AMGRAF, INC.

SOFTWARE TECHNOLOGY FOR E-BUSINESS FORMS

ONE1FORM™ DESIGNER

PLUS

MECCA™ III

FORMS COMPOSITION FEATURES

E-BUSINESS FORMS
ELECTRONIC FORMS
FILLABLE PDF FORMS
PAPER BUSINESS FORMS

Creating E-Forms

OneForm™ Designer Plus

These specifications are subject to change without notice.

AMGRAF
SOFTWARE TECHNOLOGY
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

Chapter 1: Overview

Chapter 2: Tutorial – Simple Fill-In E-Form

Chapter 3: Drawing and Modifying Electronic Fields

Chapter 4: Field Types

Chapter 5: E-Form Field Properties

Chapter 6: E-Form Preferences Tabs

Chapter 7: Adding the Finishing Touches for E-Forms

Chapter 8: Associating Databases to the E-Form

Chapter 9: E-Form Server Preferences Dialogs

Chapter 10: Creating HTML E-Forms

Chapter 11: Electronic Forms Local File Save

Chapter 12: HTML E-Forms with Signature Fields

Chapter 13: The Calculation Wizard Option

CHAPTER 1

Overview

With Amgraf's OneForm Designer Plus (OFDP) Software, one file is all that is needed for the conventional printed form and electronic form. A single design file can be published to multiple media formats as needed. It can have both HTML and PDF e-form capabilities. OneForm gives you the tools to design and implement a workflow solution for all of your business needs.

Levels of Electronic Forms

There are three levels of electronic forms:

- **Print-on-Demand,**
- **Simple Form-Fill, and**
- **Intelligent Forms with Database Links.**

Each level progresses to give more control over the use of the form, while all levels allow viewing and printing.

A **Print-on-Demand** e-form could be any form which has been drawn or imported into OneForm Designer. It does not have any electronic form field components. This form will be limited to only viewing and printing.

Examples of **Simple Form-Fill** e-forms are those in our Forms Library. These forms have electronic form fields for fill-in information, but are not linked to any input or output databases.

Intelligent Forms have database links which help to gather and retrieve information as it is needed. You can also have intelligence, such as calculations, or field validations, take place when necessary.

Electronic forms are generated as HTML pages with JavaScript coding for intelligence. By using the optional PDF Forms Generator Module option, PDF e-forms can also be created from a single OFDP file.

Software to simplify the online hosting of e-forms is also included with OneForm Designer Plus. Internet server control scripts are automatically generated for the rapid deployment of database-connected HTML and PDF forms. The highly efficient Common Gateway Interface (CGI) scripts are compatible with Microsoft and Unix web servers, and support most relational databases including those from Microsoft, Oracle, and MySQL.

Server script generation is very easy. The E-Form Server Preferences panel is used to quickly specify all of the server CGI scripts necessary to manage databases; open, save, and e-mail e-forms; find and list saved e-forms; and to perform XML data exchanges.

Basics for Creating an Electronic Form

Below are the procedures to convert any existing form into an intelligent electronic form. We will discuss each procedure in detail in the following chapters.

Draw or Import the Form: Take your already composed form and load it into OFDP. You can open a form composed in Amgraf's OneForm, OneForm Designer Plus or MECCA software. OFDP will open either (.g) graphic files or (.elf) electronic files. OFDP can import from other software files in the types of PostScript (.ps), Encapsulated PostScript (.eps), or Adobe Acrobat Portable Document Format (PDF). When creating electronic forms, save each project in its own folder and save the files with an Electronic Form (.elf) extension.

Print the Form: It is best to always print the form and proof it. Then if changes need to be made they can be made before placing the electronic form fields.

Draw Electronic Form Fields: Draw electronic form fields in each position that needs to display or collect data. These electronic form components control what is going to happen in that location of the form when it is displayed on a Windows-based PC. There are several types of actions that can take place on a PC screen so you will be describing that action through Field Types (i.e., Text, Numeric, Checkbox, etc.). Each field component also needs a Unique Field Name (i.e., UnitPrice, Quantity, etc.).

Use OneForm to Generate the Electronic Form: Use OneForm's [Generate E-Form] to generate an electronic version of the form. This will generate an HTML E-Form ready to be opened and tested within Windows Internet Explorer browser.

Test Usage of the Electronic Form: When generating the electronic form, you have the option to immediately test your results and run the form in Windows Internet Explorer. At this time you can simulate typical usage of the electronic form and test the fields. Also look over the form display and printing of the form if you are going to use the browser's print option. Once the form tests are complete, you are ready to determine how to distribute the form.

Distribute: When the form is ready a distributable project can be quickly created. HTML E-Forms use CGI scripts that are quickly generated by OneForm's E-Form Server Preferences. These files are kept within the form's project folder ready to be copied to an Internet or Intranet server.

HTML E-Forms

HTML e-forms are never downloaded onto the computer. They reside on a web server and are called upon to open whenever the e-form on the web site is accessed through the Intranet or Internet browser URL or link. When submitting data through an e-form, the forms are not directly linked to a database. The data is captured and submitted back to the server through CGI scripts.

E-forms are generated by OneForm as HTML JavaScript files. In order to display and fill the HTML e-form, the user must have a web browser with JavaScript enabled.

Also, OneForm Designer Plus creates Perl5 CGI script for the front and back-end data control of the form. The front-end controls a Search, List, and Recall of stored forms and data. This is necessary to allow people to recall earlier forms and information providing the basics of a workflow process. The back-end controls the behavior of the form when data is submitted and the saving of the data. It is necessary to format the data in such a way that it can be stuffed into a database.

JavaScript code handles the running of the E-Form. If any customization needs to take place, this can be accomplished through JavaScript programming.

