
Amgraf's

**OneForm Designer Plus
PDF Forms Generator
Module**



MECCA™ III
FORMS COMPOSITION FEATURES

Reference Manual

OneForm™ Designer Plus

© 2019 by Amgraf, Inc. All rights reserved.

These specifications are subject to change without notice.

OneForm and MECCA are trademarks of Amgraf, Inc.

All other brand and product names are or may be trademarks used to identify products or services of their respective owners.



1501 Oak Street • Kansas City, MO 64108-1424

Phone: (816) 474-4797 • Fax: (816) 842-4477

E-Mail: info@amgraf.com • Internet: www.amgraf.com

TABLE OF CONTENTS

Introduction

Getting Started	1
Software Requirement	2
Server Setup Requirement	2

Chapter 1: Setting Up the Sample Form

Open the Sample ELF	3
Add Required Submit Buttons	4
Move Buttons to Suitable Location	7
Setting the Tabbing Order	8

Chapter 2: Assigning the Output Database Tables

Assigning the Output Database Primary Table	9
Use System DSNs	10
Assigning Stepped Fields to SubTable	13
Viewing the Results	15

Chapter 3: Creating the Fillable PDF Form

Save Form and Fields as PDF	17
Save As in Acrobat for Final PDF E-Form	19
Controlling the PDF Page Size	20
Working with Multiple Pages	20

Chapter 4: Generating the Server Side Scripts

E-Form Server Preferences	21
Specify the Search and List Criteria	22
The Submit Data Button	24
The Generate Scripts Tab	25
Copy Folder to Server and Test	27

Chapter 5: Adding Fillable Fields to the Original PDF

Save Form Fields as PDF	29
Bring the Two PDFs Together with Adobe Acrobat	31
Moving the Fill Fields	32
Save As in Acrobat for Final PDF E-Form	33
Controlling the PDF Page Size	34
Adding JavaScript Programming	34

Chapter 6: PDF Save Options	
Default Settings	35
Field Options	36
Special Output Options	37
Chapter 7: Adding the Finishing Touches	
Setting the Tab Order	39
Adding Help Messages	41
Adding Input Masks	42
Numeric Format Examples	44
Changing Field Colors	46
Adding Text HyperLinks	47
Controlling the Checkbox Character	47
Adding Borders to Fields	48
Using PDF JavaScript Code	48
Chapter 8: Duplication, Continuation, and Overflow	
Multi-Page Features for Flattened PDFs	
Duplication Multi-Page Feature	49
Continuation Multi-Page Feature	52
Overflow Multi-Page Feature	55
Chapter 9: Saving to a Flattened PDF File Only	
Overview	59
Setting Up the Form	60
Using the Output Database Option	60
Generating the Server Side Scripts	62
Generating the Final HTML or PDF E-Form	64
Chapter 10: Miscellaneous Information	
Details on Fields Used	65
Editing Results Script to Ignore Drafts Saved	67
Combining Fields onto Original Art PDF Pages	68
Chapter 11: Using Import PDF	
Converting the PDF File	69
Chapter 12: PDF Files for Digital Printers	
PDF Button on Top Toolbar	71
Making PDFs for Spot Color Files	74

INTRODUCTION

Getting Started

The PDF Forms Generator Module is a software option that extends the functionality of the e-forms features in the standard OneForm Designer Plus package. With Amgraf OneForm Designer Plus and the PDF Forms Generator Module, a PDF file can quickly be fielded to support data collection and data presentation from an Internet web site.

The e-forms conversion process is easy. Using OneForm Designer Plus, a standard PDF file can be imported and displayed. Form fill fields are then added, field attributes are specified, and JavaScript coding can be inserted for form customization, intelligence, and computation. The file is then saved with the form fields as a new Internet-ready PDF fillable e-form that works with the free Acrobat Reader.

To host fillable e-forms on a web site you must have custom CGI scripts and databases. All of the required scripts are automatically generated by OneForm Designer Plus with the PDF Forms Generator Module.

Before you begin creating PDF fillable e-forms, you may want to study and practice the procedures to import PDF files within the OneForm Designer Plus Business Forms Composition User's Guide. The procedures for creating the necessary CGI scripts for your PDF e-forms are the same as creating HTML e-forms, so you may already be familiar with these steps.

There is also an option to save only the JavaScript coding and e-form fields to overlay onto the original PDF using Adobe Acrobat 5.0 (or higher) software, if there is a need to use the original PDF file.

Following are the steps to create an Internet-ready PDF e-form with fill, print, save, and recall capabilities, using the original PDF as a starting point:

1. Import Original PDF Form
2. Add Fillable E-Form Fields and Specify Field Types (if the Original PDF does Not Include Fields)
3. Add Submit Button, Database Connections, and Custom JavaScript
4. Generate CGI Scripts for the PDF E-Form
5. Save As Form and Fields with JavaScript code as PDF File
6. Open PDF File in Acrobat and Save As or Optimize Save
7. Copy Files to Internet Server

The next few chapters assume you already know how to import PDF files and use the e-form fielding tools. We will start with Step 3 and take you through the process of creating an Internet-ready PDF e-form from a fielded OneForm file. A sample OneForm file from our Forms Library named **req04.elf** is included with the PDF Forms Generator Module for use with this documentation.

Software Requirement

You will need to install the full **Adobe Acrobat 5.0 (or higher)** software on your PC before proceeding.

Server Setup Requirement for Acrobat Reader

When using OneForm's server scripts with PDF E-Forms for Acrobat Reader, you will need to copy font width files to the server. These font width files are necessary for your fillable Internet PDF e-form being used with Acrobat Reader and OneForm's Text fields.

Start by copying the ".afm" files OneForm provides. They are located in the ***OneFormInstallPath\OneForm\PerlModules\UtilityPackages\OOES*** folder. Copy the ".afm" files to the *serverpath/cgi-bin* folder.

Then copy the entire **OOES** folder mentioned above to your I-Forms server at the exact location named: *serverpath/UtilityPackages/OOES*.

CHAPTER 1

Setting Up the Sample Form

Open the Sample ELF

Start OneForm Designer Plus and open the **req04.elf** file. It can be found at the following location:

**C:\Program Files\Amgraf\OneForm\Form_Lib_Wizard
\Wiz_requisition\req04**

As you can see, the sample is a Requisition Form that is already fielded through the standard procedures.

The screenshot shows the OneForm Designer Plus interface. The main window displays a 'SAMPLE REQUISITION FORM' with the following fields and sections:

- DATE** and **REQ NUMBER** (text input fields)
- FROM:** (text input field)
- CHARGE TO:** (text input field)
- DELIVER TO:** (text input field)
- JOB NUMBER:** (text input field)
- ATTENTION:** (text input field)
- Table:** A table with 7 columns: **QUANTITY**, **ITEM NUMBER**, **DESCRIPTION**, **UNIT OF PURCHASE**, **DATE REQUIRED**, **UNIT PRICE**, and **COST**. The table contains several rows of empty cells.
- TAX EXEMPT:** YES NO
- TOTAL COST:** (text input field)
- SPECIFICATIONS:** (text input field)
- RECOMMENDED SUPPLIERS:** (text input field)
- REQUISITIONED BY:** (text input field)
- DATE:** (text input field)

The status bar at the bottom shows: For Help, press F1 | SPOT CMYK | P All | Move AnyWhere | GridSnap Off/CompSnap On | 299.77, -127.38

Figure 1-1: Our Sample Form

This form is ready to be turned into a PDF Fillable E-Form.

Add Required Submit Button

We begin the process of creating a PDF Fillable Form by adding the necessary submit button and required fields. To do this, click the tool **Add Buttons**.

Select **PDF Compatible** as the Form Type to Generate.

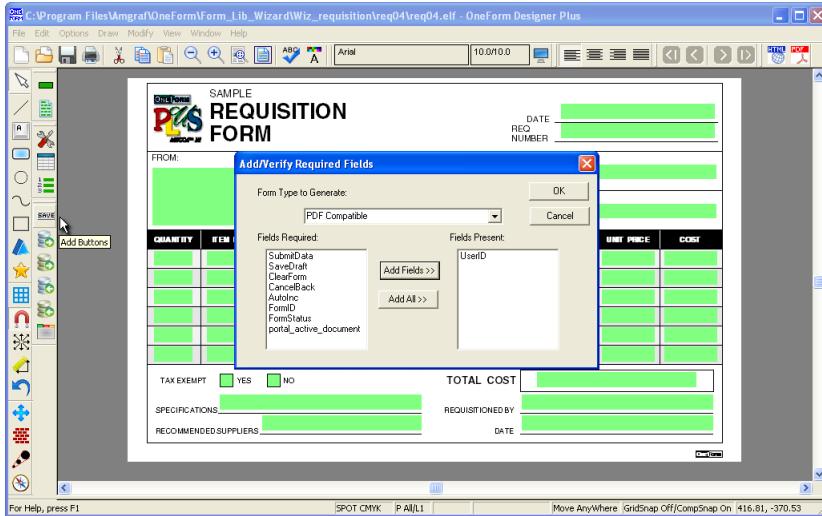


Figure 1-2: Add Control Buttons

This will show a list of possible buttons and fields that can be added to the sample form. For this example, we will select the following: **SubmitData**, **SaveDraft**, **ClearForm**, **CancelBack** button options, the **AutoInc** read only field, and the hidden field **FormStatus**.

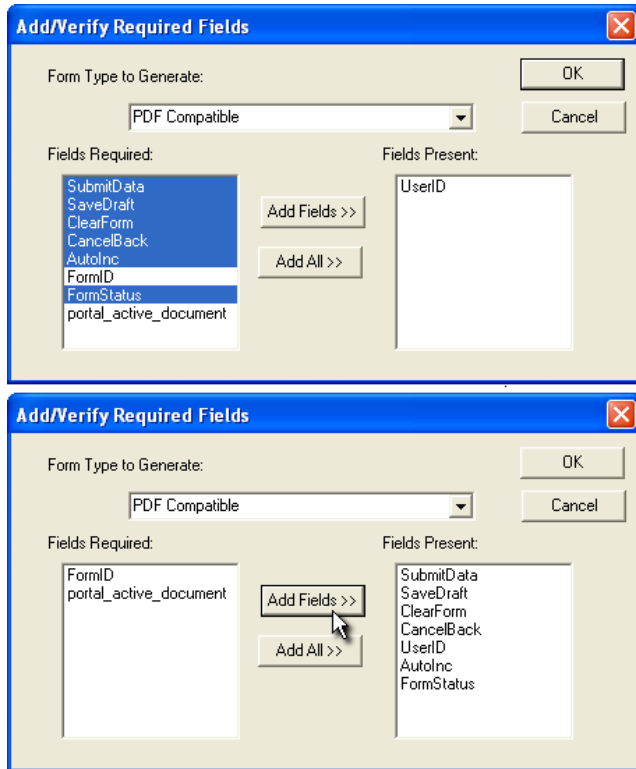


Figure 1-3: Selecting Control Buttons

Click the [Add Fields] button, then [OK] to exit this step.

Add Required Submit Button

You will see four buttons have been added to your form.

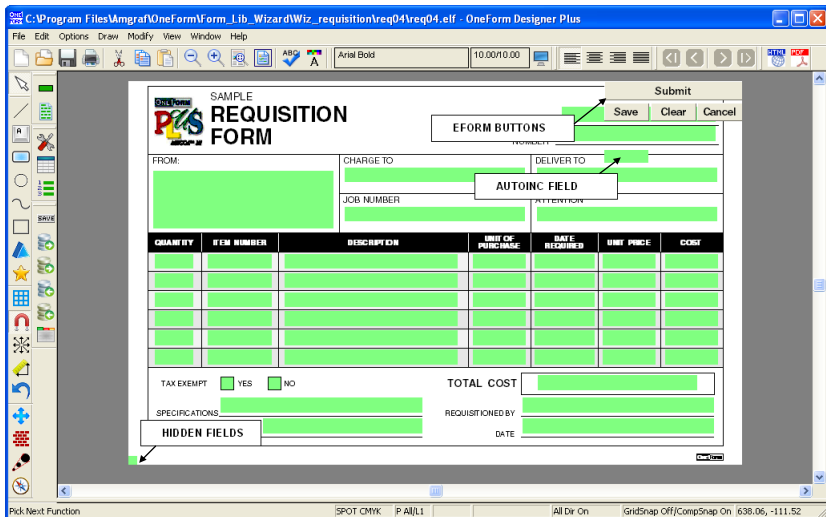


Figure 1-4: Buttons and Fields are Automatically Added

The [Submit] button (SubmitData field) does exactly that, submits the user's data to the server.

The [Save] button (SaveDraft field) also submits data, but flags it as a FormStatus of 2. This can be used with your forms to represent unfinished forms. You can then search for "Saved Drafts" versus "Final" forms, if needed. More information on setting and using this is found on in Chapter 9.

The [Cancel] button will close the form, while the [Clear] button clears all data on the form. These buttons are for PDF fillable e-forms only. There are separate buttons for HTML e-forms.

The AutoInc field will be used as the primary key for the output data table. It automatically numbers each record sequentially as it is submitted.

The SubmitData, AutoInc, and FormStatus fields are required. The others can be deleted if not wanted.

Move Buttons to Suitable Location

The buttons will always go to the upper right-hand corner, so move the buttons to a more suitable location on the form.

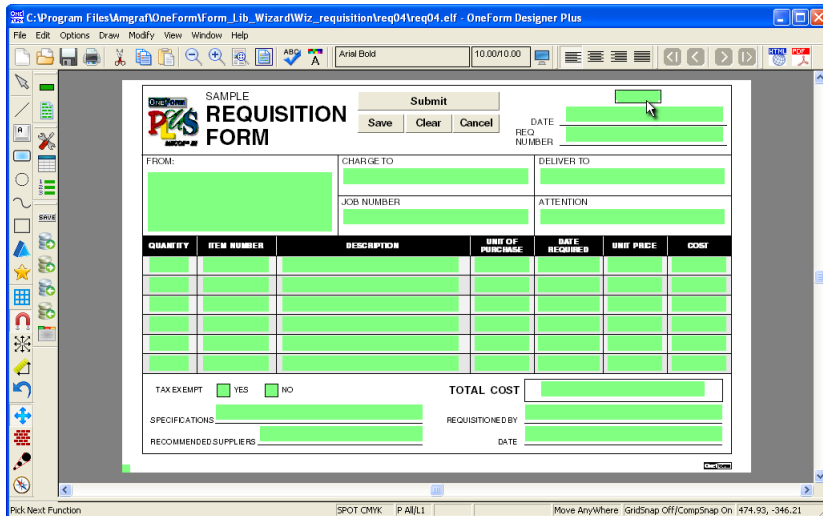


Figure 1-5: Reposition the Control Buttons and AutoInc Field

The AutoInc field is also in the upper right-hand corner. On this form it has fallen on top of the line underneath the Requisition Number field. Move it to a new location.

Setting the Tabbing Order

Since you have added fields to the form, let's establish their tabbing order. Go to **Options, EForm Options, EForm Set Tab Order**. You will notice the buttons have a tab order of 0. Use the **Restart Tab Order** option in the right-mouse pop-up menu and select the last tab on the form as the restart number. Then pick each button assigning it a tab order.

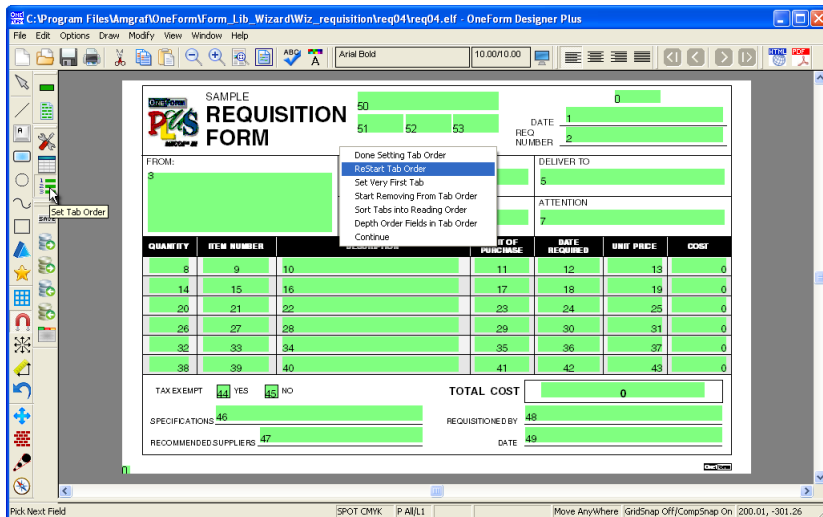


Figure 1-6: Setting Tab Order of the Control Buttons

Use the pop-up menu's **Done Setting Tab Order** to save the tab settings. This would be a good time to also **Save** the changes made to the file.

Note: It is much easier to establish tabbing order in OneForm so be sure to always include "Setting Tab Order" into your steps of preparing an e-form. Otherwise, the tabbing order will be random in the PDF file.

CHAPTER 2

Assigning the Output Database Tables

OneForm Designer Plus can create new tables for your e-form within an existing database through a CGI script. Our example will use a DSN connection to a MySQL database to check for existing table names.

Here we will cover the steps for assigning output database table names. These tables will be used to store and retrieve the data submitted from the PDF fillable e-form. The stored data can be retrieved back into its original e-form through a "Primary Key" field.

Our Primary Output Key field for this example is a Numeric field named "AutoInc".

Assigning the Output Database Primary Table

Start by clicking the **New Output DB Table** tool (or you can go to Options, EForm Options, EForm Output Database, New Output Table).

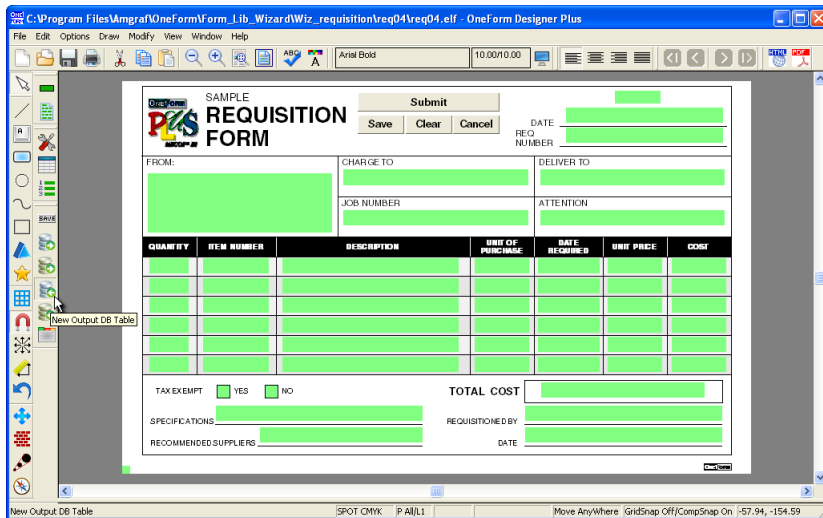


Figure 2-1: Use the New Output Database Table Tool

Use System DSNs

Check the option for "Use System DSNs" and **locate your existing database file** from the drop-down list. Ours is called "IForms". Once located, it will automatically name the new primary table based on the ELF name.

