White Paper: Best Practices in Form Design for ABBYY Flexicapture Fixed Form Processing

User Friendly Consulting, Inc. White Paper

By: Jim Hill

Published: May 2014

Best Practices in Form Design for ABBYY Flexicapture Fixed Form Processing

May 2014

Table of Contents:

Introduction to ABBYY Flexicapture

[ABBYY Flexicapture](#) is a comprehensive forms processing and document capture solution. It includes standard import features for scanning documents, receiving documents from email, processing images from a watched network folder, and manually processing images. Standard export features include output of files to a wide variety of formats, enterprise content management systems, and Microsoft SharePoint. The product is available in two types of architecture depending upon your needs. The standalone system is for a single user processing smaller volumes of documents from a single workstation. The distributed version is the scalable client-server application that runs in Microsoft IIS. The distributed flavor provides for large scale processing of large volumes of documents spread across multiple servers for load balancing and failover in a Microsoft Server cluster. Within the standalone and distributed installation types are additional consideration including whether there is need to process fixed forms, semi-structured forms, or unstructured forms. The last two types require the use of the Flexilayout Studio product and licensing. Structured forms can be processed by the standard versions of Flexicapture standalone and distributed. Both versions include a tool called the FormDesigner for the creation of "scannable" paper forms.

Considerations in Form Design That Impact Proposals and Request for Quotations for Flexicapture

There are several form design considerations that must be taken into account in order to provide an accurate quotation for an ABBYY Flexicapture solution. First, the analyst must determine whether the forms to be processed require the use of the basic fixed-form product, or the more advanced Flexilayout Studio. Since the price difference between the two is significant the customer is not going to be pleased if they determine that the fixed-form product is unable to produce acceptable results on their documents.

A second and extremely important consideration of form design that would impact the quotation is the requirement for the Flexicapture software to process tables or repeating groups. This is significant because some forms naturally lend themselves to the use of a table object in the form design, whether or not the form is created in the ABBYY FormDesigner, Adobe InDesign (the preferred method), or the customer has an existing form that requires processing. Again, since there is a significant difference in price to add the table item / repeating group option to the system your customer won't be happy if you determine that you require the use of this technology to successfully process the form.

Finally, determine in advance and include in the quote specific detail about who is responsible for form design and testing. The form design process can be very time consuming and you do you customer a large favor by delineating in your proposal the responsibilities and charges for form design and validation.

Hints for Using the ABBYY FormDesigner Tool

You can save yourself considerable grief by understanding the ABBYY FormDesigner as a tool for creating prototypes of forms to be exported image format to Adobe InDesign for the final design. There may be exceptions to this principle for certain simple forms with limited, widely spaced elements. This advice of using the FormDesigner for prototyping only will save you considerable grief when it comes to producing forms which work well in Flexicapture, even though the FormDesigner automatically creates much of the document definition (template).

The following are some examples on how the FormDesigner could help you design a better form once you begin the production form design using a form design tool like Adobe InDesign.

- Use the designer to provide the exact dimension of form objects, then use these dimensions when you build the form in InDesign. For example, using the ABBYY tool to design a series of text entry fields with 4mm wide by 5 mm tall marking frames. *Figure One*
- Use the designer to quickly alternative prototype versions of forms containing a "straw man" set of form fields. Consider a prototype form with the form elements generated by the ABBYY FormDesigner.
- Build an image containing the required page anchor elements, such as black squares or angled corner marks.
- Create barcode element for matching of the document definition.
- Create all of the columns and one row of data for a table element.
- Use the designer to a "filling template" element that shows users of the form how to properly print the handwritten letters and mark checkboxes. *Figure Two*
- A table object with checkbox or text entry fields. *Figure Three*
- Text entry fields using constrained elements.
- Build a prototype form with a combination of elements for rapid testing with the customer.

Figure One: Text Entry Field (4 x 5 mm)

Download

Figure Two: Sample Filling Template from ABBYY
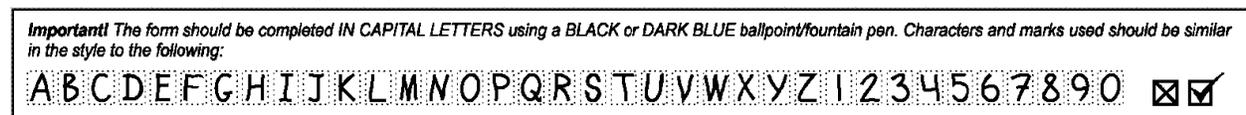
Download

Figure Three: Table Object with Entry Fields

| Small	Med	Large	XL
☐	☐	☐	☐

Download

Form Design Considerations for Various Field Types

In general you want to be certain to provide as much information as possible in the document definition field properties for each field on the form. This may seem intuitive until you realize that the more information about the field the less possibilities Flexicapture will have to assign to the field. There can be a small difference for example between specifying just a data type as a name versus a first name. Finally, anytime you can utilize a database lookup to restrict the possible values for the field you will increase the accuracy of the recognition.

