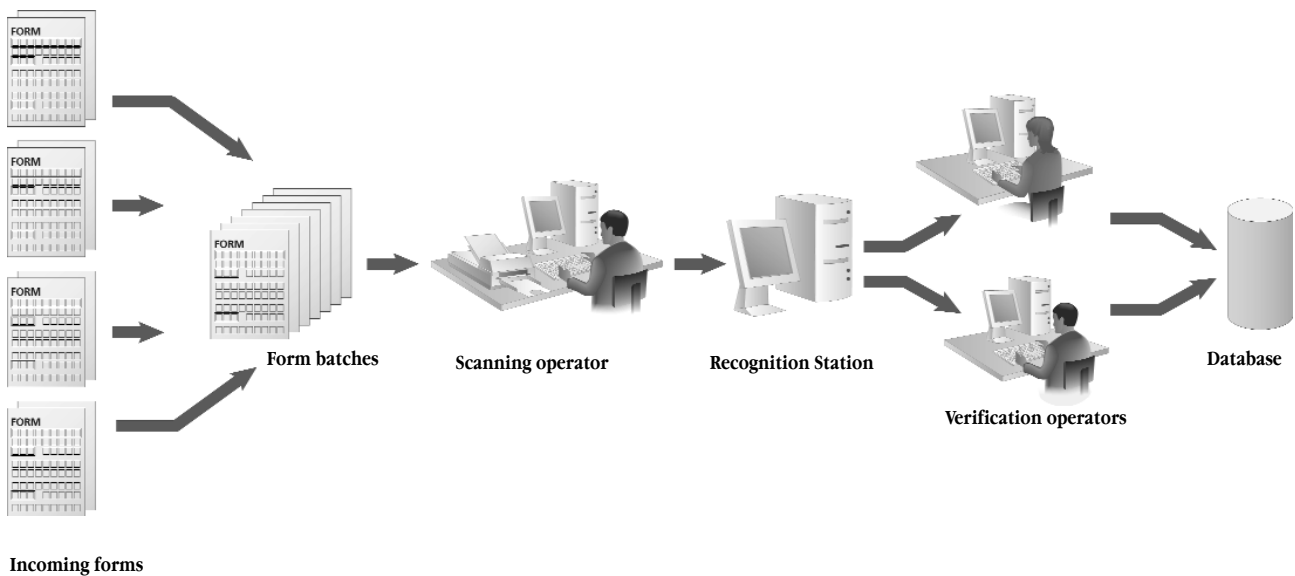
1. High-speed production scanners must be used.
2. A distributed data capture solution is required. Each operator must be responsible for specific operations carried out on only one workstation - scanning, recognition, verification, or export.
3. Much more stringent quality control is required compared to the warehouse example above: you can always make corrections in an order, but it is virtually impossible to make corrections in millions of tax returns.
4. Several working shifts are to be organized for maximum speed of processing. Among other things, this will use the high-speed scanners to their full capacity.

The best solution in this case would be ABBYY FormReader Enterprise Edition.

These two examples show that an optimal data capture solution can be selected for each project, however large or small. When deciding on what forms processing solution to choose, one has to

take into account all the specifics of the project and the capabilities of a particular application. Therefore, the customer and the solution provider must work hand in hand at the project design and implementation stages.

Let us now consider the basic principles of deploying large-scale data capture solutions. These principles are based on our successful experiences in implementing large-scale projects where ABBYY data capture products have been used.



Gathering forms for subsequent processing.

Data capture basics

Batch processing

Batches are collections of forms. Each batch has a unique identifier. An important advantage of this approach is that it structures information streams and facilitates administration, routing and storage of data.

Operator specialization

Production capture solutions operate like assembly lines where each person is responsible for a specific operation. Specialization increases productivity and makes the system highly scalable. You can easily add, say, more Scanning Stations and scanning operators without interfering with the job of the recognition or verification operators.

Scalability

As has been said above, the entire system consists of highly specialized modules whose number can be easily increased or reduced to meet particular processing requirements. Suppose the initial configuration included one recognition module (a powerful two-processor server) and eight verification modules. If verification becomes a "bottleneck", you can easily add any number of verification modules to increase verification throughput. This makes the system very flexible and manageable and saves customers a lot of money.

Processing queues

Batch routing is an important concept in data capture. Batch movement cannot be arbitrary but should be optimised to reflect the logic of forms processing.