Look at the Existing Table Names and make sure there is no conflict. You can rename your table if necessary.

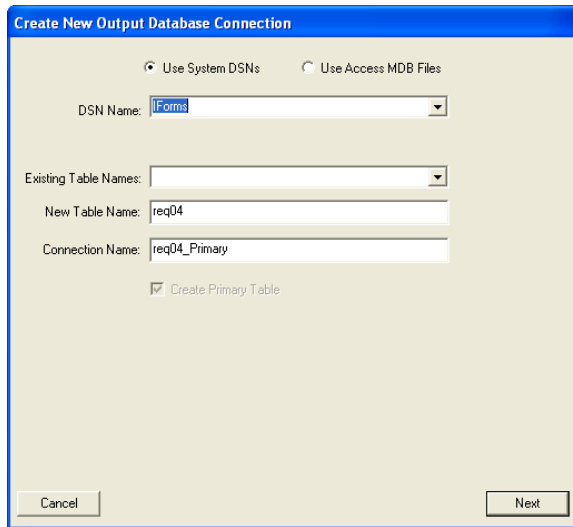


Figure 2-2: Use System DSNs

Click [**Next**]. This will take you to the Output Database Fields Dialog for selecting what fields are to be captured.

Note: This example is pointing to a previously defined System DSN connection. If no DSNs are listed, you may use a local MDB database instead. If that is the case, select the "Use Access MDB Files" instead and [**Browse**] for the MDB database.

Assigning the Output Database Primary Table

The Output Database Fields Dialog lists the fields on the e-form in the left-hand window area. The right-hand window area will list fields as they become assigned to the table.

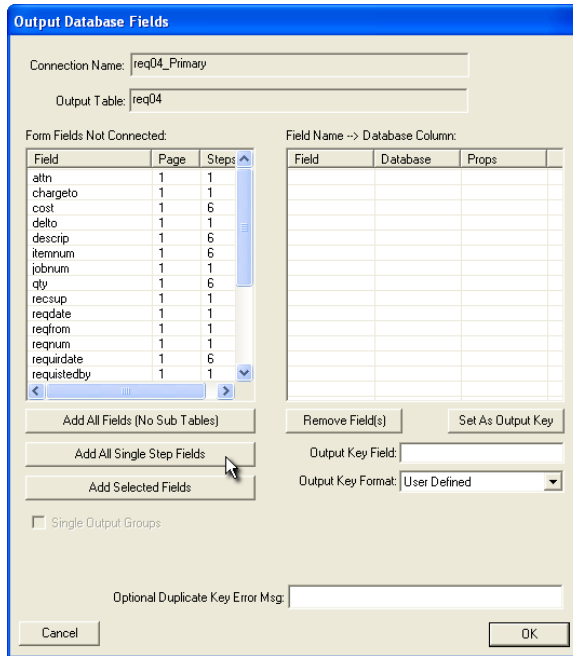


Figure 2-3: Output Database Fields Dialog

There are three options for assigning fields:

[Add All Fields (No SubTables)] – This button takes all fields and assigns them to the primary output database table. Even stepped fields will be included and their DB field names would be something like: price0, price1, price2, etc.

[Add All Single Stepped Fields] – This will assign only the single fields to the Primary table, leaving the stepped fields to be assigned to a different table later.

[Add Selected Fields] – Use this option to identify one or more fields to add individually. Select them using the standard Windows highlighting techniques.

Assigning the Output Database Primary Table

Click [**Add All Single Stepped Fields**]. This will collect only the single fields into the Primary table and move them to the right, leaving the stepped fields to be assigned later into a subtable.

The right-hand window identifies fields you have associated. From this window, **highlight** the primary key field **AutoInc** and click [**Set as Output Key**].

For the *Output Key Format*, select **AutoIncrement** from the list.

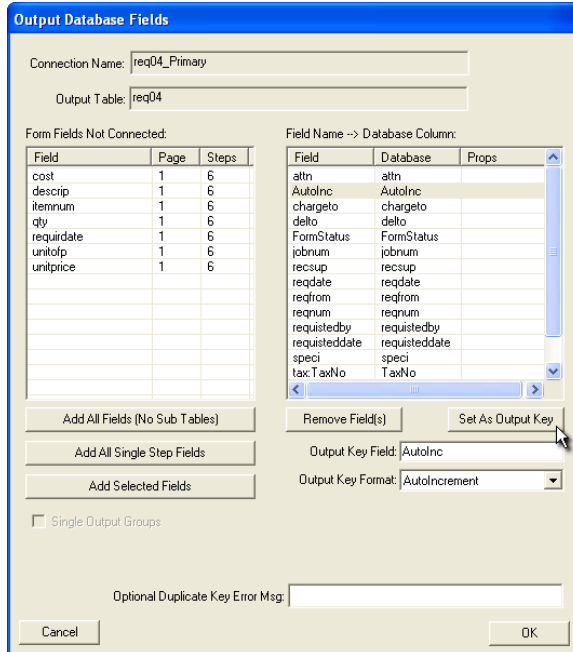


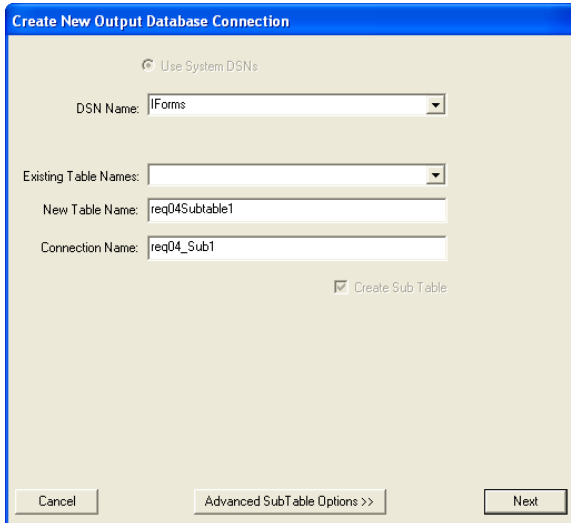
Figure 2-4: The Primary Key is the field named "AutoInc"

Click [**OK**] to assign the table to the fields. This will close the dialog.

Assigning Stepped Fields to a SubTable

Our e-form has seven columns, each holding a stepped field to make up the rows for input. Assigning stepped fields to its own subtable allows efficient data handling. Only rows used during input will be captured, ignoring all rows which are empty.

Start by clicking the **Options** menu, **EForm Options**, then the **EForm Output Database, New Output Table** option.



The screenshot shows a dialog box titled "Create New Output Database Connection". It features a blue header bar and a light beige main area. At the top, there is a checkbox labeled "Use System DSNs" which is checked. Below this, there are four input fields: "DSN Name" with a dropdown menu showing "Forms"; "Existing Table Names" with an empty dropdown; "New Table Name" with the text "req04Subtable1"; and "Connection Name" with the text "req04_Sub1". A checkbox labeled "Create Sub Table" is checked. At the bottom of the dialog, there are three buttons: "Cancel", "Advanced SubTable Options >>", and "Next".

Figure 2-5: SubTables are Automatically Named

Each e-form can have only one Primary table, so it will automatically put them into a subtable. Click **[Next]** to continue.

Assigning Stepped Fields to a SubTable

This time **highlight the stepped fields** within the left-hand window and then click **[Add Selected Fields]**. This will move them into the right-hand window showing they are assigned. The question mark (?) flags it as a stepped field.

Output Database Fields

Connection Name: req04_Sub1

Output Table: req04Subtable1

Form Fields Not Connected:

Field	Page	Steps

Field Name -> Database Column:

Field	Database	Props
cost	cost?	
descrip	descrip?	
itemnum	itemnum?	
qty	qty?	
requidate	requidate?	
unitofp	unitofp?	
unitprice	unitprice?	

Remove Field(s)

Output Key Field: AutoInc

Output Key Format: AutoIncrement

SubTable Key: SubKey

SubTable Key Format: Primary_%.d

Add Selected Fields

Single Output Groups

Cancel OK

Figure 2-6: [Add Selected Fields] will Assign Properties to Highlighted Fields

Click **[OK]** to save the subtable association. The dialog will close.

After assigning database table properties, remember to **Save** your file. These values will be used when it comes time to generate CGI server scripts.

Viewing the Results

Going to the **View** menu and selecting the **EForm Fields, Show DB Table Names**, will display the field name within the DB table. The number behind each name is a reference to a table.

Look at the field jobnum(1) under JOB NUMBER. The (1) represents the first database table assigned; the primary output table.

Look at the fields descrip?(2) under the Description column. The ? means the field is stepped and the (2) represents the second database table assigned; the output subtable.

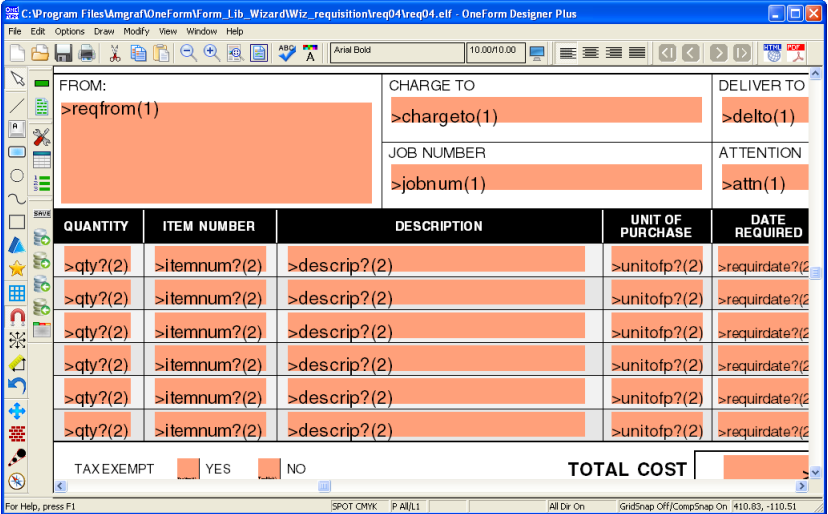


Figure 2-7: Viewing the Database Assignments

CHAPTER 3

Creating the Fillable PDF Form

Save Form and Fields as PDF

Fillable PDF forms are created with OFDP using the PDF Forms Generator Module. To create a PDF form, click the **Save As PDF** tool. This will create a complete PDF E-Form with the form artwork components (lines, text, boxes, etc.) along with the fill-in fields and any embedded JavaScript code.

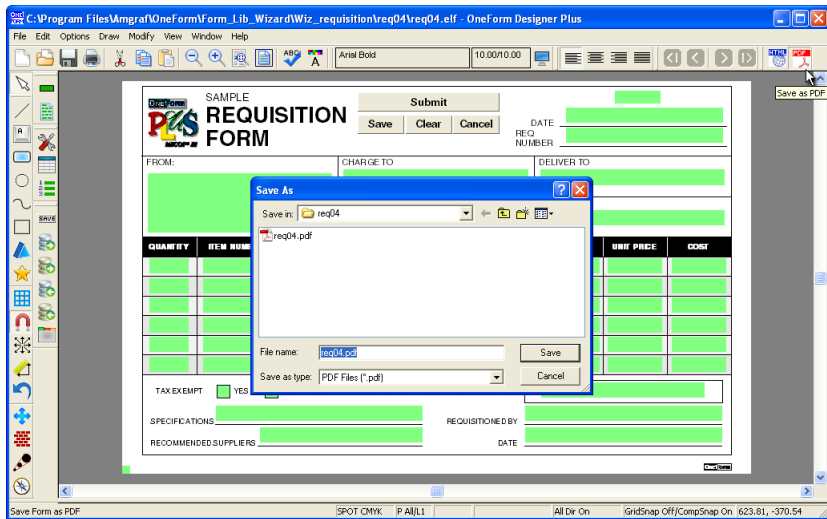


Figure 3-1: Saves the Entire Form as a PDF File

Note: PDF filename Must be same as "Form Name" of EForm Preferences.

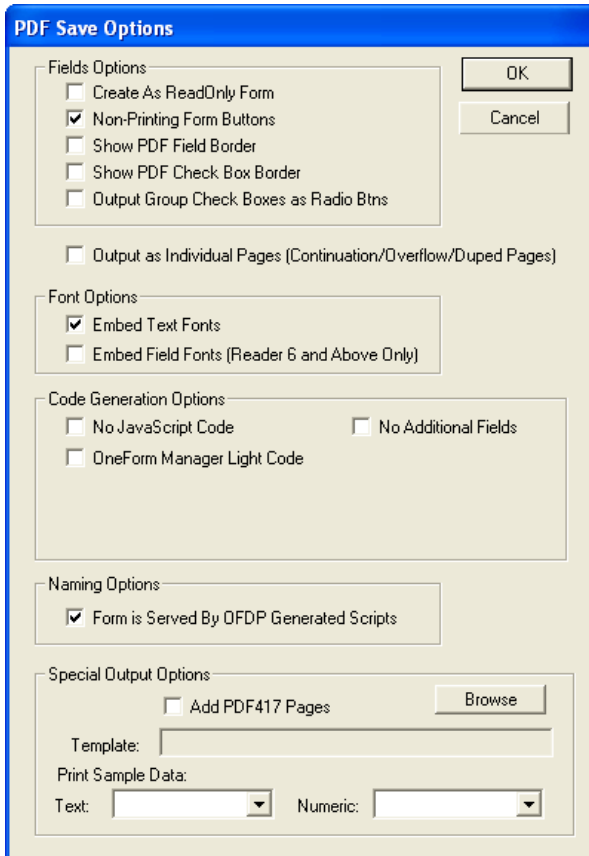


Figure 3-2: Use Default Settings

Okay the PDF Options using the default settings. **The final .pdf file MUST be named using the same base name as the CGI scripts.** This is so the CGI scripts can locate it. It is usually the base filename of the .elf file, but is really the Form Name field of the EForm Preferences. We named ours "req04.pdf". It will place the .pdf file in the same folder as the .elf file in all lowercase.

Note: The PDF file created will always be considered "Portrait" and uses the page size of the drawing properties within OneForm Designer Plus.

Save As in Acrobat for Final PDF E-Form

It is necessary to resave the PDF file within Adobe Acrobat Professional before testing or placing on the server. It must be compressed and normalized to the PDF Version 1.4 standards (Acrobat Version 5.0) for our server scripts, otherwise as it opens you will see "file is damaged and can't be repaired".

Note: DO NOT Test the Final Form before resaving, this will save it with user data.

Each version of Acrobat has a different menu for normalizing the file:

Adobe Acrobat Professional Version 5.0 – Use the File, Save As option and overwrite the existing PDF file.

Adobe Acrobat Professional Version 6.0 and Adobe Acrobat Professional Version 7.0 – Use the Advanced, PDF Optimizer option and overwrite the existing PDF file. The optimizer dialog has several settings that need to be looked at:

Preset: Standard

Make Compatible with: Acrobat 5.0 and later

Cleanup: Optimize the PDF for Fast web view

If you ever make any modifications to the PDF form, save it by using the normalize methods above.

Note: You may modify your PDF form and add custom JavaScript programming using the toolset in Adobe Acrobat. Keep in mind, however, that your OneForm Designer source files will not be updated with any of the changes made in Acrobat. Also, changes made to an Acrobat form are not active until the file has been resaved.

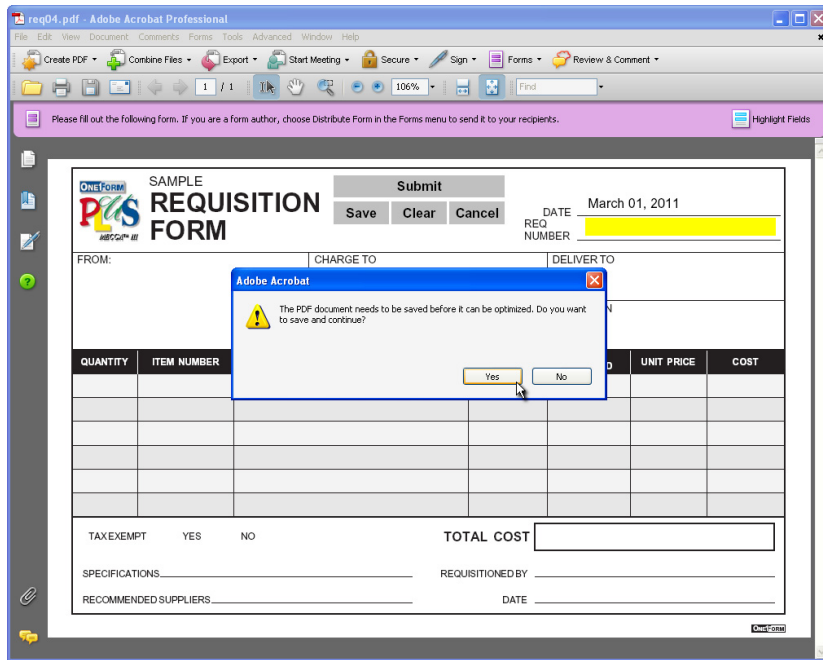


Figure 3-3: A PDF Fillable E-Form

Controlling the PDF Page Size

OneForm Designer Plus always uses the Drawing Properties page size when generating PDF files and will see this as a "Portrait" page.

Combining background artwork with fields using the Acrobat *Document Replace* pages option, causes the page size to become the dimensions of the combined in "Replaced Page".

Working with Multiple Pages

Use the standard *Form Preferences, MultiPage Tab* to link the .elf files. This will create a multi-page PDF when you use the Save As PDF tool.

CHAPTER 4

Generating the Server Side Scripts

E-Form server scripts are generated as described in Chapter 14 of the OneForm Designer Plus User's Guide. Please refer to that chapter for more information.

Here we will only cover the steps that are critical for creating PDF Fillable Forms. The objective is to generate all the server scripts to host the PDF form, and to connect the fill fields to a server output database.

E-Form Server Preferences

Start by clicking the **Server Preferences** tool. This dialog is also found under the *Options, EForm Options, EForm Server Preferences* menu.

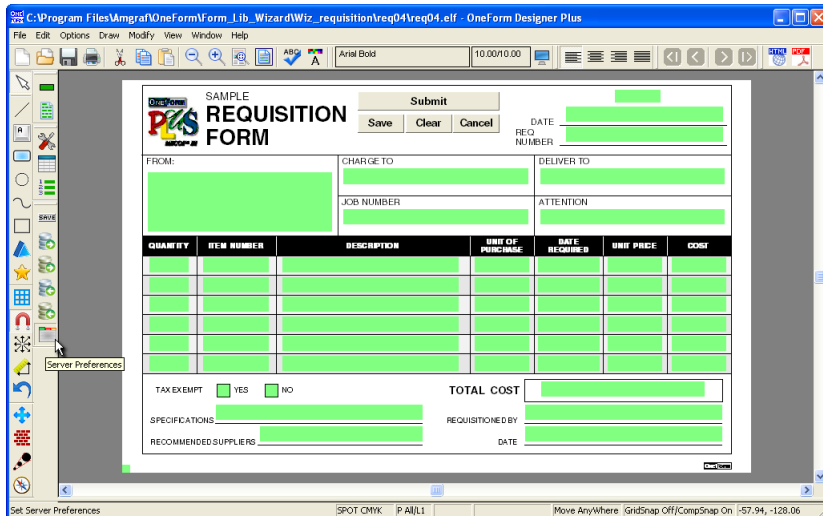


Figure 4-1: Go to EForm Server Preferences

If you have used these tabs to create electronic HTML forms, then the next few steps will be familiar.