Text Entry Field for ICR:

We have found that we can achieve superior recognition results when using the character box series marking type and then designating the number of cells when you create the document definition for handwriting recognition (ICR). These full squares seem to provide a better method for users to enter data and Flexicapture works better when using this marking pattern (see Figure Four). The second type that works very well is this same type of character box series but with the cell boxes marked in a dropout color. When you create the document definition with this type you will just need to change the marking type to "simple" since when the form is scanned the character box series marking will disappear leaving just the characters themselves. Finally, we have not had very good results with using the "dotted frame" marking method as provided by the FormDesigner. The issue with these marking types is that during the scanning process the dots tend to enlarge during the degree of freedom added during the scanning operation. Think of what is known as the "fax game." Someone in an office makes a copy of document and send this first copy through a fax machine, which is in effect a very low quality scanner. The person on the other end receives the fax and sends it by fax to another person, and the cycle repeats. After a dozen faxes the document has changed quite a bit by darkening text areas and noise (dots and speckles). Evidently the same thing happens and when text entry fields are marked with the dotted frames they tend to thicken with scanning. Even when the document definition is set to de-speckle you will experience poor recognition results for handwritten values.

Figure Four: Text Entry Field using Character Box Series
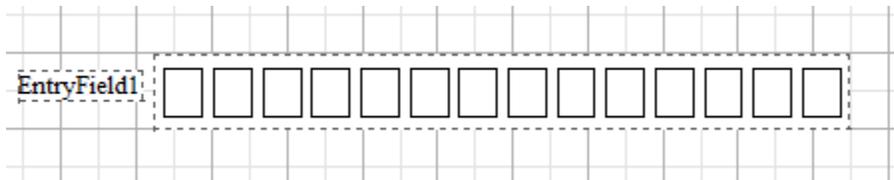
Download

OMR Checkmark Field:

We have found superior results when using the rectangular field marking for these fields. As an alternative we tried using oval markings instead of rectangles, then when the document definition was created it specified rectangular values since there is no option for ovals. This marking type worked very well even with the ovals.

Suggestions for Including Options in Business Proposals

Our advice to anyone writing a proposal for Flexicapture is to include the table object (and repeating groups) add-on as an option, along with the semi-structured Flexilayout Studio pricing in each proposal. This only takes a small amount of time and could avoid an unhappy customer when the need for one of these options arises.

Scenario:

I was working with a company who desired to extract handwritten text and checkmark data from a form. During the sales cycle we created a template for demo purposes using the fixed form Flexicapture standalone system without the table option. Since we obtained good results we never included the option or even mentioned the possible necessity of the option for processing tables in the quotation. Once the customer purchased the system they set out to build new order forms in order to optimize the extraction results using the ABBYY FormDesigner Tool to create a prototype> Next they finished up the form design using Adobe InDesign. It was quickly determined that the customer needed the table option on their system because of the particular design of their form. When I approached the customer with this information they tersely asked me why even the possibility of this need was mentioned during the courtship. I explained that their original form did not lend itself to the use of the table objects but their new form design did work best with the use of this more advanced technology. After some discussion they agreed to the purchase the option.

Conclusion

Use the ABBYY FormDesigner tool to create high quality prototype forms for testing and evaluation, then create the production form in Adobe InDesign using the form elements produced in the ABBYY tool. Use caution in specifying page elements on forms such that you optimize the data extraction capability of the ABBYY Flexicapture system.

Jim Hill, UFC, Inc. Contact me at: (248) 447-0102

Appendix: Scripting Example for Counting Checkbox Values

We were asked by a potential client whether we could 1. Count up the number of times a group of checkboxes were marked, and 2. Count the number of markings for the group to which the checkmark belonged, either normal or significant. Some checkmarks were in the normal category, while some were assigned to the more critical "significant" category. The following is the script that we created in the document definition for this operation. You may find it useful when you have to tally and make mathematical calculations for OMR fields.

```
Dim deficiencies
Dim sigdeficiencies
'Created by Jim Hill, UFC, Inc.
'The script determines a count of checkboxes on the forms for the four field groups (1.Controls -
4.Other)
'It reports a separate value for total number of deficiencies and another for significant
deficiencies
'Significant deficiencies are bolded fields on the image and the fields are designated -SIG
'Set the variable to zero so it's not a null so it can be incremented
'But only do this the first time for checkbox group 1 then increment throughout the project
deficiencies = 0
sigdeficiencies = 0
'Change the field value to the proper value, could do this in a for next loop but there are only
three values
if Me.Field("1a").Value = "Y" then
    deficiencies = deficiencies + 1
end if

'increment then repeat the counter for the remaining fields
if Me.Field("1b").value = "Y" then
    deficiencies = deficiencies + 1
end if

if Me.Field("1c-SIG").value = "Y" then
    deficiencies = deficiencies + 1
    'increment the significant deficiencies since 1c is a bolded value on on the form (significant)
field
    'change this for each bolded field on the form
    sigdeficiencies = sigdeficiencies + 1
end if
if Me.Field("2a").Value = "Y" then
```

```
      deficiencies = deficiencies + 1
   end if
   if Me.Field("2b").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("2c-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("2d-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("2e-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("2f-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("2g").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("2h").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("2i-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("3a-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("3b").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("3c").Value = "Y" then
```

```
      deficiencies = deficiencies + 1
   end if
   if Me.Field("3d-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("3e").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("4a").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("4b").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("4c-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   if Me.Field("4d").Value = "Y" then
      deficiencies = deficiencies + 1
   end if
   if Me.Field("4e-SIG").Value = "Y" then
      deficiencies = deficiencies + 1
      sigdeficiencies = sigdeficiencies + 1
   end if
   Me.Field("Count of Deficiencies").Value = deficiencies
   Me.Field("Count of Significant Deficiencies").Value = sigdeficiencies
```