Depending on the stage of processing of a particular batch, the system a particular status to the batch. In a complex data capture system several batches can have "verify" status at a given moment. They will be placed into the verification queue, and as soon as one of the Verification Stations is freed up one of the queued batches will be sent to this station. This allows the system to evenly distribute the workload between the stations and operators, so that they do not stand idle or become overwhelmed.

Data flows

To optimise batch processing, batches are routed and placed into processing queues. If a problem occurs with any document (e.g. it was poorly scanned), the problem batch will be immediately taken out of the queue so as not to interfere with the processing of the other batches. As a rule, problem batches are set aside to be processed manually - the operator will have to identify the cause of the problem and select the right solution. In this particular example the document will be sent to the Scanning Station to be re-scanned. It should be noted that the processing of the other batches will continue at the same speed.

Production capture

Production capture has its own specifics. Large-scale projects require dedicated software and hardware, special training for the personnel and careful organization.

Forms processing software

Experience shows that processing more than 3,000 forms per day by more than three operators requires a distributed software solution. Each operator will be able to concentrate on their specific task and deliver the best quality. ABBYY FormReader Enterprise Edition is a distributed data capture solution that can process practically unlimited numbers of forms. Additionally, the program can assemble multi-page documents and validate data using sophisticated rules.

Production scanners

High-speed production scanners are essential in large-scale projects. In theory, an alternative solution might be to distribute scanning between a large number of cheap low-end scanners. In practice this approach never works, as ordinary scanners tend to break down, which entails additional costs.

Hardware

- Almost any PC will do as a Scanning Station, the only critical requirement being sufficient hard-disk space.
- Recognition Stations must be more powerful than Scanning Stations - powerful processors and large RAM are essential. Multi-processor servers are often used. Many data capture programs, ABBYY FormReader included, support multi-processor systems.

- Verification Stations must be equipped with good-quality computer screens to reduce the operators' eye-strain.
- The Export Station must have enough RAM, so that data may be exported in background mode.
- The local area network must have a high throughput since internal traffic can be fairly high - one batch can be dozens of megabytes in size.
- A powerful PC is required for the database server, as FormReader stores all settings and entered data in an internal database.

Export of data

Unlike desktop systems, large enterprise systems usually do not save data to files. Very often a special export module is created which exports data into an external data storage or management system. An alternative method of exporting data is to save them into an XML file which can then be processed by a specially created XML analyser. All ABBYY FormReader products support saving in XML.

Personnel training

The staff who will run and administer the system have to be specially trained. Even though ABBYY FormReader's interface is fairly straightforward, some preliminary instructions are necessary. ABBYY specialists will always provide expert advice and training for operators when implementing large-scale projects. Hands-on training is particularly effective and allows operators to start processing forms within a very short time.

Using ABBYY Technologies to Solve Untypical Tasks

Sometimes ABBYY products may look like unlikely solutions for some not very typical or unusual tasks. Indeed, why select FormReader for processing forms, say, in Portuguese if FormReader does not support this language? However, FormReader can be effectively used even in cases similar to the one above.

What if FormReader does not support a required language?

Suppose FormReader does not support the language you need to recognize, e.g. Korean or Thai. Or forms are completed using cursive script and all letters are glued together. These situations have one thing in common: the program cannot recognize the letters in the fields, but the operator can read them. The following solutions can be recommended:

1. When designing a form, **use as few text fields as possible**: replace them by check boxes or groups of check boxes. Explanations to the check boxes (i.e. the wording of choices) can be in any language since the program will not have to recognize them.
2. Use **digital fields** and **bar-code fields** - in most cases data can be captured from such fields automatically.
3. Use the **Key From Image (KFI)** technique. In ABBYY FormReader 6.0 this option is called "manual input fields", because the operator enters data in such fields manually. The program will display the images of data fields, so that the operator may read the information from these fields and enter it into a database. Note that this does not diminish the other important advantages of automated forms processing:

- All the relationships linking the data fields and the fields in the database have been carefully thought out in advance and are defined in the form template - the operator does not need to rack his brains trying to find out in which field to enter this or that piece of information;
- The operator does not have to divide his attention between the paper document, the keyboard and the computer screen;
- Data can still be validated using automatic validation rules, however complex and sophisticated - it makes no difference to the program whether the information to be validated was captured automatically or typed in manually.

Therefore, using KFI or manual input fields is still much more convenient and produces more accurate results than pure manual typing.