Specify the Search and List Criteria

We will start by specifying search criteria. Click on the **Specify Search Form** tab. Here we want to change the label for the *AutoInc* field. Highlight **AutoInc** and click the **[Modify]** button.

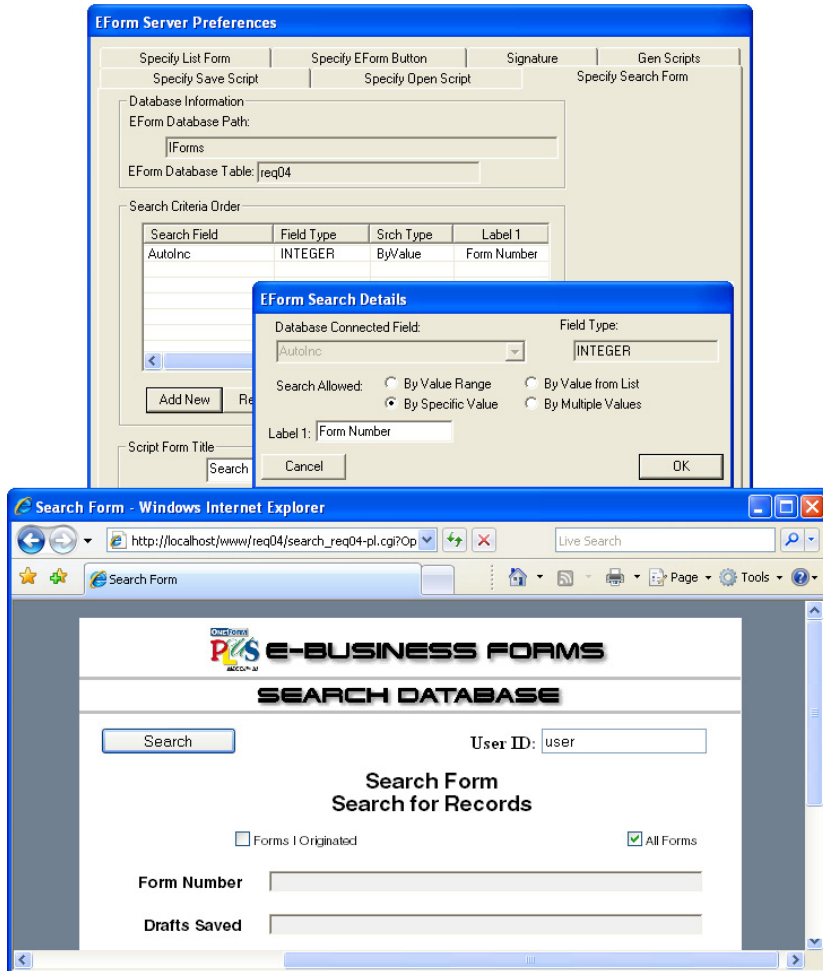


Figure 4-2: Modify "AutoInc" in the Search Script

Change the label to be **"Form Number"** and then **[OK]**. This looks better on our search screen than the word "AutoInc".

Click **[Add New]** and locate the **FormStatus** field from the drop-down list. Label it **"Drafts Saved"**. Then **[OK]**.

The hidden field FormStatus keeps track of which Button was used: [Submit] or [Save]. When the [Save] button is used the FormStatus will be 2.

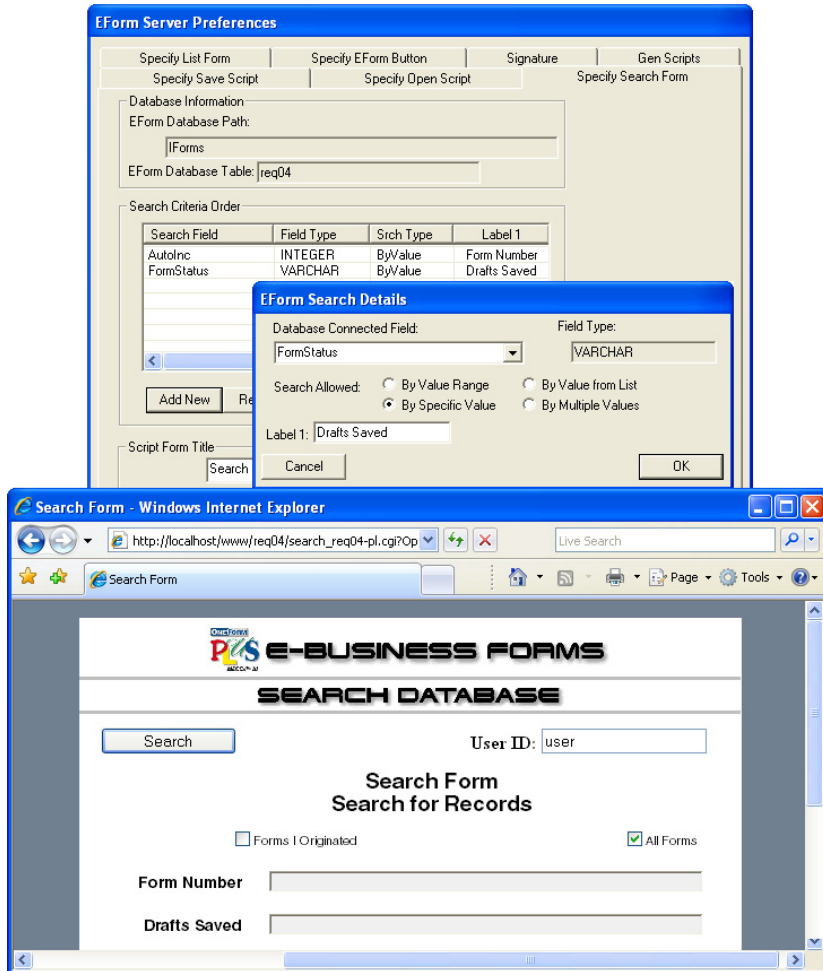


Figure 4-3: Add "FormStatus" to the Search Script

The results list will use the same fields as the search form, so click on **[Copy to List]**.

Clicking **[Apply]** will save your changes to this tab.

The Submit Data Button

Go to the **Specify EForm Button** tab. Be sure to indicate that the **SubmitData** button is to **Submit All Data on Form** by clicking the checkbox as shown in the following illustration.

The screenshot shows the 'EForm Server Preferences' dialog box with the 'Specify EForm Button' tab selected. The 'Form Button' dropdown is set to 'SubmitData'. Under 'Button Display', 'Display as Button' is selected. The 'Submit All Data on Form' checkbox is checked, and its path is 'save_\$.frm.pl.cgi'. The 'Prompt User for Completion' checkbox is also checked. There is a table for 'Script' with columns 'Argument Name' and 'Field Name'. At the bottom, there are 'Apply Changes' and 'Close Dialog' buttons.

Argument Name	Field Name

Figure 4-4: Submit Data Button Settings

You can also "Prompt the User for Completion" by checking this option. Clicking [**Apply**] will save your changes to this tab.

The Generate Scripts Tab

The **Gen Scripts** tab allows you to create the server scripts and administrative HTML pages needed. Make sure to check that you want the **PDF Open Option** for fillable PDF E-Forms. There is also a **PDF Flat Open Option** that can be used to retrieve records into a flattened PDF Form.

Clicking **[Apply]** will save your changes to this tab.

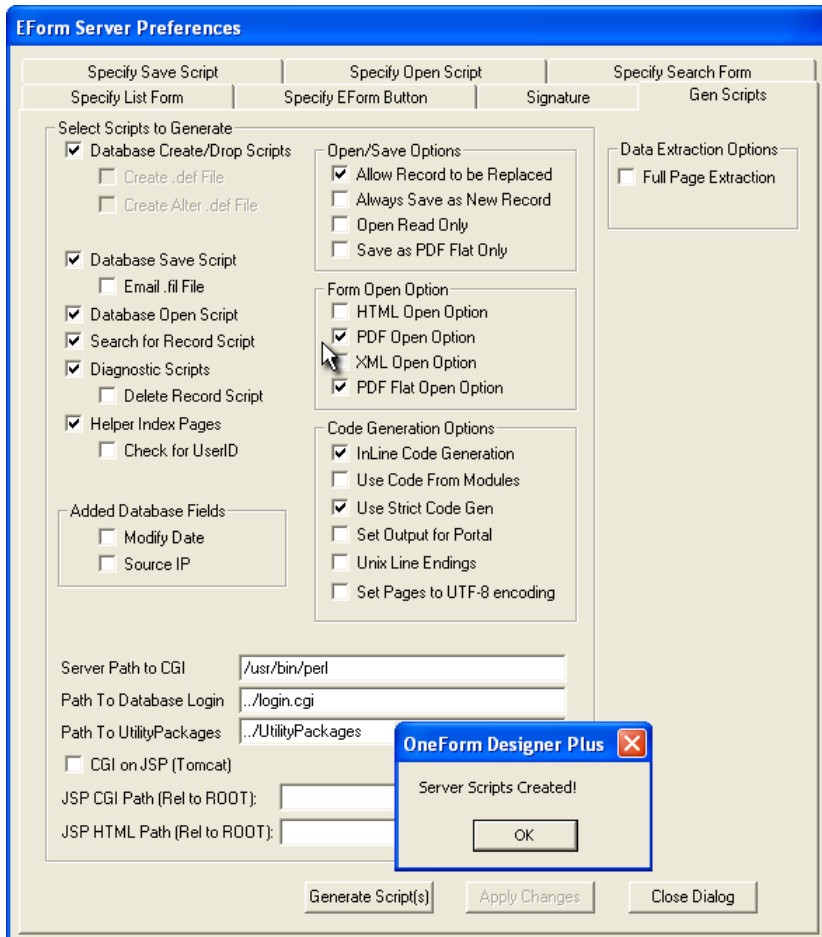


Figure 4-5: Submit Data Button Settings

Click the **[Generate Scripts]** button to generate the server scripts. A list of scripts is shown on the next page.

A pop-up message will confirm that your scripts have been created. You can **[OK]** this and **[Close]** the EForm Preferences dialog.

The Generate Scripts Tab

When opening your working folder, you will see that there are a number of **.cgi** and **.html** files that have been created.

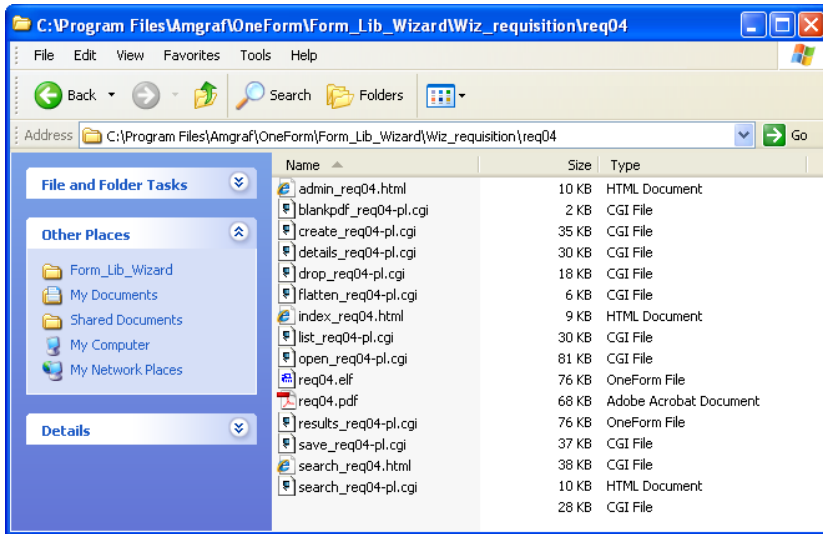


Figure 4-6: Files are Automatically Created by [Generate Scripts]

Copy Folder to Server and Test

The files listed below are the ones needed in your working folder. Copy the folder **req04** to your web server. (The .elf file is not necessary, but it does no harm to have it stay with the scripts on the server.)

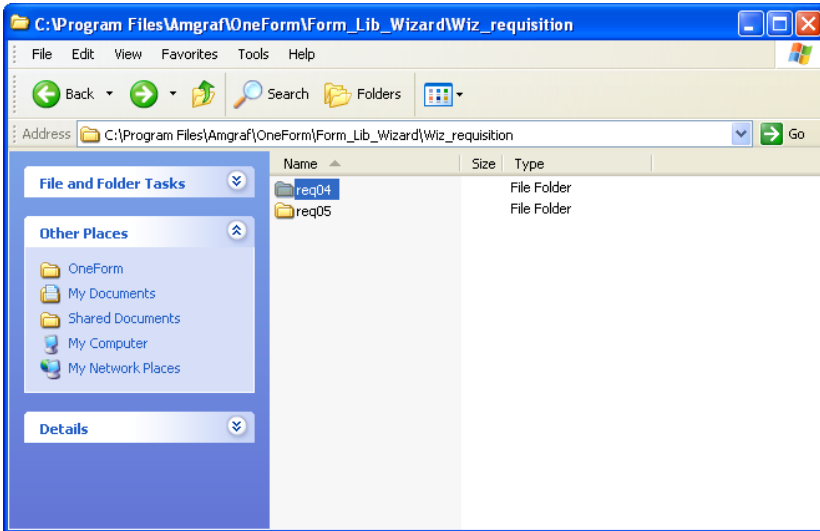


Figure 4-7: Needed Folder Contents

Test the hosting of the PDF Fillable Form by locating the Administrative Page on your Internet web server. My URL is:

serverpath/req04/admin_req04.html.

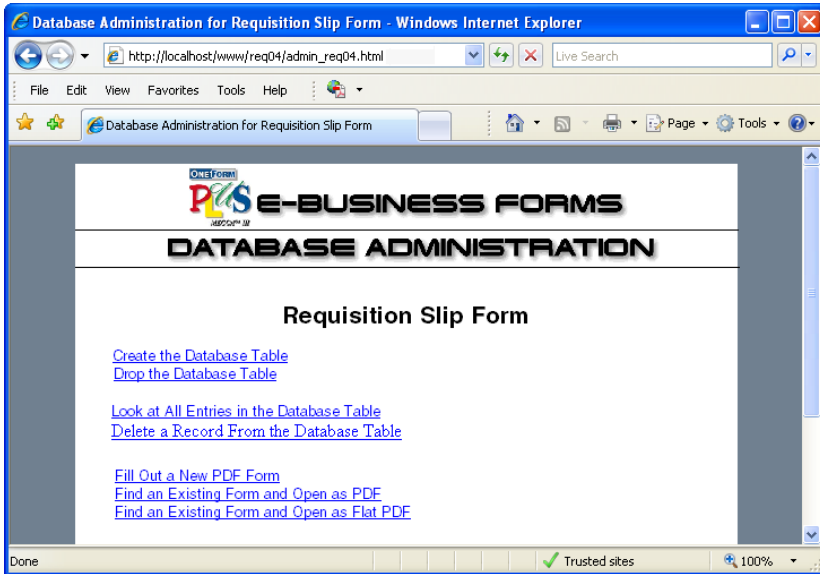


Figure 4-8: Administrative Page for Testing the PDF Fillable E-Form

Go through the standard testing procedure for E-Forms:

- Create a Data Table
- Fill and Submit Several Times
- Check Tabbing Order of Entire Form
- Search for Submitted Data

Note: Any database connection requires the server to know of the database through a "login.cgi". If you are experiencing problems at this stage refer to the *E-Forms Server Setup Reference Manual*.

Going direct to a local Access MDB file requires a "Direct to MDB" login.cgi, the ODBC Access driver, and two Perl Modules: Perl DBI and Perl DBD-ODBC.

CHAPTER 5

Adding Fillable Fields to the Original PDF

Save Form Fields as PDF

If you want to use an original PDF file and just add OneForm's fillable fields and database controls to it, use the **Save Form Fields as PDF** option instead.

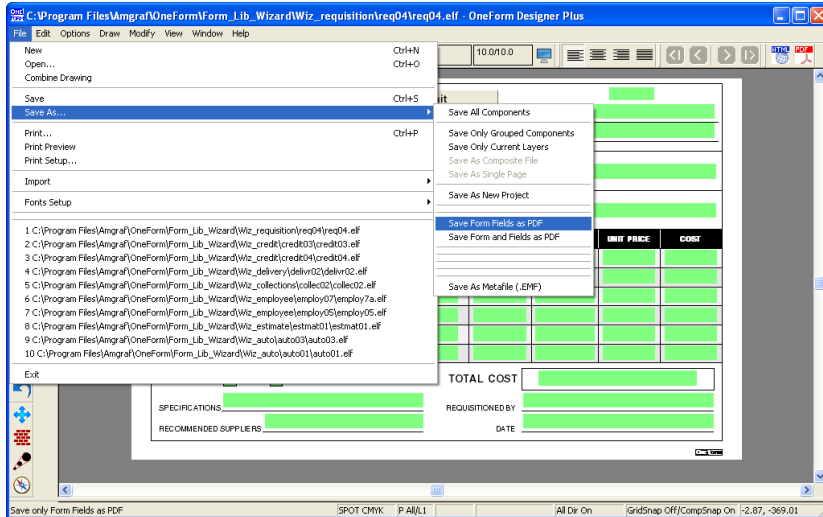


Figure 5-1: Saves the Fields on a Blank Page

This will only save the fill fields and JavaScript code into a blank page PDF file. Make sure to name the PDF file appropriately so that you know it is only fields by using "fields" in the name. You can then merge the original into it through Acrobat.

Note: The PDF file created has the page size of the drawing properties within OneForm Designer Plus and is always considered "Portrait".

Note: JavaScript coding inserted into the ELF file for form customization, intelligence and computation, will also be saved into the PDF. Examples of JavaScript for PDF Forms can be found in the OneForm Designer Plus Forms Library by going into the form's *EForm Preferences*.

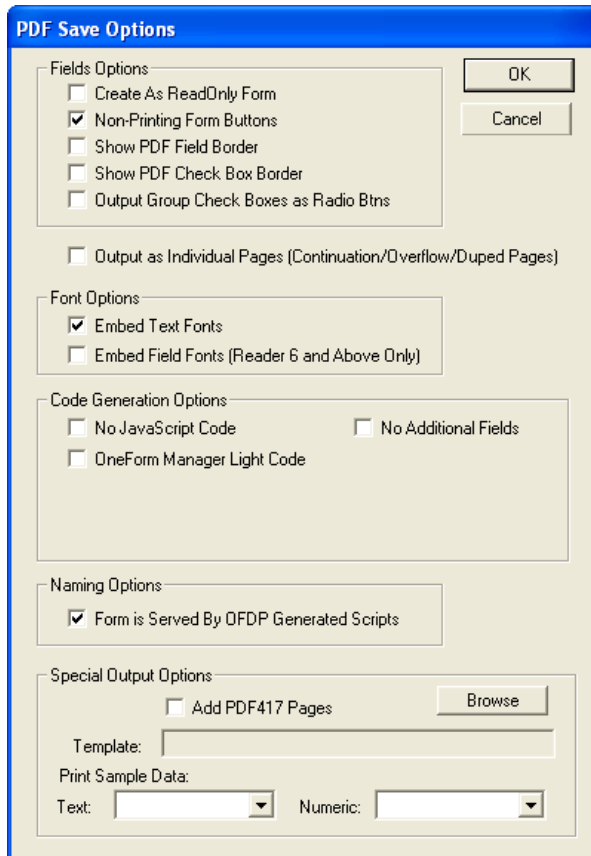


Figure 5-2: Use Default Settings

Our example is using the PDF default settings for saving the "req04fields.pdf".

Bring the Two PDFs Together with Adobe Acrobat

Launch the Adobe Acrobat Professional software and open the file named **req04fields.pdf** first. Click the **Form, Text Field Tool** to show the fill fields.

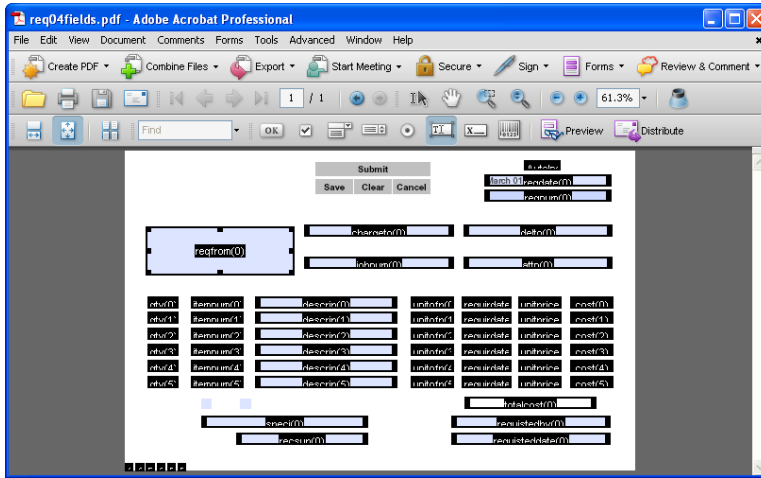


Figure 5-3: Open the Fill Fields PDF File in Adobe Acrobat

In the **Document** menu, use **Replace Pages** to select the original image PDF file. Click **[OK]** to combine the two PDF files together. You must always open the OneForm saved PDF with the fields first, and pull the form background artwork (the original PDF) into it. "Replace Pages" option ignores form fielding components, so you cannot use it both directions.

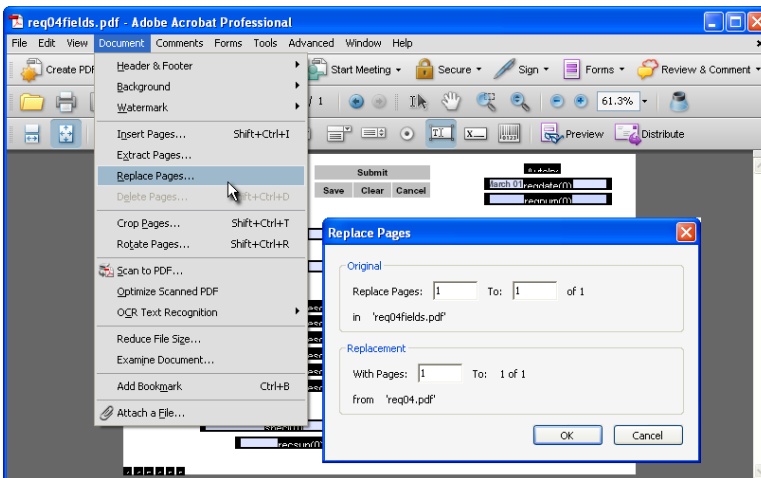


Figure 5-4: Use Replace Page Function to Combine Fill Fields with Form Image

Moving the Fill Fields

When the background appears, the fill fields may not be in the correct position. However, they will be relatively positioned and they can be easily selected and moved.

Select the fields by pressing the **Ctrl-A** keys and then move them where they belong. You can use the Click and Drag method of repositioning the selected fields or your "arrow keys".

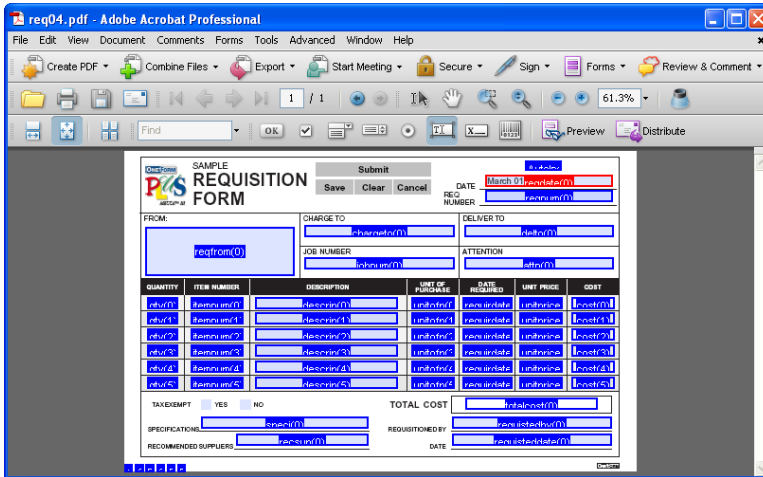


Figure 5-5: Select All the Fill Fields for Repositioning

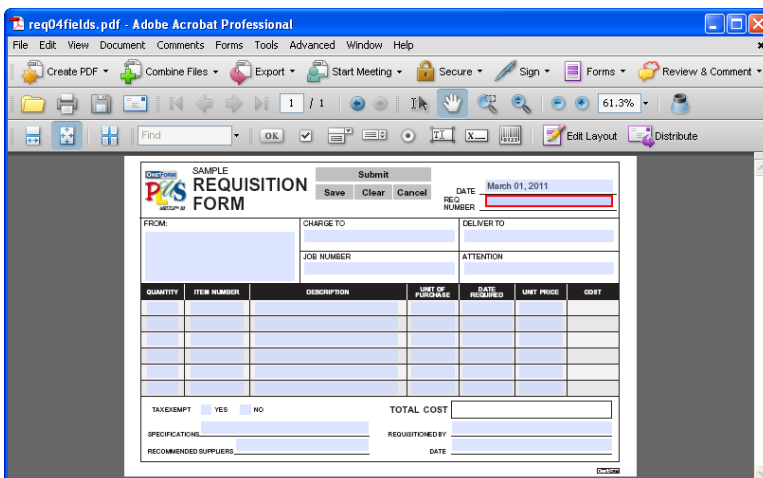


Figure 5-6: Click the Hand Tool to See the Finished PDF Fillable E-Form

Save As in Acrobat for Final PDF E-Form

It is necessary to resave the PDF file within Adobe Acrobat Professional before testing or placing on the server. It must be compressed and normalized to the PDF Version 1.4 standards (Acrobat Version 5.0) for our server scripts, otherwise as it opens you will see "file is damaged and can't be repaired".

Note: DO NOT Test the Final Form before resaving. It will save it with user data, etc.

Each version of Acrobat has a different menu for normalizing the file:

Adobe Acrobat Professional Version 5.0 – Use the File, Save As option and overwrite the existing PDF file using the filename assigned by OFDP.

Adobe Acrobat Professional Version 6.0 and Adobe Acrobat Professional Version 7.0 – Use the Advanced, PDF Optimizer option and overwrite the existing PDF file using the filename assigned by OFDP. The optimizer dialog has several settings that need to be looked at:

Preset: Standard

Make Compatible with: Acrobat 5.0 and later

Cleanup: Optimize the PDF for Fast web view

If you ever make any modifications to the PDF form, save it by using the normalize methods above.

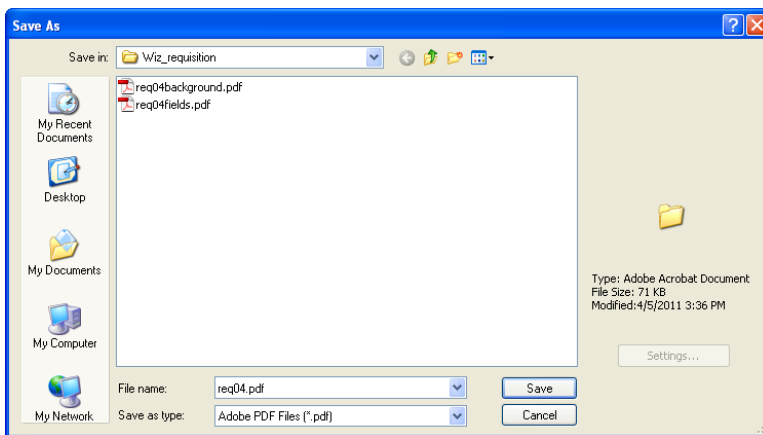


Figure 5-7: Final PDF with Basename of CGI Scripts

We named our final PDF "req04.pdf". The final PDF must always be the same as the basename of the CGI scripts, with the extension ".pdf".

Miscellaneous Information

Since the PDF form has turned out to be letter-size, you may wish to crop the image using the Document menu's Crop Pages option within Adobe Acrobat.

Controlling the PDF Page Size

OneForm Designer Plus always uses the Drawing Properties page size when generating PDF files and will always produce a "Portrait" page.

Combining background artwork with fields using the Acrobat *Document Replace* pages option, causes the page size to become the dimensions of the combined "Replaced Page".

When printing to a PDF file for the background art of the form, the defaults of the distiller printer properties controls the page size. Acrobat Distiller usually defaults to centering within an 8.5" x 11" page. At print time you can set it through the *Properties, Advanced, Custom Size option* and type in a page size.

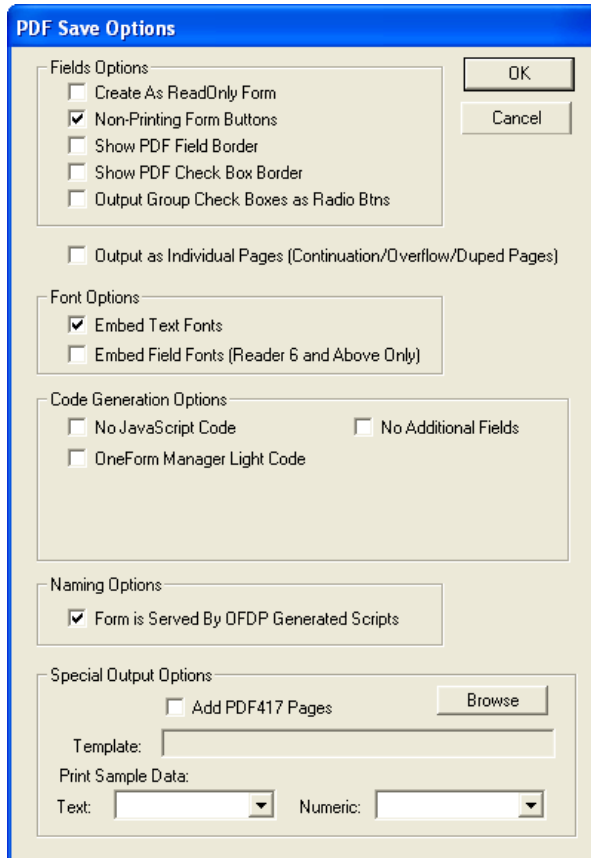
Adding JavaScript Programming

You may modify your PDF form and add custom JavaScript programming using the toolset in Adobe Acrobat. Keep in mind, however, that your OneForm Designer source files will not be updated with any of the changes made in Acrobat. Also, changes made to an Acrobat form are not active until the file has been optimized.

CHAPTER 6

PDF Save Options

Selecting any PDF Save option will take you to the following PDF Save Options Dialog.



The screenshot shows the 'PDF Save Options' dialog box with the following settings:

- Fields Options:**
 - Create As ReadOnly Form
 - Non-Printing Form Buttons
 - Show PDF Field Border
 - Show PDF Check Box Border
 - Output Group Check Boxes as Radio Btms
- Output as Individual Pages (Continuation/Overflow/Duped Pages)
- Font Options:**
 - Embed Text Fonts
 - Embed Field Fonts (Reader 6 and Above Only)
- Code Generation Options:**
 - No JavaScript Code
 - No Additional Fields
 - OneForm Manager Light Code
- Naming Options:**
 - Form is Served By OFDP Generated Scripts
- Special Output Options:**
 - Add PDF417 Pages
 - Template:
 - Print Sample Data:
 - Text:
 - Numeric:

Buttons: OK, Cancel, Browse

Figure 6-1: Default Settings

The dialog is divided into six sections. The options checked by default are normally wanted: buttons not printing, fonts embedded, and the file name in lowercase for OFDP server scripts.

Field Options

Create as ReadOnly Form – This option allows you to make a PDF ReadOnly version without having to change any properties in the fields.

Non-Printing Form Buttons – This causes the buttons to be visible but not print. This is checked by default.

Show PDF Field Border – This places a visible border around all Text, Numeric, Date, and ComboBox fields.

Show PDF Check Box Border – This places a visible border around Checkboxes.

Output Group Check Boxes as Radio Btms – This will use the Radio Button field type action for all Checkboxes. (They will still be square in the PDF.) It requires Advanced Properties Checkbox Group Options to be set in OneForm. And "Use Group as Single Data Value" within the Form Preferences Custom Features.

Individual Pages

Output as Individual Pages (Continuation/ Overflow/ Duped Pages) – This will output multiple page PDFs as single individual pages, (i.e., invoice_1.pdf, invoice_2.pdf) needed by the Amgraf Forms Portal.

Font Options

Embed Text Fonts – This will embed the Text Fonts used in the form into the PDF file.

Embed Field Fonts (Reader 6 and above only) – This will embed Field Fonts used in the form into the PDF. Only Reader 6 and above allow control over Field fonts.

Code Generation Options

No JavaScript Code – The PDF generator will dropout any JavaScript code when making the PDF. It will still have the fields on the form, but no custom JavaScript code placed under the CustomInfo Tab or Form Preferences.

No Additional Fields – If you are creating a PDF form just for PDF flattening, you do not need the additional fields added to the form for fillable PDFs.

OneForm Manager Light Code – This option allows a fillable PDF to open in OneForm Manager Lite.

Naming Options

Form is Served by OFDP Generated Scripts – The OFDP Generated Scripts require that the PDF filename be lowercase with no punctuation (other than the underbar (_) character).

CHAPTER 7

Adding the Finishing Touches for PDF E-Forms

Additional features can be added to your form that can enhance the usage of the form. Some features discussed in this chapter include:

- Setting the Tab Order
- Text Hyperlinks
- Adding Help Messages
- Checkbox Print Character
- Input Masks & Numeric Formats
- Adding Borders
- Changing Field Colors
- Using PDF JavaScript

Setting the Tab Order

The tab order is the sequence in which the user will move through the fields on the form when the [Tab] key is pressed. When the e-form is opened, the cursor will automatically begin at the first field in the tab order. As the user presses the [Tab] key, the cursor will move to the next field.

NOTE: The tab order must be set before the PDF is made. If the tab sequence has not been defined, an error message will appear. Minimum requirement is to *Sort Tabs in Reading Order*.

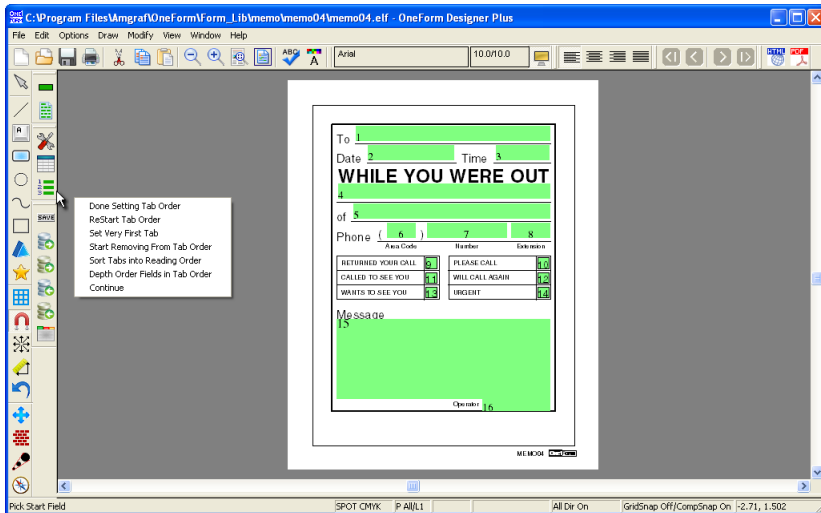


Figure 7-1: E-Form Set Tab Order

Sort Tabs in Reading Order

Step 1: Click the **Set Tab Order** tool. Notice that numbers will appear on the fields. These numbers represent the default tab order. They appear displaying the print properties font style, size, and composition mode.

Step 2: Open the pop-up menu with a **right-click**. Choose **Sort Tabs in Reading Order**.

Step 3: Go back to the pop-up menu and choose **Done Setting Tab Order**. This saves the tabbing order within the EForm Preferences.

Step 4: You can always see the tab order by going to the **View** menu and toggle the **EForm Fields, Show Tab Numbers**.

Resetting a Tab Sequence

Step 1: Click the **Set Tab Order** tool. The numbers represent the current tab sequence.

Step 2: Open the pop-up menu with a **right-click**. Choose **Restart Tab Order**.

Step 3: Next, **click on the field preceding** the tab you want to change. This would be the first correct tab number.

Step 4: Continue to **click on the fields in the order** in which you want the end-user to tab. Notice the numbers changing in sequence as you click.

Step 5: Once all fields in that sequence have been included in the tab order, right-click and choose **Done Setting Tab Order** to save the tabbing order.

Tab Order Pop-up Menu

The following options are found in the pop-up menu when working with Set Tab Order.

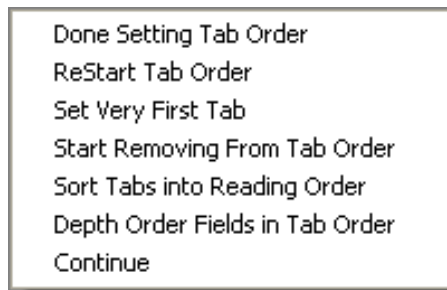


Figure 7-2: Tab Order Pop-Up Menu

Done Setting Tab Order: This ends each tab order session.

Restart Tab Order: This lets you create or edit the tab order. Only the fields that are clicked on will be modified. When restarting pick a good tab and proceed from there. Finish with a Done Setting Tab Order.

Set Very First Tab: This sets tab number 1. Always use this option to choose a different first tab.

Start Removing From Tab Order: Selecting this option and then picking on a field, will remove it from the tabbing order. It will have a 0 as its tab sequence.

Sort Tabs in Reading Order: This will set the tab order in natural reading order using each field's vertical position.

Depth Order Fields in Tab Order: The depth order of fields are as they are drawn. This will change their depth order relationship to be the tabbing order. This works well when you have fields on top of each other.

NOTE: Fields which are *Read Only* cannot be tabbed into. Their tab number will be 0.

Tabable Check Boxes

CheckBoxes by default are made tabable, but if they all have a 0 tab number, they may have been turned off. You can fix this by going through the following steps:

Step 1: Click the **EForm Preferences** tool.

Step 2: Click on the **[Edit Form Global Info]** button in the middle of the **Form ID Tab**.