우체국보험 청약서				(우체국 제출용)			
청약사항				증서(청약)번호			
보험종류	어깨동무2종 코드(5123)			구분	보험가입금액	보험료	
보험기간	()년, (80)세, ()종신			주계약	1,000 만원	19,700 원	
납입기간	(5)년, ()세, ()월시납			특약	만원	원	
수금방법	2	①방문 ②창구		특약	만원	원	
		③우체국어체 ④은행어체		특약	만원	원	
연금개시연령	()세			합계보험료		19,700 원	
부활계약				부활			
부활국	(국기호)	수금자	(코드)	부활기간	~	부활보험료	원
부활이자	원						
단체계약	단체번호	신규여부	①신규 ②추가	※ 일반단체의 경우 별첨 "일반단체 피보험자 청약명세서"에 기재하여 주십시오.			

Even if FormReader does not support the language of the document to be recognized, it nevertheless can greatly speed up data entry.

Sample Korean form.

Remote scanning and processing faxed forms

Images to be processed may be received from sources other than the scanner. If, for some reason, you cannot deploy FormReader directly at the location where the forms are completed and gathered, you can use remote scanning or gather forms by fax.

Imagine a situation where a survey is being conducted in several cities. The central site is located in the regional administrative centre, and you wish to avoid shipping forms to the central processing site. ABBYY FormReader can easily solve the problem. Forms will be scanned locally, saved as image files (e.g. in black-and-white TIFF format each file will be from 10 to 100 KB in size) and sent by e-mail to the central site. In the processing centre, the images will be passed on to the Scanning Station where they will be automatically grouped into batches.

Alternatively, forms may be sent in by fax. Even though faxing may introduce linear distortions such as stretching or compression, FormReader will correct the images and process them. The only requirement is that faxed forms have reference points in the form of black squares placed in the corners. The program will use these black squares to match the forms with their template. If the position of black squares shifts slightly during faxing, FormReader will adjust the template accordingly, so as to find all the data fields on the slightly distorted forms.

Distributed verification

Verification does not require much skill on the part of the operator. All that is needed is a PC (its processing power is not crucial) and attention to detail. Very often home-workers are hired as verifiers. In this case the only requirement is that the verifier has an internet connection.

If this approach is used, ABBYY recommends using terminal access. A terminal server (TS) is installed in the processing centre, and the home-worker's PC is used as a terminal client (TC). The TS runs a program that manages the flow of data to be verified. The verification operator can see everything that happens in the

screen of the client as if he were sitting in front of the server console. The TC application intercepts the operator's actions and reports them to the TS. In its turn, the server-side program performs the operator's commands. The results are then displayed on the TC.

ABBYY FormReader 6.0 products have been tested under Microsoft Terminal Services.

Processing "flexible" forms

Previously we divided all forms into two large classes - those with rigidly structured fields and those with "flexible" or "floating" fields.

Structured forms are best processed using form templates, whereas unstructured forms require a different approach. ABBYY has developed a method for capturing data from unstructured forms that delivers the same level of quality as obtained with structured forms. The method consists in analysing the "logical" structure of unstructured forms and is called FlexiCapture. The program will analyse the form and determine the location of fields and the type of data in these fields relying on the information about the form's structure provided by the user. Suppose you want the program to find the SSN fields on the forms and capture information from these fields. In the template you will specify that the letters SSN are located to the left of the field, and the field itself consists of a particular number of character spaces (the number of digits in an SSN is known in advance). The program will easily find these letters on the forms and read information from the fields in question. This method of analysing geometrically indeterminate document structures is called Intelligent Field

Recognition, or IFR.

ABBYY has developed a language that can be used to describe the structure of almost any form currently in use. This is a much more powerful tool than the traditional templates for structured forms. ABBYY's FlexiCapture technology is currently used by more than 500 Russian banks to capture data from payment documents.

ABBYY FlexiCapture Studio is a powerful data capture application that allows users to create form descriptions without writing program code. Any advanced user can master the program's interface and learn to create form descriptions.

The user "teaches" FlexiCapture Studio to look for particular fields on the form. To achieve this goal, a formalized description of each field is created in FlexiCapture Studio - fields are described through their various parameters and surrounding objects, such as captions, images, separators, blank spaces, etc. The program will use this description to find the best match. This description is tested on a large number of forms, and then adjusted and entered into ABBYY FormReader or ABBYY FineReader Engine as any template.