Step 3: Select **PDF Forms** from the "Features for Form Type" dropdown list.

Step 4: Select **Tabable Check Boxes** from the "Available Features" dropdown list.

Step 5: Select **Y (for Yes)** from the "Feature Setting" dropdown list.

Step 6: Click **[OK]** to save this change into the file.

Step 7: Now when you go to **Set Tab Order** tool you can pick checkboxes. The program will recognize checkboxes as being tabable and assign them a tab number.

Adding Help Messages to Fields

Textual Help messages can be added to individual fields on your form. When the cursor hovers over a field in the PDF, the help text for that field will appear as shown below.

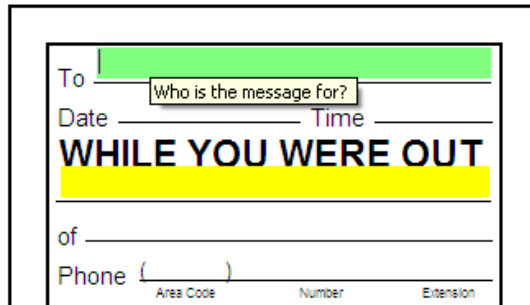


Figure 7-3: Hover Cursor Over Field for Help Message

Each field can have its own textual help message. This is under its EForm Field Properties. Input the text into the "Help Text" field on the first tab Field ID.

Adding Input Masks

A formatting mask can be added to Text and Numeric fields to control the look of the data after the user tabs out of a field. For example, putting hyphens into a Social Security Number or adding a dollar sign to represent currency.

Text Field's [Add Mask] Button Lists Common Masks

Text fields have a button specifically for this purpose. The **[Add Mask]** button found on Field ID Tab of a Text field brings up the following dialog with a list of common masks. Choose the mask type for your field.

The screenshot shows a dialog box titled "Add Input Mask". It contains the following elements:

- Field Name:** A text box containing "PhoneNumber".
- Field Type:** A text box containing "Text".
- Mask Type:** A dropdown menu showing "Phone2".
- Sub Type:** A dropdown menu showing "Mode0".
- Mask Qualifiers:** A section containing:
 - Min Chars:** A dropdown menu showing "7".
 - A list of mask options with "PhoneNumber" selected.
 - Three other empty dropdown menus.
- Mask Sample:** A text area showing "(000) 000-0000 or 000-0000" and "(Min Chars 0 allows 7 or 10 digits.)".
- Buttons:** "OK" and "Cancel" buttons in the top right corner.

Figure 7-4: Text Field's Input Mask Dialog

This above example is of the PhoneNumber field being formatted to 7-digits with a hyphen after the first 3-digits using the Phone2 mask. Usually the Input and Output Field is your current fieldname.

Common Text Masks

Bank (ABA Routing Numbers): This adds a check digit to an 8-digit routing number. (NNNNNNNNN) There are nine varieties of Bank formats.

City State: This pulls the text from separate City and State fields then inputs the comma. (AnyCity, AnyState)

City State Zip: This pulls the text from separate City, State, and Zip Code fields then inputs the comma. (AnyCity, AnyState NNNNN)

Date String: This formats the input into a date format.

Federal ID Number: This checks for 9-digit input and places a hyphen in the Federal ID structure (NN-NNNNNNN).

MECCA Units: Converts input into hundredths of points.

Pad Zero: Formats the number to a 6-digit padded with zeros.

Percent: Formats the input to a percentage.

Phone: Formats the input to a 10 or 7-digit with hyphens (000-000-000 or 000-0000).

Phone Style 2: This has four styles of phone formats requiring 7 or 10-digits:

Mode 0: (000) 000-0000

Mode 1: 000 000-0000

Mode 2: 000-0000-0000

Mode 3: 000 0000 0000

Proper Name: Converts input to Initial Caps.

Social Security Number: Converts input to the Social Security Number format (NNN-NN-NNNN).

Sub String: Requires two values: First Character and Length. It returns a sub-string starting at the first character value for length of characters.

Time String: Converts input into a time format (03:00 pm). It can pull from the computer's time.

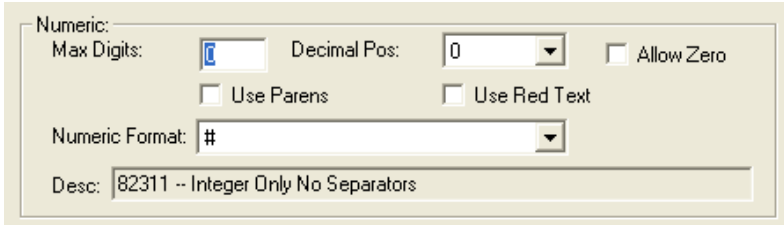
Zip Code: Formats input into a 5 or 9-digit zip code format (NNNNN-NNNN).

Zip Code Barcode: Adds Check Digits to a valid zip code format.

For currency formats make your field a Numeric Field Type and use one the its formats shown on the next page.

Numeric Format Examples

By default the Input Tab, Numeric format section allows unlimited numbers with no formatting. The # format is no decimals always, ignoring the Decimal Position field.



Numeric:
 Max Digits: Decimal Pos: Allow Zero
 Use Parens Use Red Text
 Numeric Format:
 Desc:

Figure 7-5: Input Tab's Numeric Format Section

Max Digits: is how many number keystrokes are allowed (0=unlimited). This does not include commas or the decimal characters.

Decimal Position: controls how many digits (0-9) allowed after the decimal. Requires a decimal numeric format to be selected from the list. Using a Decimal Position of 7 through 9 works with only the format shown below.

Numeric Format: A list of the available formats are shown below and how they relate to the maximum digits and decimal selections.

Default No Decimal	Max Digits	Dec. Pos.	Numeric Format
12345	0	0	#
\$12345	0	0	\$#
12,345	0	0	#,###
\$12,345	0	0	\$#,###
\$49,380	Total with Suppress Zeros		
With Decimals (1-6)			
1234567.89	9	2	#.00
\$1234567.89	9	2	\$#.00
1,234,567.89	9	2	#,##0.00
\$1,234,567.89	9	2	\$#,##0.00
\$4,938,271.56	Total with Suppress Zeros		
7 Through 9 Decimals Have Only 1 Valid Format			
1234.1234567	14	7	#.00

Figure 7-6: Standard Numeric Formats

The "Decimal Only" format is for HTML e-forms only. Checkboxes are available to allow zeros to display and control the look of negative numbers.

HTML Only - No Decimal/Decimal Character Displays

1,234,567.	7	0	#,##0.
\$1,234,567.	7	0	\$#,##0.
\$2,469,134.	Total with Suppress Zeros		

No Decimal w/Parentheses for Negative Numbers

(12,345,678)	8	0	#,###
(\$123,456)	8	0	\$#,###
(\$12,469,134)	Total with Suppress Zeros		

2 Decimal w/Parentheses and Red for Negative Numbers

(1,234,567.89)	9	2	#,##0.00
(\$123,456.78)	9	2	\$#,##0.00
(\$1,358,024.67)	Total with Suppress Zeros		

Figure 7-7: Using Parentheses and Red for Negative Numbers

Suppressing Zeros in Total Fields and the `_ThisForm` Notation

By default zeros are suppressed in numeric fields, but fields used as totals with code will display zeros. Suppressing zeros for total fields requires coding to be inserted. If you use the Calculation Wizard it will be done automatically. Below is an example of suppressing zeros. There are many examples in the Forms Library.

BeginPDFMainCalculation

```
// suppress zeros
  if ( _ThisForm["total1"].value == 0 )
    { _ThisForm["total1"].value = "" ; }
EndPDFMainCalculation
```

When using PDF JavaScript code to describe fields as shown above, we use a field naming method "`_ThisForm`". It is toggled in the EForm Preferences, Edit Form Global Info, Form Type - PDF Forms, Use `_ThisForm` Notation.

Changing Field Colors

Electronic Fields have default colors. For example, the user input is Black and a Green background on a field identifies the active field cursor location. We will discuss how you can change some of these field colors within your form.

Changing Field Background Colors

On the EForm Preferences Form ID Tab, is a variety of color options for the fields on your form. By default all field regular field backgrounds are transparent, the current field background is green, required fields are yellow, and read-only fields are white.

For PDF e-forms these options allow you to modify the Field Background, Current Field, Required Field, Read Only Field, and Button colors.

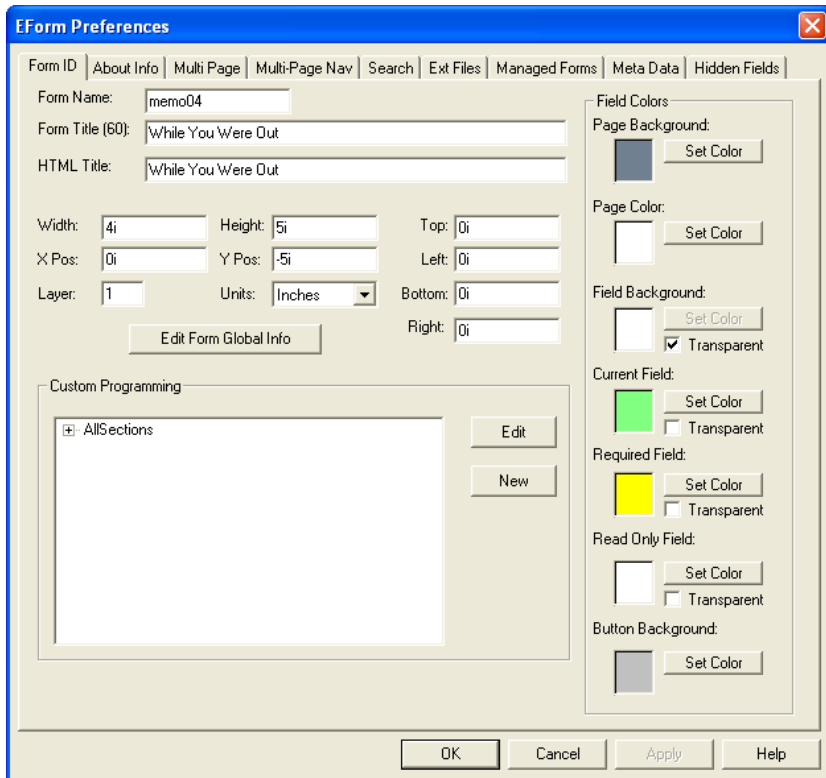


Figure 7-8: Changing Default Field Colors

Using PDF JavaScript: You can override the Print Tab's value for an individual field with JavaScript code. The background is the "fillColor". See the examples on the next page for using PDF JavaScript.

Changing Color of User Input

The default color for field user input text is Black. Changing color of the user input text is done through the field's Print Tab. It will be slightly different according to the color model.

Spot Color Model: If your form is using the Spot Color Model, you will use the Print Tab's "Layer" attribute within Field Properties to control the color of the user input. Default is Layer 1 Black. If you know the layer you want, just select it from the list. If you need to see the layer color choices, click on the Color [...] button. Hint: The file's Layer color assignments are controlled through the Drawing Properties, [Set Layer Colors] dialog. These then display in the Print Tab.

Process Color: Use the Print Tab's "Color" attribute within Field Properties to control the color of the user input. Default is our Four Color Process (FCP) Color 700, which is Black. If you know the FCP number for the color, you can type it in on the Print Tab FCP field. If you need to [Select a Color], a chart of available FCP colors will appear.

RGB Color: Use the Print Tab's "Color" attribute within Field Properties to control the color of the user input. Default is "000000" Black. If you know the Hex RGB value, type it in the RGB Field. If you need to [Select a Color], a color picker window will appear. Hint: If you want to paste in a Hex Value, do so in the Red field and press [Tab].

Using PDF JavaScript: You can override the Print Tab's value for an individual field with JavaScript code. The user input is the "textColor". See the examples on the next page for using PDF JavaScript.

Adding Text HyperLinks

The text markup commands HyperLink Start and End (\HLS^ and \HLE^) allows for a hyperlink to be embedded while in the Notepad text editor. Syntax follows:

```
\hls href="www.amgraf.com" target="_top" name="link1" id="link1" ^  
TEXT FOR LINK  
\hle^
```

Controlling the Printed Checkbox Character

By default the printed checkbox character during the flattening process is an "X". The control is found under Form Preferences, Edit Global Info. Go to Form Type of "PDF Forms" and locate the option "Checkbox Print Character".

Typing in an "8" will give you an "X", and a "4" will give you a "checkmark". This will put an environment variable in the PDF file: my \$CheckBoxChar = "4";

Adding Borders to the Fields

You can always draw borders on the form with the box option, but the Save As PDF dialog has two options for turning on field borders. One for regular fields and one for checkboxes.

Using PDF JavaScript: You can turn on the border for an individual field with JavaScript code. The border is the "strokeColor". See the examples on the next page for using PDF JavaScript.

Using PDF JavaScript Code

You can control the color or add intelligence to any field by placing code in either the Form Preferences or the Custom Info Tab of a Field. To override the Print Tab's value for an individual field, place code for basic colors within the Form Preferences. It will go within the begin and end for PDF main calculation. Example follows:

```
BeginPDFMainCalculation

// Changing Field Background with Basic PDF Color Names
_ThisForm["FieldName0"].fillColor = color.cyan;

// Changing Text Input Color with Basic PDF Color Names
_ThisForm["FieldName0"].textColor = color.blue;

// Adding a Field Border with PDF Color Names
ThisForm["FieldName0"].strokeColor = color.black;

EndPDFMainCalculation
```

Color Names: transparent, black, white, red, green, blue, cyan, magenta, yellow, dkGray, gray, ltGray.

The above example makes the field's background cyan, makes the user text blue, and places a black border around the field.

Recognized PDF Begin/End Strings

When coding custom JavaScript for PDF e-forms, following are the recognized Begin/End Strings.

BeginPDFBlur	BeginPDFFocus
EndPDFBlur	EndPDFFocus
BeginPDFCalculate	BeginPDFFunction
EndPDFCalculate	EndPDFFunction
BeginPDFEnter	BeginPDFMainCalculation
EndPDFEnter	EndPDFMainCalculation
BeginPDFExit	BeginPDFMouseUp
EndPDFExit	EndPDFMouseUp

CHAPTER 8

Duplication, Continuation, and Overflow Multi-Page Features for Flattened PDFs

There are times when the flattened PDF file needs to print more pages to hold the data captured in the HTML version of the form. This is why features such as duplication, continuation, and overflow for flattened PDFs were made available.

Duplication Multi-Page Feature

Duplication allows you to fill out the same form again without leaving the form and starting a new record. example a statement of work proposal is usually made up of many pages. The Duplication feature allows this to happen. The duplicated form is available in the HTML for end user data entry and also printed in the flattened PDF. An example of duplication is in the OFDP Forms Library – proposal/prop0102.

When an ELF is marked as a duplication page, it is dynamically displayed again under a page tab in the HTML when the [New Section] button is clicked. Its fields are associated to a sub-table that allows each row to represent an individual page. During printing, each sub-table record, is printed sequentially in the flattened PDF.

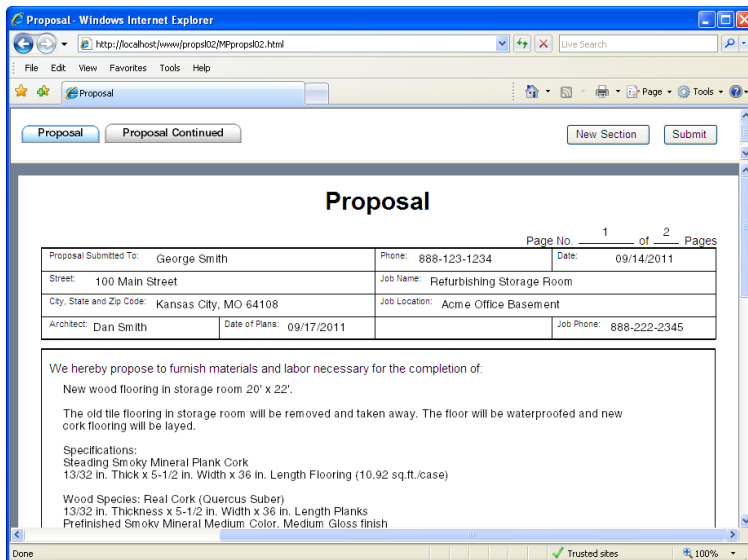


Figure 8-1: Duplication Example - HTML Data Capture

Duplication Multi-Page Feature

Proposal

Page No. 2 of 2 Pages

Proposal Submitted To: George Smith		Phone: 888-123-1234	Date: 09/14/2011
Street: 100 Main Street		Job Name: Refurbishing Storage Room	
City, State and Zip Code: Kansas City, MO 64108		Job Location: Acme Office Basement	
Architect: Dan Smith	Date of Plans: 09/17/2011	Job Phone: 888-222-2345	

We hereby propose to furnish materials and labor necessary for the completion of:
 New paint on all four walls and ceiling. New bookshelves along three walls of storage room.
 The entire storage room will be painted with latex Bright White paint, including ceiling.

Proposal

Page No. 1 of 2 Pages

Proposal Submitted To: George Smith		Phone: 888-123-1234	Date: 09/14/2011
Street: 100 Main Street		Job Name: Refurbishing Storage Room	
City, State and Zip Code: Kansas City, MO 64108		Job Location: Acme Office Basement	
Architect: Dan Smith	Date of Plans: 09/17/2011	Job Phone: 888-222-2345	

We hereby propose to furnish materials and labor necessary for the completion of:
 New wood flooring in storage room 20' x 22'.
 The old tile floor in the storage room will be removed and taken away. The floor will be waterproofed and a new cork flooring will be laid.

Specifications:
 Steading Smokey Mineral Plank Cork
 13/32 in. Thick x 5-1/2 in. Width x 36 in. Length Flooring (10.92 sq.ft./case)
 Wood Species: Real Cork (Quercus Suber)
 13/32 in. Thick x 5-1/2 in. Width x 36 in. Length Planks
 Prefinished Smokey Mineral Medium Color, Medium Gloss Finish

Construction: 3-ply Engineered Cork-HDF-Cork
 Planks have a microbevel
 Janka hardness rating: 830
 Installation: Floating
 Commercial use
 30-year wear/lifetime structural
 MFG Brand Name : Steading MFG Model # : ST9577

WE PROPOSE: hereby to furnish materials and labor - complete accordance with the above specifications, for the sum of:
 Two Thousand Dollars _____ dollars (\$ 2,000.00)

Authorized Signature: _____ All material is guaranteed to be as specified. All work to be completed in a substantial workmanlike manner according to specifications submitted, per standard practices. Any alteration or deviation from above specifications involving extra costs will be executed only upon written orders, and will become an extra charge over and above the estimate. All agreements contingent upon strikes, accidents or delays beyond our control. Owner to carry fire, tornado and other necessary insurance. Our workers are fully covered by Workmen's Compensation Insurance.

Note: This Proposal may be withdrawn by us if not accepted within 10 days.

Signature: _____ ACCEPTANCE OF PROPOSAL The above prices, specifications and conditions are satisfactory and are hereby accepted. You are authorized to do the work as specified. Payment will be made as outlined above.
 Signature: _____ Date of Acceptance: September 14, 2011

PROPSL02

the sum of: _____)
 0.00 _____)
 COM = \$5000.00

pleted in a substantial standard practices. Any costs will be executed and above the estimate. and our control. Owner is are fully covered by

ifications and condensed to do the work as

Figure 8-2: Duplication Example - PDF Flattened File

How Project was Created

1. The original form was drawn and fielded as usual.
2. Finishing touches such as Required fields, Set Tab order, Title, etc. was completed. ELF file saved.
3. Next we mark the ELF as a Duplication through the Multi-Page Tab. Highlight the Duplication Page row and right-mouse to get the pop-up menu. Select "Set Page to Allow Duplication". Then go to the Form Preferences

and setup the "User Navigation Bar for Single Pages". This is necessary to get the navigation bar when there is just a single page. It will hold our "New Section" button and page tabs. It is now setup for duplication and needs to be made server ready.

4. The standard required buttons for PDF forms was added and we are using AutoInc.
5. Now we assigned the output data tables to the fields. The Primary holds the AutoInc primary key field, for this example because the Sub-table holds all fields that are to be duplicated.
6. Duplication needs reference to external JavaScript files, and code in the Form Preferences for the navigation bar action, so I copied it into the file. (Get it from the propl02 sample).
7. The file was then processed. Generate the HTML and Scripts. Creating the PDFs is different. Save as the PDF twice. Once as normal and Once with Duplication checked. Use the same base name for the PDF as the ELF, both times. This creates two PDFs: one with the standard name and one with an _1. Do not optimize PDFs.

Testing Duplication Pages on the Web Server

You will notice the header has a New Section button, for opening the duplication HTML each time it is clicked. This allows you to key in additional forms. Be sure to save, submit, and print. You will notice the flattened printed document will be one page or multiple pages according to the amount of forms filled.

RefJavaScripts on the server needs continuation_pages.js, global_form.js, and xml_process.js.

Cgi-bin on the server needs NavTabsDisplay-pl.cgi.

Continuation Multi-Page Feature

Also, there are times when you need more rows of data to print, than you have physically on the printed page, such as an inventory form. Our Continuation multiple page feature is used for forms that hold rows of fields. There is a continuation sample in the OFDP Forms Library – requisition/req05.

The continuation page holds rows of stepped fields that are copied from the first page. They are used to hold the remainder of the rows that will not fit on the first page. Continuation pages are available in the HTML for end user data entry by clicking the [New Section] button. At the time of saving the data any blank rows will be removed. On printing, if the row data exceeds the number of rows supplied on the first page, the remainder will print on the continuation page. The design of the continuation page rows can be modified to be placed vertically higher on the page and hold more steps. Examples of continuation page usage would be an inventory form, a price list, or an order form.

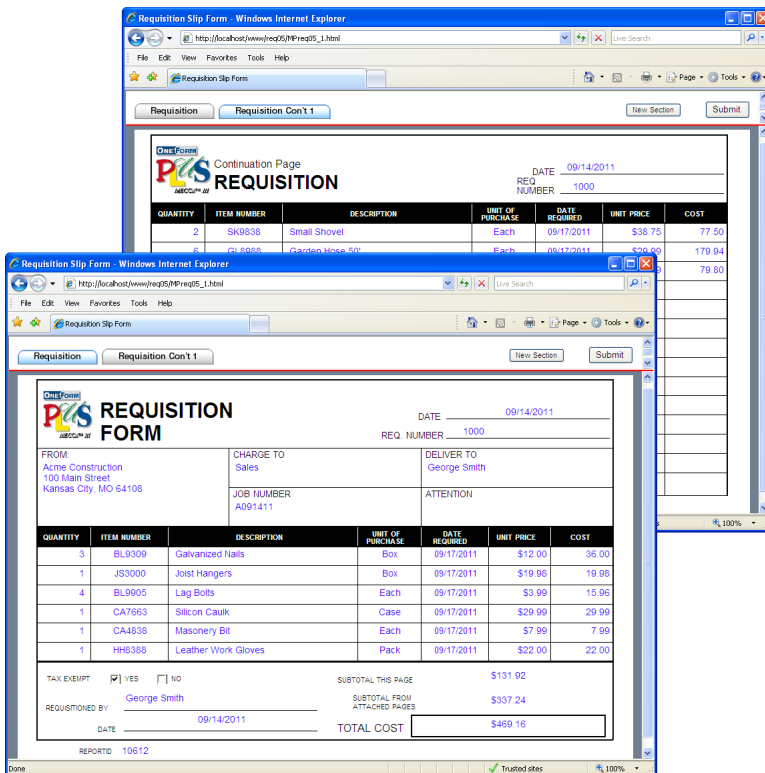


Figure 8-3: Continuation Example - HTML Data Capture

OneForm REQUISITION
Continuation Page

DATE: 09/14/2011
REQ NUMBER: 1000

QUANTITY	ITEM NUMBER	DESCRIPTION	UNIT OF PURCHASE	DATE REQUIRED	UNIT PRICE	COST
2	SK9838	Small Shovel	Each	09/17/2011	38.75	77.50
6	GL8988	Garden Hose 50'	Each	09/17/2011	29.99	179.94
20	PR3000	Bag of River Rock Boulders	Each	09/17/2011	3.99	79.80

OneForm REQUISITION FORM

DATE: 09/14/2011
REQ NUMBER: 1000

FROM: Acme Construction
100 Main Street
Kansas City, MO 64108

CHARGE TO: Sales

DELIVER TO: George Smith

JOB NUMBER: A091411

ATTENTION:

QUANTITY	ITEM NUMBER	DESCRIPTION	UNIT OF PURCHASE	DATE REQUIRED	UNIT PRICE	COST
3	BL9309	Galvanized Nails	Box	09/17/2011	\$12.00	36.00
1	JS3000	Joist Hangers	Box	09/17/2011	\$19.98	19.98
4	BL9905	Lag Bolts	Each	09/17/2011	\$3.99	15.96
1	CA7663	Silicon Caulk	Case	09/17/2011	\$29.99	29.99
1	CA4838	Masonry Bit	Each	09/17/2011	\$7.99	7.99
1	HH8388	Leather Work Gloves	Pack	09/17/2011	\$22.00	22.00

TAX EXEMPT YES NO

REQUISITIONED BY: George Smith

DATE: 09/14/2011

REPORTID: 10612

SUBTOTAL THIS PAGE: \$131.92
SUBTOTAL FROM ATTACHED PAGES: \$337.24
TOTAL COST: \$469.16

Figure 8-4: Continuation Example - PDF Flattened File

How Project was Created

1. The original form was drawn and fielded as usual, but used stepped fields for the continuation rows. (A stepped field Sub-table will hold the continuation data.) Finishing touches such as Set Tab order, Title, etc. was completed. ELF file saved.
2. Using "Save As in OneForm" to create a new ELF file, a copy of the file was made. It was renamed by adding "c" to the name (i.e. req05_1c.elf). The name is an identical copy of the original, followed by a "c" meaning "continuation".
3. Then the "C" ELF was opened and the cosmetic changes were made and tab order was reset. Things you can do: Remove any fields not needed that are not part of the stepped rows. If you do decide to keep duplicates of the regular fields from the original page, make them read-only. You can move the stepped rows if you want to edit their properties and add more steps. That is the only change you can make to the stepped fields. If you need to add any other regular fields, go ahead, but make sure they are associated with the Primary output table in step 7.
4. Next was linking the two ELF's together through Multi-Page. You open the original ELF and use the Multi-Page Tab to add a link to the continuation ELF (always placing it at the end). Modify both to add the Page tab heading. Highlight the Continuation Page row and right-mouse to get the pop-up menu. Select "Mark as Continuation Page" and choose the req05_1.elf as the page to continue. When saving and reopening the original page, it is a

multi-page project. It is now setup for continuation and needs to be made server ready.

5. The standard required buttons for PDF forms was added. One change is that continuation pages are required to use ReportID instead of AutoInc. And the SubmitData action needs a line of code added for this to work. (You can find both in our example file.)
6. Continuation needs the Form Preferences, MetaData Tab to have a Form GUID and a version number. Just clicking on the New GUID button will generate the unique GUID number. Then type in 1.00 as your version.
7. Now we assigned the output data tables to the fields. The Primary holds all fields except the stepped rows. The Sub-table holds only the stepped rows that are to be used for continuation.
8. Continuation needs reference to external JavaScript files, and code in the Form Preferences for the ReportID and GlobalVars, so I copied it into the file and edited the stepped field names. (Get it from the req05_1.elf sample).
9. The file was then processed. Generate the HTML and Scripts. Creating the PDFs is different. Save as the PDF twice. Once as normal and Once with Continuation checked. Use the same base name for the PDF as the ELF, both times. This creates three PDFs: both pages, page 1 only, and page 2 only. Do not optimize PDFs.

Testing Continuation Pages on the Web Server

You will notice the header has a New Section button, for opening the continuation HTML each time it is clicked. This allows you to key in additional rows. Be sure to save, submit, and print. You will notice the flattened printed document will be one page or multiple pages according to the amount of data. It will not print a blank continuation page.

RefJavaScripts on the server needs continuation_pages.js, global_form.js, and xml_process.js.

Cgi-bin on the server needs query-AutoNumber-pl.cgi, query-UserProfile-pl.cgi, query-UserID-pl.cgi, and NavTabsDisplay-pl.cgi.

Overflow Multi-Page Feature

There are times when you need the printed document to print more data than can physically fit in a field's space on the printed page. An example would be Text fields that need a scroll bar in the HTML for you to key in or read all of the text. In this instance, the form can be designed to use the Overflow multiple page feature to print the remainder of the text on a following overflow page. The Overflow example in the OFDP Forms Library – medical/medcal04.

An overflow page holds a scrolling text field that is copied from the original page to print overflow data that does not fit in the physical print space on the original page. Overflow pages are made available for the printing phase only. The data is keyed into the original HTML scrolling text field. When printing, the visible text that can be seen at the top of the HTML scrolling text field will print on the original page, the hidden remainder text will print on a following overflow page designed for this purpose. The text field is a copy from the original page, the width cannot change. The vertical position and depth can be modified. An example of overflow page usage would be an employee evaluation comment section. The comment field can start at the bottom of the original page and only show a few lines. On the overflow page, it can start at the top of the page and be extended to fill the entire page. When printing, the overflow page will only appear if the data exceeds the visible space on the original page.

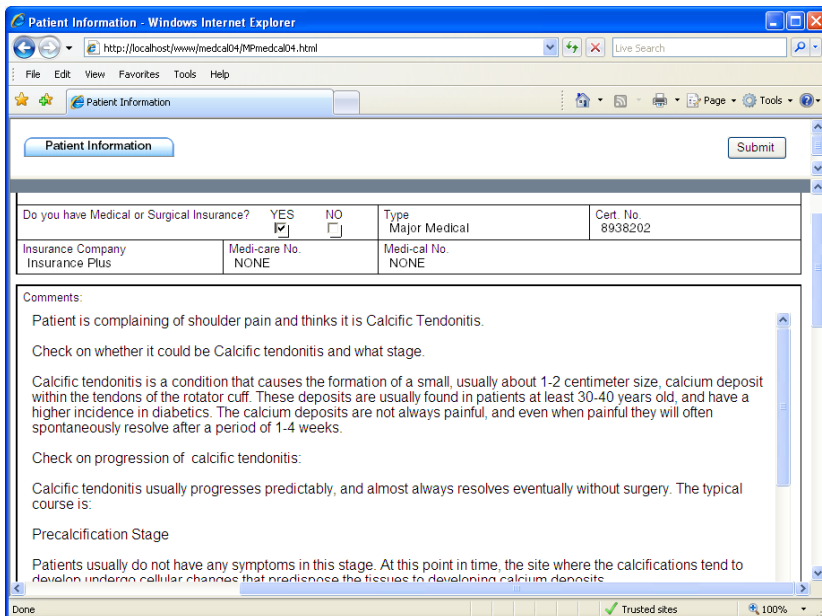


Figure 8-5: Overflow Example - HTML Data Capture in Scrolling Text Field

PATIENT INFORMATION (Cont.)

Date
09/14/2011

Patient (First, Middle, Last Names. Please Do Not Use Initials) GEORGE J. SMITH	Birthdate 03/10/1951
--	-------------------------

Comments:

Calcific Stage

During this stage, the calcium is excreted from cells and then coalesces into calcium deposits. When seen, the calcium looks chalky. It is not a solid piece of bone. Once the calcification has formed, a so-called resting phase begins; this is not a painful period and may last a varied length of time. After the resting phase, a resorptive phase begins--this is the normal appearing rotator

PATIENT INFORMATION

Date
09/14/2011

Patient (First, Middle, Last Names. Please Do Not Use Initials) GEORGE J. SMITH	Birthdate 03/10/1951
--	-------------------------

Marital Status				
Married <input checked="" type="checkbox"/>	Single <input type="checkbox"/>	Widowed <input type="checkbox"/>	Divorced <input type="checkbox"/>	Separated <input type="checkbox"/>

Home Address				
100 Main Street				
City	State	Zip Code	Home Phone	
Kansas City	MO	64108	888-123-1234	

Patient Employed By Acme Construction		Occupation Carpenter		
Business Address		Soc. Sec. #		
100 Main Street		333-333-3333		
City	State	Business Phone		
Kansas City	MO	888-123-1234		

Name of Spouse Jane Smith		Occupation Bookkeeper		
Spouse Employed By Acme Construction				
Business Address				
100 Main Street				
City	State	Business Phone		
Kansas City	MO	888-123-1234		

Patient Referred By
none

If Patient is Minor, Name of Responsible Parent

Do you have Medical or Surgical Insurance?		Type	Cert. No.
YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	Major Medical	8938202

Insurance Company	Medicare No.	Medical No.
Insurance Plus	NONE	NONE

Comments:

Patient is complaining of shoulder pain and thinks it is Calcific Tendonitis.

Check on whether it could be Calcific tendonitis and what stage.

Calcific tendonitis is a condition that causes the formation of a small, usually about 1-2 centimeter size, calcium deposit within the tendons of the rotator cuff. These deposits are usually found in patients at least 30-40 years old, and have a higher incidence in diabetics. The calcium deposits are not always painful, and even when painful they will often spontaneously resolve after a period of 1-4 weeks.

Check on progression of calcific tendonitis:

Calcific tendonitis usually progresses predictably, and almost always resolves eventually without surgery. The typical course is:

Precalcification Stage

Patients usually do not have any symptoms in this stage. At this point in time, the site where the calcifications tend to develop undergo cellular changes that predispose the tissues to developing calcium deposits.

MEDCALDA

Figure 8-6: Overflow Example - Two Page PDF Flattened File

How Project was Created

1. The original form was drawn and fielded as usual, but used a scrolling text field for the comments field. This is a field that holds more Input Lines than the physical depth can hold.
2. Finishing touches such as Required fields, Set Tab order, Title, etc. was completed. ELF file saved.

3. Using "Save As in OneForm" to create a new ELF file, a copy of the file was made. It was renamed by adding "o" to the name (i.e. medcal04_1o.elf). The name is an identical copy of the original, followed by a "o" meaning "overflow".
4. Then the "O" ELF was opened and the cosmetic changes were made and tab order was reset. Things you can do: Remove any fields not needed that are not part of the overflow field(s). If you do decide to keep duplicates of the regular fields from the original page, make them read-only. You can move the overflow field(s) up vertically if you want to modify their depth and make them deeper to allow more text to print. That is the only change you can make to the overflow field(s). If you need to add any other regular fields, go ahead, but make sure they are associated with the Primary output table in step 7.
5. Next was linking the two ELFs together through Multi-Page. You open the original ELF and use the Multi-Page Tab to add a link to the overflow ELF. Highlight the Overflow Page row and right-mouse to get the pop-up menu. Select "Mark as Overflow Page" and choose the medcal04_1.elf as the page to continue. It is now setup for overflow and needs to be made server ready.
6. The standard required buttons for PDF forms was added. One change is that the SubmitData button needs a line of code added for this to work. (You can find it in our example file.)
7. Now we assigned the output data tables to the fields. The Primary holds all fields except the overflow field(s). The Sub-table holds only the overflow scrolling text field(s).
8. Overflow needs reference to external JavaScript files, and code in the Form Preferences, so I copied it into the file. (Get it from the medcal04 sample).
9. The file was then processed. Generate the HTML and Scripts. Creating the PDFs is different. Save as the PDF twice. Once as normal and Once with Continuation checked. Use the same base name for the PDF as the ELF, both times. This creates three PDFs: both pages, page 1 only, and page 2 only. Do not optimize PDFs.

Testing Overflow Pages on the Web Server

Filling out the HTML is done as any other form. You will notice a scroll bar on the overflow text field. This allows you to key in additional lines of text than can be seen without scrolling. Be sure to save, submit, and print. You will notice the flattened printed document will be one page or multiple pages according to the amount of data in the scrolling text field.

CHAPTER 9

Saving to a Flattened PDF File Only

OFDP has an option for *Saving to a Flattened PDF File Only* where the e-form data is bypassing database tables. Instead the data is posted to the e-form's save script on the server and routed directly to the e-form's open script. At that time your browser displays an option (via a link) to merge the data and flatten the PDF file.

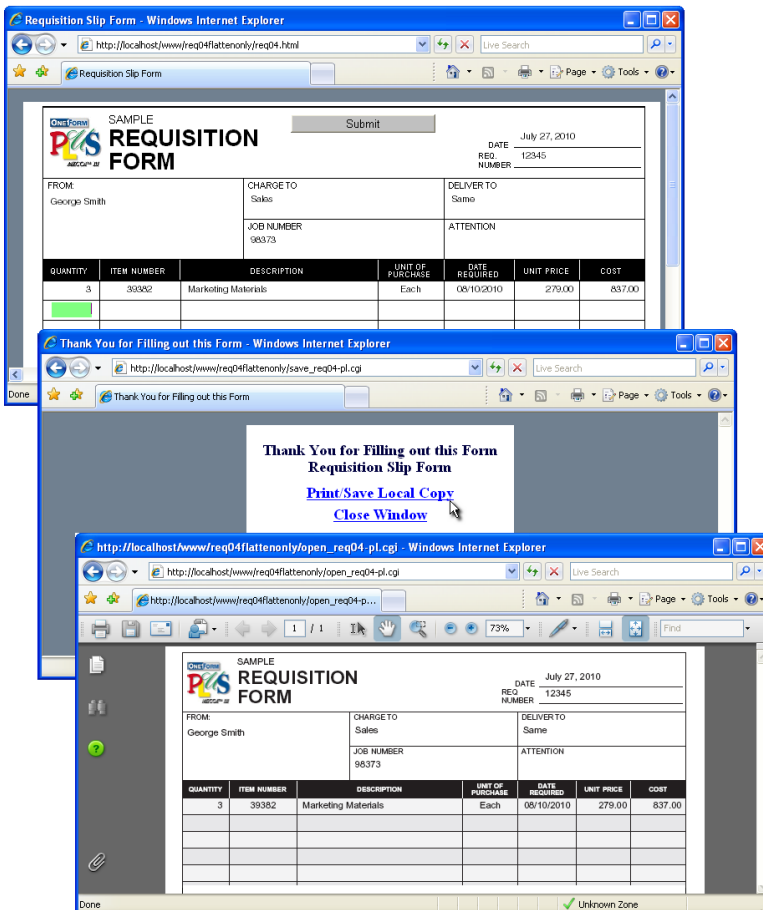


Figure 9-1: Submit Brings Up the Print/Save Local Copy

Clicking the Submit button brings up an HTML page with a link "Print/Save Local Copy". When you click on the link it will open the flattened PDF in Acrobat Reader for printing and/or saving a local copy through Reader.