20013540

12915 - P 3656 - P

MORTGAGE

THIS MORTGAGE is made this 3rd day of JULY, 2001 between the Mortgagor, JEFF GASKINS AND MELODY GASKINS, HUSBAND AND WIFE HUSBAND AND WIFE

and the Mortgagee, MILLENNIUM BANK, N.A., NATIONAL BANK a corporation organized and existing under the laws of VIRGINIA whose address is 1601 WASHINGTON PLAZA, RESTON, VIRGINIA 20190

(herein "Borrower").

(herein "Lender").

WHEREAS, Borrower is indebted to Lender in the principal sum of U.S. \$ 90,000.00 which indebtedness is evidenced by Borrower's note dated JULY 3, 2001 and extensions and renewals thereof (herein "Note"), providing for monthly installments of principal and interest, with the balance of the indebtedness, if not sooner paid, due and payable on AUGUST 1, 2016;

TO SECURE to Lender the repayment of the indebtedness evidenced by the Note, with the interest thereon; the payment of all other sums, with the interest thereon, advanced in accordance herewith to protect the security of this Mortgage; and the performance of the covenants and agreements of Borrower herein contained, Borrower does hereby mortgage, grant and convey to Lender the following described property located in the County of CLERMONT, State of Ohio:

SEE LEGAL DESCRIPTION ATTACHED HERETO AND MADE A PART HEREOF AS EXHIBIT "A".

Tax# 28-28-07-B-071

Capturing data from forms that are not machine-readable

Sometimes it is not necessary to capture all the data available on a document. This is particularly true when digitising archives. In this case only certain fields are selected for recognition. The program creates a unique index on the basis of these fields and converts the document images into a suitable storage format. Next time when someone looks for a particular page, they will be able to find its image by carrying out a search in the index.

Documents stored in archives are usually not machine-readable:

- The location of fields is not fixed,
- There are no reference points (e.g. black squares or crosses),
- Fields may include words in cursive writing,
- Fields may be obstructed by stamps or inscriptions.

This means that such documents cannot be processed using the conventional template approach.

Solution

1. The traditional approach must be used wherever the fields have fixed locations - templates must be created for such documents. Captions or tables that are present on all the documents can be used as reference points. Usually archive documents contain about a dozen such fields, which is sufficient for reliable data capture. The title of the document can be used as an identifier. This method can be used to create at least part of the index. For example, in the case of waybills or invoices the program can recognize bar codes, the country of origin, the telephone of the sender and the postal code of the addressee.
2. Fields without a fixed location can be recognized using the FlexiCapture technology. This technology can find any field if information about its surrounding elements is available. The only drawback is that creating formalized descriptions is fairly expensive and can only be done by a specialist.

3. Sometimes good results can be achieved using the Key From Image approach, or manual input fields. This approach has already been described on p. 27. The program helps the operator find the required field on the form and displays its image. Then the operator types in the data manually. This method can be used to enter information from any document that cannot be read by computer and is more convenient than ordinary manual typing.

Automated input of separate fields from not machine-readable form.



Conclusion

We have considered the main features of automated data capture. We started by introducing the terms and concepts commonly used by data capture professionals, then took a closer look at forms processing proper, concentrating on the most important aspects. We proved that automated forms processing has a lot of advantages over manual typing and showed how the quality of capture could be improved. Next, we provided an overview of possible capture solutions for different kinds of tasks and offered some advice on form design.

When writing this White Paper we drew heavily on our experience with ABBYY products, but most of the recommendations in this document will also apply to other data capture software. We have attempted to present information related to data capture in

a manner that would be of interest both the specialist and the novice alike. We hope that we have achieved this goal and the information in this White Paper will prove useful to all those developing, implementing and running such projects.

Contacts

ABBYY Software House (Moscow)

P.O. Box#54
Moscow, Russia, 129301
Tel.: +7 095 783 4700
Fax: +7 095 783 2663
formreader@abbyy.ru

ABBYY Europe GmbH

Anglerstrasse 6, Munich,
Germany, 80339
tel.: +49-89-511159-0
Fax: +49-89-511159-59
sales@abbyyeu.com

ABBYY Ukraine

P.O. Box#23
Kyiv, Ukraine 02002
Tel: +380 44 490 9999
Fax: +380 44 495 2080
sales@abbyy.ua

ABBYY USA

3823 Spinnaker Court,
Fremont, CA 94538
Tel.: +1 510 226 6717
Fax: +1 510 226 6069
sales@abbyyusa.com