In this chapter we will cover the steps for preparing your ELF for the *Saving to a Flattened PDF File Only* capability. This chapter assumes you have prior knowledge of standard OFDP database connected server-ready HTML and PDF e-forms.

Setting Up the Form

You will go through the regular steps for setting up your form by following the steps *Setting Up the Sample Form* as described in *Chapter 1*. Since we only need fields for data transfer and not management, you can also use our generic field naming (Field01, 02, etc.).

The required fields are: **SubmitData** and **FormStatus**.

Even though we are bypassing database tables, the setup for the form uses the database options for field assignments. If you don't already have an Output Key established, OFDP's **AutoInc** field will work.

The Field Data Assignments

OFDP's EForm Output Database options will be used to assign the form's data to fields, since they control the e-form's save CGI script.

Note for VB E-Form ELF Files Only: If you have a database connected form with a VB "Sequential Enabled" output key field attached to "23: SeqNum", you must turn off the Sequential Enable option in that field. Then use the EForm Output Database, Edit Fields/Columns option to add the new FormStatus field and reassign the Output Key field.

If you already have e-form database connections, you can go directly to *Generating the Server Side Scripts*.

Using the Output Database Option

The Output Database option "New Output DB Table" quickly assigns a database table/field to the form fields automatically. This is the option we will use to get our fields associated to their data. Our examples use a System DSN to locate the ODBC database. An ODBC system data source is visible to all users on the PC.

Step 1: Click the **New Output DB Table** tool and check the option **Use System DSN** and locate the existing database file from the DSN Name list. The New Table Name will automatically populate as the Primary table for the form. By default it uses the form name. Click **[Next]**.

Step 2: This will take you to the Output Fields Dialog for selecting what fields are to be captured. Select **[Add All Fields (No Sub Tables)]**. This will identify all fields on the form to have its own field column within the Primary table. Since we only need field associations, you can put all fields in the Primary table.

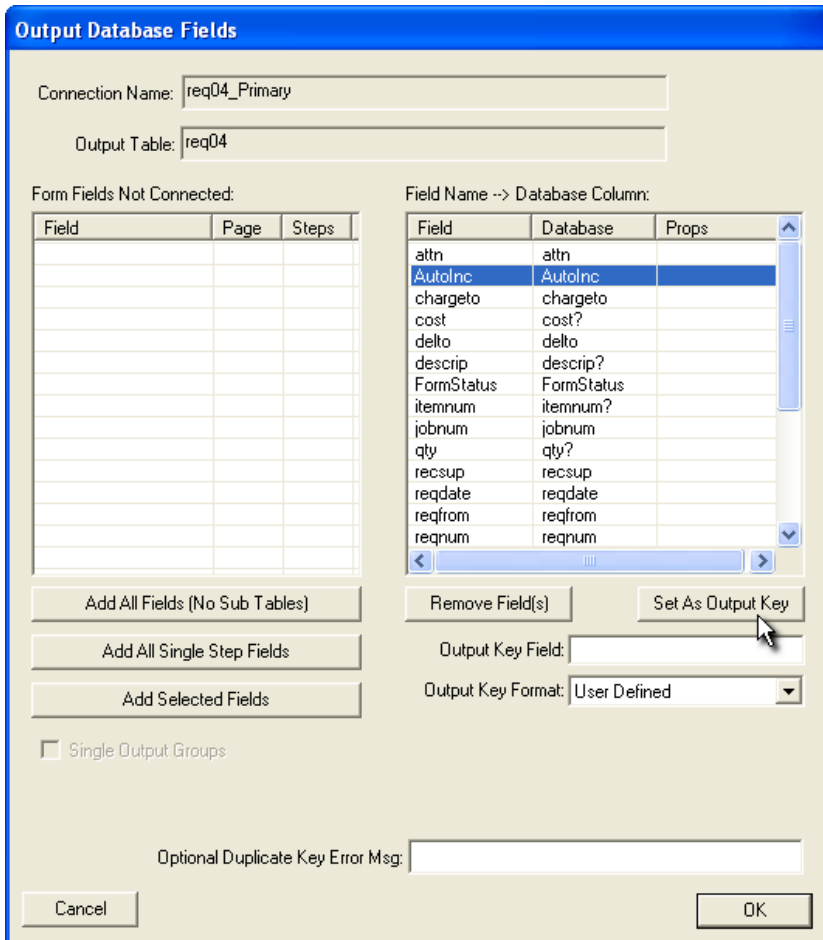


Figure 9-2: Output Database Dialog

Step 3: The fields in the right-hand window identify which fields are assigned. Highlight the **AutoInc** primary key field from the right-hand window, and click [**Set as Output Key**]. Click [**OK**] to assign the table to the fields and close the dialog.

What all of this did was update the Database Tab of those fields. If you open the field properties of *chargeto* and go to the Database Tab, you will see that it is associated to an Output database table with a DB field assignment.

Generating the Server Side Scripts

E-Form server scripts are generated as described in Chapter 14 of the OneForm Designer Plus User's Guide. Please refer to that chapter for more information. Here we will only cover the steps that are critical for saving to a flattened PDF file.

The Submit Data Button

Start by clicking the **Server Preferences** tool and go to the **Specify EForm Button** tab.

If you have used these tabs to create electronic HTML forms, then the next few steps will be familiar.

On this tab be sure to indicate that the **SubmitData** button is to **Submit All Data on Form** by clicking the checkbox as shown in the following illustration.

The screenshot shows the 'EForm Server Preferences' dialog box with the 'Specify EForm Button' tab selected. The 'Form Button' dropdown is set to 'SubmitData'. Under 'Button Display', 'Display as Button' is selected. The 'Submit All Data on Form' checkbox is checked, and the 'Path' field contains 'save_\$.frm.pl.cgi'. The 'Prompt User for Completion' checkbox is also checked. Below these are fields for 'Link to HTML File' and 'Send Data to Script'. At the bottom, there is a table for script arguments and buttons for 'Add New Param' and 'Remove Param'.

Argument Name	Field Name

Figure 9-3: Submit Data Button Settings

The Generate Scripts Tab

The **Gen Scripts** tab allows you to create the server scripts and browser HTML pages needed. They will be placed in your working folder. Make sure to check the Open/Save Option of **Save as PDF Flat Only**.

For the Form Open Options, you can choose how you want to capture the data. If you choose **HTML Open** as we did, you will need to supply an HTML E-Form by generating the HTML version of the form. Or you can have them fill out the fillable PDF (**PDF Open Option**) and then submit it for flattening. The **PDF Flat Open Option** is required.

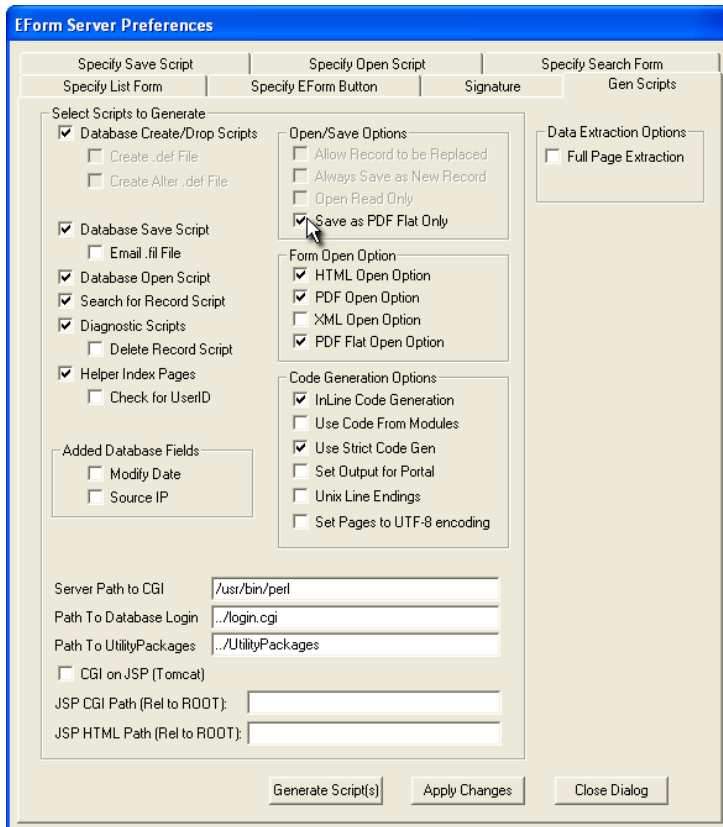


Figure 9-4: Generate Scripts Settings

[Apply] will save your changes to this tab. Click the [Generate Scripts] button to generate the server scripts.

A pop-up message will confirm that your scripts have been created. You can [OK] this and [Close] the EForm Preferences dialog.

Generating the Final HTML and PDF E-Form

Go through the standard procedures for creating the fillable PDF e-form. This will be used in the flattening PDF process, even if you are using an HTML e-form for data capture.

Creating the PDF E-Form

To create a fillable PDF e-form, click the **Save As PDF** tool.

This will create a complete PDF e-form with the form artwork components (lines, text, boxes, etc.) along with the fill-in fields.

[**Okay**] the default PDF Options. The final .PDF file **MUST be named using the same base form name** as the CGI files and reside in the same folder. This is so the CGI scripts can locate it.

It is necessary to resave the PDF file within Adobe Acrobat Professional before testing or placing on the server. Use either "Save As" or "Optimize."

For complete details see *Chapter 3: Creating the Fillable PDF E-Form*.

Creating the HTML E-Form

Click the **Generate EForm** tool. Choose "**Generate Code for" Electronic Form - HTML** and click [**Okay**].

This will create an HTML file that has instructions for the [Submit] button to open a flattened PDF file.

For complete details see *Chapter 14: Creating HTML Electronic Forms*.

Copy Files to the Server

It is time to copy the project folder and its contents to the document root on the web server and test from there.

Copy the files as directed by your Webmaster.

CHAPTER 10

Miscellaneous Information

Details on PDF Fields Used

You may notice that the generated OFDP PDF file includes some or all of the following fields.

SubmitData – A button field labeled "Submit" which executes the `save_form-pl.cgi` script when pressed.

SaveDraft – A button field labeled "Save" which executes the `save_form-pl.cgi` script when pressed for a save "draft" version. This button is optional but if using requires the hidden field *FormStatus* to be on the form and in the output database. See *Editing the Results Script to Ignore Drafts Saved*.

ClearForm – A button field labeled "Clear" which clears all data from fields without closing the form, allowing you to start over. This button is optional.

CancelBack – A button field labeled "Cancel" which quits the current form without saving and takes you to the `index_xxx.html` page. This button is optional.

ServerName – This is a required hidden field. It is used in the following items:

- `_InitSelection` – Concatenated with the relative URL for script.

- `_InitDependentFields` – Same as above.

- `_SubmitForm` – Tested to make certain it is not empty.

- [Cancel] Button

- [Submit] Button

When opening a form for either new or edit, the `open_form-pl.cgi` script will set the value of this field to the concatenation of the CGI environment variables "SERVER_NAME" + SCRIPT_NAME less the name of the `open_form-pl.cgi`.

FormStatus – This is a required hidden field. It is used in the following items:

- [Save] (Draft) Button – (value set to 2)

- [Submit] Button – (value set to 1)

This field is required to be present but its contents are only of concern if you are implementing the *SaveDraft* button. Since under normal operation local saving of a PDF is not allowed, this field conveys the intent of the user about the disposition of the data being submitted.

It is then up to the server to honor the clients intent.

PDFsource – This is a required hidden field. It is used in the following items:

- [Save] (Draft) Button – (value set to 2)

- [Cancel] Button – (value set to -1)

When opening a form for either new or edit, the `open_form-pl.cgi` script will set the value of this field to 1. Normal Submit will cause the value to be returned to `save_form-pl.cgi` untouched.

If the `save_form-pl.cgi` script finds the value to be non-zero, it will send back an `xdfd` response that merely contains a link back to the save script. This action will cleanly close the Acrobat helper function and have it start the `.html` response that is normally seen after completing a form.

Since the `save_form-pl.cgi` script is used for both `.html` and `.pdf` forms, it is important this field be present and value set properly or you will have the `.pdf` form appear to hang or the `.html` form start Adobe Reader for no reason.

FDfDataStatus – This is a required hidden field and is used in the `_SubmitForm`.

This field is used to provide the synchronization between the PDF form sending the request for data and the server providing the response with the data.

The field is set to 1 to indicate that we are in process waiting for data from the server. The FDF response from the server, will reset the value to zero and by virtue of this action will trigger the Calculate event for this field. The Calculate event will detect that we were in fact waiting for data from the server and pop off the top request for data.

If there are more requests for data pending, the next request will be initiated at this time.

PDFQueryDataParams – This is a hidden field. And can be used in the `_SubmitForm`.

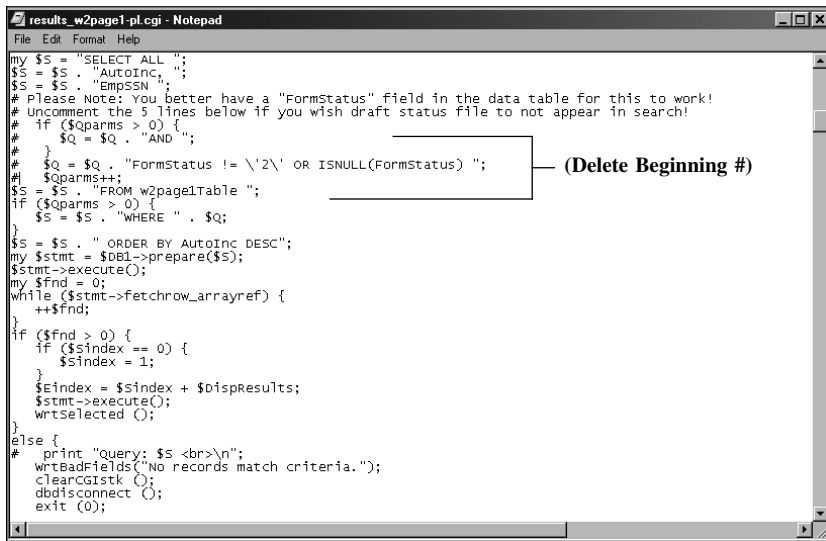
This field is used to provide to the server script all the parameter values that would normally be passed to the script as part of the request string (technically this is the HTTP GET request). Since PDF communicates back to the server only via a HTTP POST request, this is a way of putting the arguments into a predictable location.

The items are put into this field as `key=value` pairs and the server script is responsible for splitting the contents into separate variables.

Editing the Results Script to Ignore Drafts Saved

Earlier we mentioned that the [Save] button flags the data as being Saved to the database not Submitted. This can be used as a Save Draft.

The FormStatus hidden field is keeping track of items being Saved by flagging them with a 2. If you only want Submitted data to show in the Results list and not Drafts, you can edit the results list and uncomment five lines of code by deleting the pound signs. Looking for "FormStatus" will take you to the correct location.



```
File Edit Format Help
results_w2page1-pl.cgi - Notepad
my $S = "SELECT ALL ";
$S = $S . "AutoInc ";
$S = $S . "EmpSSN ";
# Please Note: You better have a "FormStatus" field in the data table for this to work!
# Uncomment the 5 lines below if you wish draft status file to not appear in search!
# if ($qparms > 0) {
#   $Q = $Q . "AND ";
#   $Q = $Q . "FormStatus != \'2\' OR ISNULL(FormStatus) ";
#   $qparms++;
# }
$S = $S . "FROM w2page1Table ";
if ($qparms > 0) {
  $S = $S . "WHERE " . $Q;
}
$S = $S . " ORDER BY AutoInc DESC";
my $stmt = $DBI->prepare($S);
$stmt->execute();
my $fnd = 0;
while ($stmt->fetchrow_arrayref) {
  ++$fnd;
}
if ($fnd > 0) {
  if ($$index == 0) {
    $$index = 1;
  }
  $Eindex = $index + $DispResults;
  $stmt->execute();
  wrtselected ();
}
else {
  # print "query: $S <br>\n";
  wrtBadFields("No records match criteria.");
  clearGistk ();
  dbd!$connect ();
  exit ();
}
```

Figure 10-1: Editing the "results_req04-pl.cgi" with Notepad

This will cause the results list to ignore any records with a FormStatus of 2. Then to see any forms which are drafts, search for 2 on the Search Form.

**Combining Fields onto Background Art PDF Pages
(instead of replacing background of fields)**

This is not the standard recommended procedure, but it can be tried if you absolutely can't go the other direction.

1. Open each of the PDF files (the fields and background), and show thumbnails on both. Window Cascade to see both files.
2. Go to fields thumbnail and drag over to background thumbnail area and make page 2. The drag will copy the page's javascript along with the fields.
3. Close the fields.pdf.
4. Click on the Fields Tool to show the fields, then select all (Ctrl-A).
5. Go to Tools, Form, Fields, Duplicate, All. This will copy the fields to the background page.
6. Delete page 2 thumbnail (fields only) through the right-mouse popup menu.
7. Move fields on page 1 to correct position if necessary.
8. Be sure to use the Optimize feature when saving.

CHAPTER 11

Using Import PDF

The import PDF option converts Adobe Acrobat 5.0 PDF files to OFDP ELF files.

Before starting your conversions, prepare the PDF for Import.

- Open the PDF in Acrobat Professional and look at the File's Document Properties and make sure that it has "No Security" as the Security setting. It does not import protected files.
- Make the PDF Acrobat Ver. 5.0 compatible. Our import option requires this step. Use the Advanced, PDF Optimizer option and overwrite the existing PDF file. Use the following settings: *Preset: Standard, Make Compatible with: Acrobat 5.0* and later.

Converting the PDF File

Importing the PDF file is easy. Go to the **File** menu and select **Import, PDF Files**. Browse for your file and **[OK]**.

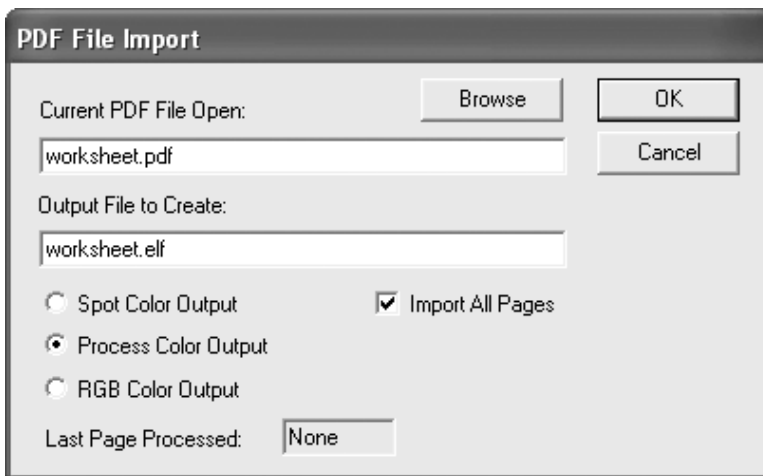


Figure 11-1: Browse for the PDF File

By default, it will import one page at a time as a Spot Color file. The naming convention is `basename_pagenummer_elf`. And it will place the file(s) in the same folder as the PDF.

Since PDF files can hold multiple pages, the next step is to **select which page to import**. To get a list of pages, click onto the plus sign (+) to expand the list. Then highlight a page and select **[OK]**. The ELF file will be opened in OneForm. Before working on the files, open each one and use **Save As** and save over the existing ELF.

When working with PDF Import, you have a choice of Spot Color or Process Color models. Unless you need separations, Process Color would be the better choice. If "Import All Pages" is chosen, the ELF files are saved and Page 1 is loaded in OFDP immediately, bypassing the PDF dialogs.

Font Setup for Import

If you need to map your fonts for the first time, or know you will need to index a new font, do not choose "Import All Pages". This will take you to the PDF Import Tabs, after selecting a page.

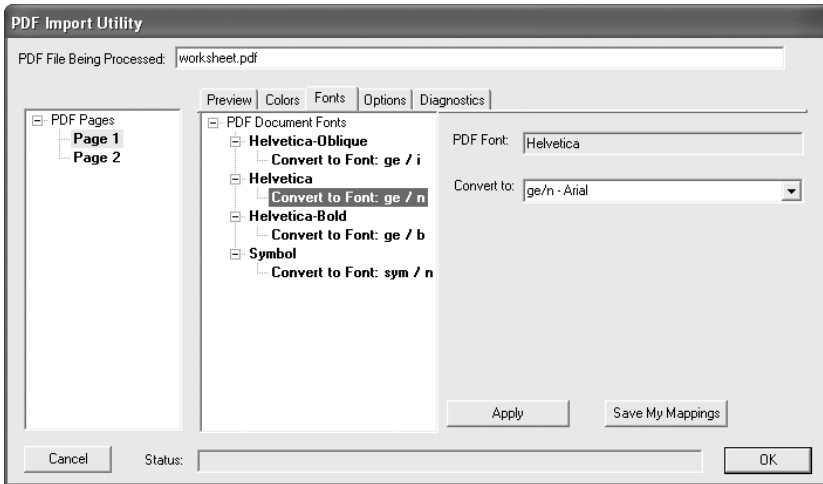


Figure 11-2: The Font Tab

In the tab for Fonts, it will list the fonts being used during the conversion. Just click on the plus sign (+) next to "PDF Document Fonts", and then each succeeding one to see each font.

If the font does not exist on your system, it will default to "(ge/n) Arial". Clicking on the "Convert to Font ge/n" will display the font name associated with the font mnemonic.

Compare the "PDF Font" to the "Convert To" font being mapped. You may index it to another font of your choice and then [Apply].

If the font map is saved with [Save My Mappings], it will be used as the default on the next import.

Miscellaneous Information

Before working on ELF files created by import, do a **Save As** and save over the existing ELF or rename it using OFDP filenames conventions. If the PDF had fields on it, they will come in as our TEXT Field type. You can now modify the fields, or add any others needed.

CHAPTER 12

PDF Files for Digital Printers

The preferred way to make PDF files with OFDP is to use the red PDF button located on the right side of the top toolbar. As an exercise to use this method, launch OFDP and start a new drawing. Name the drawing *PrintTest.g*.

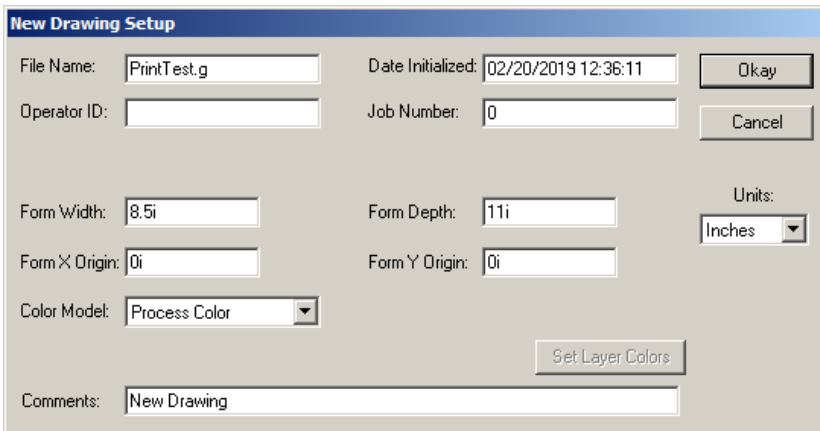


Figure 12-1: Start a New File

Draw a few lines and create a piece of text. Then click the red PDF button on the right side of the top toolbar. A file list panel will open as shown below.

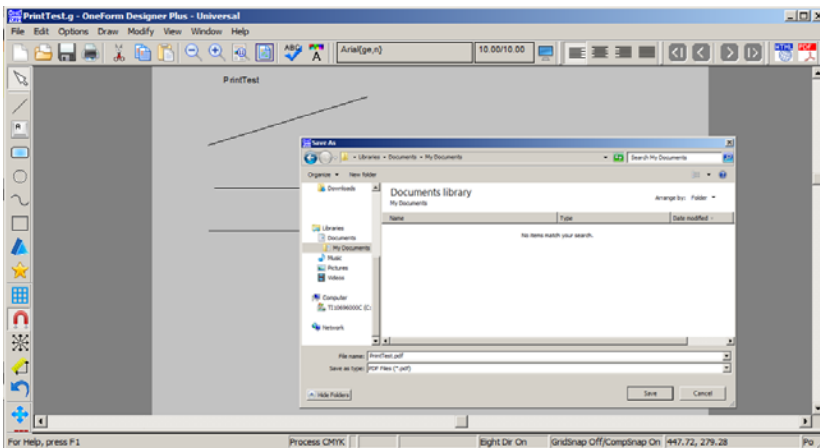


Figure 12-2: Click the PDF Button on Top Toolbar

Navigate to a location for storing the PDF file and click the [Save] button.

The "PDF Output Props" dialog then opens as shown below. Click the [OK] button to make the PDF file.

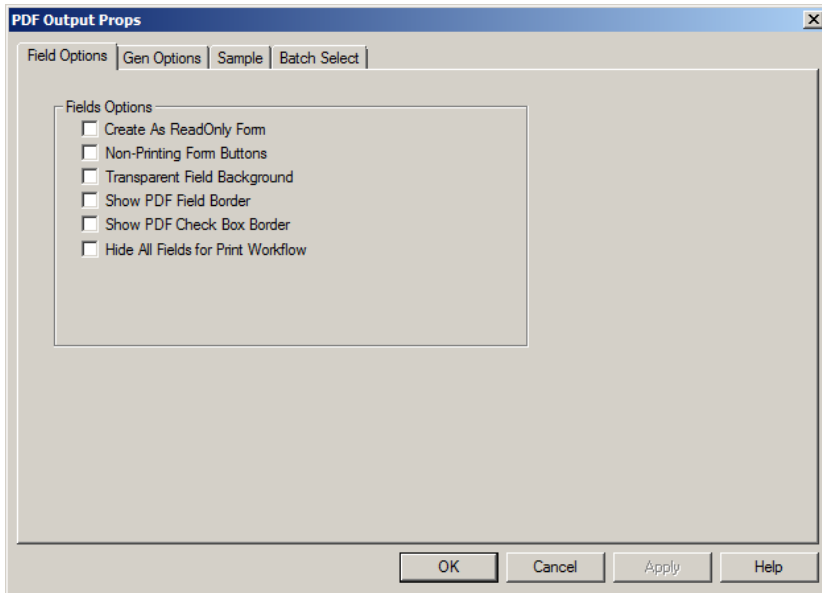


Figure 12-3: Click [OK] to Make a PDF File

The options above apply to making PDF files with electronic form fill fields. You can hide them if needed.

Now use Windows Explorer to locate and open the resulting PDF file in Adobe Acrobat.

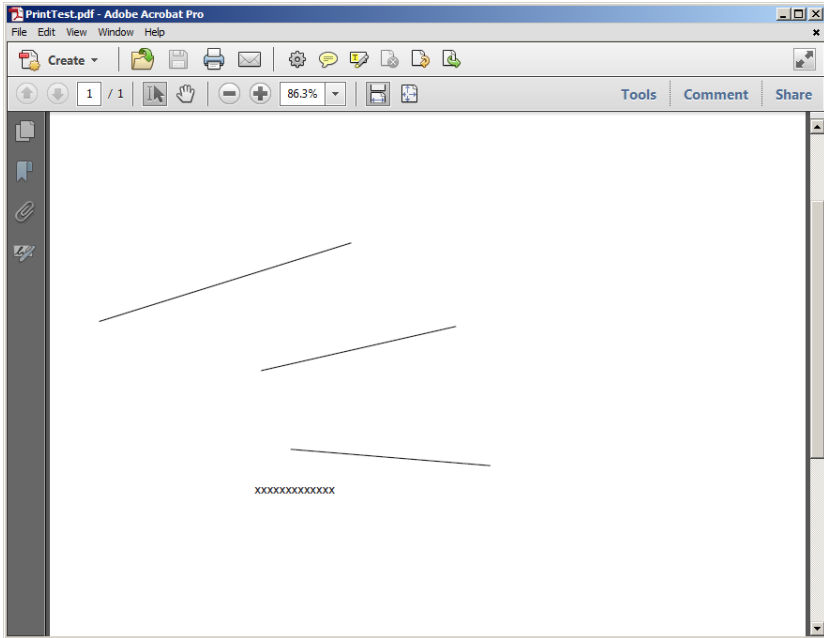


Figure 12-4: Browse for the PDF File

Note: If you make any choices in the "PDF Output" dialogs, you should re-save your drawing so that the choices will be pre-set the next time you need to make another PDF.

Making PDFs for Spot Color Files

For Spot Color drawings, you will need to add a Spec File to your drawing and then choose options from the *PDF Output Props, Separations* dialog. Let's start with the Output Specifications for Spot Color.

Parts Output Specifications for Spot Color

Color separations for printing are controlled by the *Drawing Options, Output Specifications, Parts* dialog. Here you indicate which layers are needed for each separation.

The screenshot shows the 'Spot Color Plot Specs' dialog box. At the top, it displays 'Part: 1', 'Max Separations: 2', 'Choke/Spread: (mils)', and 'Description: Trim Marks On'. Below this is a grid for selecting layers for each separation. The grid has 32 columns representing layers and 8 rows representing separations (Sep 1-8). The 'Dark' column has a value of 700. The 'FCP#' column has values of 185, 0, 0, 0, 0, 0, 0, 0 for Sep 1-8 respectively. The 'Steps' row has checkboxes for each layer. Below the grid are fields for 'Steps Across/Around: 0 / 0', 'Step Size Across/Around: 0 / 0', 'Scale X/Y: 0.00 / 0.00', and 'Units: Inches'. At the bottom, there are sections for 'Layer Status Legend' and 'Layer Printing Functions'. The 'Layer Status Legend' includes 'Layer is Ignored', 'Layer is Printed', and 'Layer is Knocked-Out'. The 'Layer Printing Functions' includes 'Print Positive, K-O Others', 'Knock-Out int this Sep', 'Print Positive, IGN Others', 'Ignore in all Seps', 'Static Layer (Pos All Seps)', 'Clear All Layers', and 'OK'.

Figure 12-5: Specifications for Spot Color

This is an example of a 2 color project with 3 layers being used. You simply click on which layers are needed for each color separation. Notice that layer 4 prints on both separations.

Separation 1 represents BLACK - Layer 1 is the Black Artwork, Layer 4 is the Page Trim/Cut Marks

Separation 2 represents RED - Layer 2 holds the Red Components, Layer 4 is the Page Trim/Cut Marks

Creating a Simple Part Specification

Choose Add Part to create a new part and a prompt to name the part will appear. Once you have named the part, the Output Specification Dialog box will appear.

Max Separations: This field tells the system how many separations this definition will print

Description: This holds the part name.

Sep 1 through 8 Layers: Check which layers you want turned on for each separation needed. Only layers turned on will print.

FCP#: The color number this separation represents. This is for reference only.

Making the PDF for Separations

When you are making PDF files for spot color jobs with a Spec File for two (or more) colors, a Separations Tab will be available on the "PDF Output Props" dialog. Go to the Separations Tab and select the checkboxes for outputting separations as shown below:

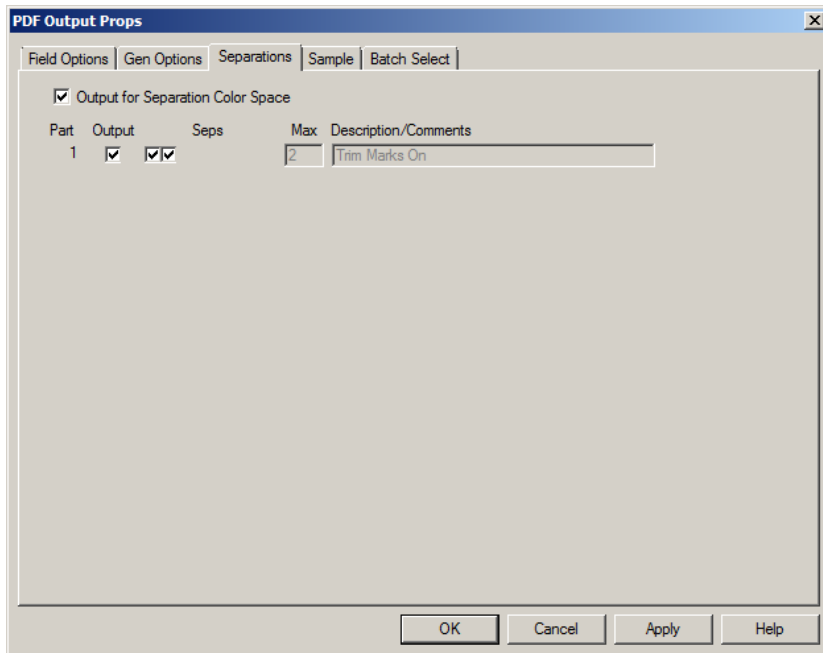


Figure 12-6: Separations for Spot Color

This will place colors in the PDF file to be used for the Spot Color Separations.

The PDF file is now showing the spot colors needed for separations as shown below.

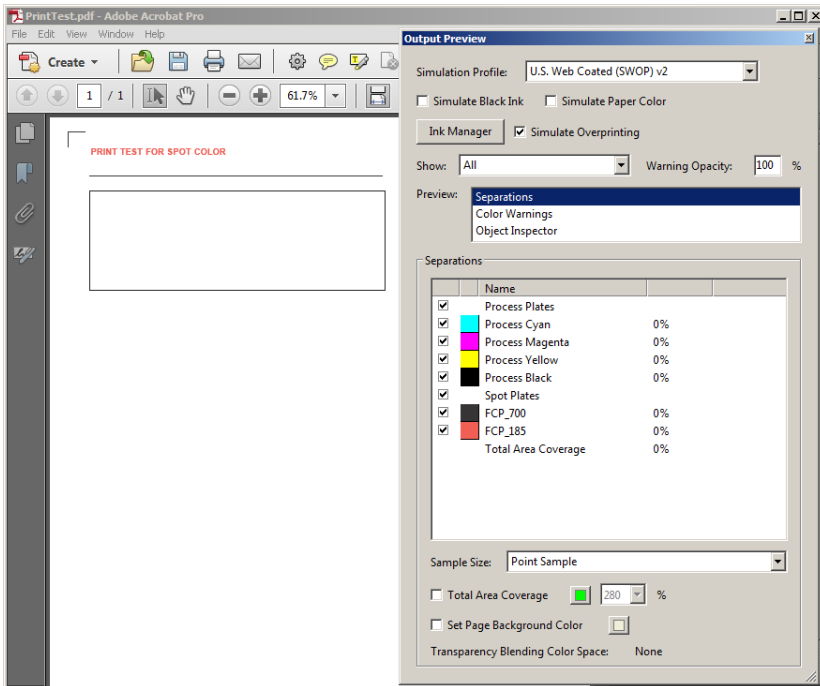


Figure 12-7: Example of Spot Color PDF

INDEX

Add

- Fillable Fields to Original PDF 29
- Finishing Touches 39
- PDF JavaScript Programming Code 48
- Required Submit Buttons 4

Adobe Acrobat

- Bring PDF Together 31
- Save As 19, 33

Assigning

- Output Database Primary Table 9
- Stepped Fields SubTable 13

Background, Combine Fields 68

Button

- Add Required 4
- Move Suitable Location 7
- Submit Data 24

Change Field Color 46

Checkbox Character 47

Combining Fields, Background 68

Continuation Multi-Page Feature 52

Controlling PDF Page Size 34

DSNs, Use System 10

Data Submit Button 4, 24

Database, Assign Output Tables 9

Default Settings for PDF 18

Drafts, Ignore Saved 67

Duplication Multi-Page Feature 49

ELF, Open Sample 3

Editing Results Script 67

Fields

- Add Fields to Original PDF 29
- Assign Stepped Fields to SubTable 13

Field Color 46

- Combine Fields onto Background 68
- Move Field Position 32
- PDF Save Options 39
- Save Form and Fields as PDF 29, 35
- PDF Fields Details 65

Fillable PDF Form 17

Final, Save As in Acrobat 19, 33

Flattened PDF Multi-Page Features 49

Flattened PDF Only, Save As 59

Font Options 36

Form

- Create Fillable PDF 17
- Fields, Save 17, 29
- Copy to Server 27
- Sample 3

Generate Scripts Tab 21, 25

Help Messages 41

Hosting, Form Server 27

HyperLinks 47

Ignore Drafts Saved 67

Input Masks for Fields 42

JavaScript Code 48

Miscellaneous Information 65

Move

- Buttons Suitable Location 7
- Fill Fields Position in PDF 32

Multiple Pages 20

Numeric Formatting 44

Options in PDF Dialog 36

Output Database Table 9

Overflow Multi-Page Feature 55

Page, Control PDF Size 20, 35

Position, Move Fill Fields 32

Preferences, E-Form Server 21

Programming, Add JavaScript 35, 48

Required

- Submit Buttons, Add 4
- Server Setup 2
- Software 2

Results

- Edit Results Scripts 67

Sample 3

Save As

- Save Form and Fields as PDF 17, 29, 35
- Save Draft 67
- Save Final PDF E-Form 19, 33

Save As Flattened PDF Only 59

Scripts

- Generate Server Scripts 21
- Generate Tab 25

Server

- Copy Folder 27
- Form Hosting 38

Index

- Generate Server Scripts 21
- Server Setup Requirement 2
- Size, Control PDF Page Size 20, 34**
- Software Requirement 2**
- Special Output Options 37**
- SubTable, Assign Stepped Fields 13**
- Submit**
 - Buttons, Add Required 4
 - Data Submit Button 24
- System DSNs, Use 10**
- Tab Order**
 - Set Tab Order 8, 39
 - Table Check Boxes 41
- Table**
 - Assign Output Database 9
 - Assign Stepped Fields 13