Visual Basic 6.0 E-Forms

For legacy purposes, you can still use the Basic Field Properties and the Generate option to create Visual Basic 6.0 electronic forms.

This will require you set the OFDP Profile for *Visual Basic Forms.* This profile will open up menu options for Visual Basic Exe e-forms. See the *VB E-Forms* reference manual for information on designing and working with them.

This type of e-form is placed into an executable file that is downloaded onto each computer. When the form is needed, the user opens the EXE and the form displays for user input. The user can input data into the fillable fields and print the form. If the form is linked to a local database, the data can be saved and recalled through the form.

OneForm Designer Plus Program Options and Profile Settings

OFDP E-Form Field Properties has two sets of dialogs: a Basic and Advanced. It is recommended that you use the advanced field properties, since many features are not found in the Basic dialogs kept for legacy purposes for VB e-forms. By default you will have the menu "Use Advanced Elf Properties" toggled on, as shown in *Figure 1-1.*

The OFDP "Set OneForm Profile" designates menu settings according to the type of forms needed: E-Forms and Paper Forms are the most common. These profile selections are found under the Options, Program Options menu and defaults to E-Forms.

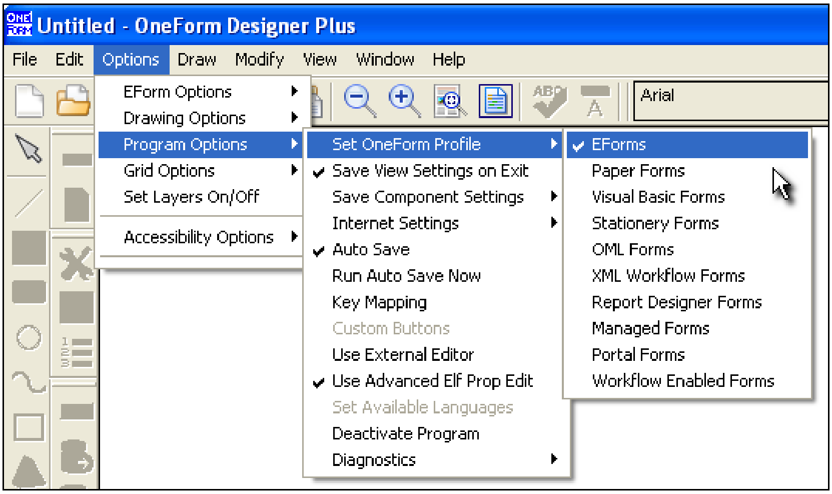


Figure 1-1: E-Forms is the Default Profile with Use Advanced Elf Properties

Note: If changing your profile, OneForm must be exited and restarted for the change to be applied.

Other Profiles

The profiles listed below are all described in their own reference manuals.

OML Forms: The free OneForm Manager Light Option uses this profile. This is a local computer e-form file management program.

Paper Forms: This is for designing artwork for manufactured printing of paper forms. When Paper Forms is selected, Typeset Output replaces the Print option and Output Specifications for Parts becomes available.

Managed Forms, Portal Forms, and *Workflow Enabled* are options that control the OFDP Forms Portal.

Stationery Forms: The OFDP Stationery Option uses this profile.

Visual Basic Forms: This profile will open up menu options for Visual Basic Exe e-forms. See the *VB E-Forms* reference manual for more information.

XML Workflow and *Report Designer* are custom features.

CHAPTER 2

Tutorial – Simple Fill-In E-Form

Paper business forms, especially internal forms, are prime candidates for replacement with electronic forms. Using Amgraf OneForm Designer Plus, paper forms can quickly be converted to E-Forms.

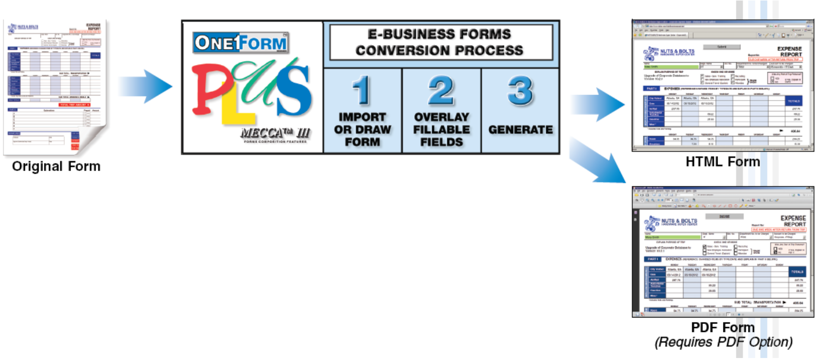


Figure 2-1: OneForm Designer Plus Conversion Process

Most Importantly, with OneForm Designer Plus (OFDP), you can freely distribute and use the E-Forms you create, because no filler software licenses are ever required.

Overview

In the following tutorial we will show you just how easy it is to make your business forms ready to use with Windows Client-Server PC's, or for posting on the Internet or internal Intranets.

Step 1 – Create a base form design.

There are three ways to get started with OneForm Designer's electronic forms, we have the tutorial designed to step you through each startup choice.

Step 1A: Draw a new form using the OneForm graphical tools, or

Step 1B: Import existing form design files from a desktop publishing system via the Encapsulated PostScript or PDF interpreters, or

Step 1C: Copy a form from the OneForm Forms Library, then edit the form by adding personalization and logos.

Step 2 – Insert fillable fields.

Fill-fields can quickly be overlayed on top of the form image using the OneForm autofield tool. Field types in this tutorial include textual, numeric, check box, drop-down list, date, and time. Fields can be further defined to control and validate input attributes such as character/line count, case, and numeric formatting.

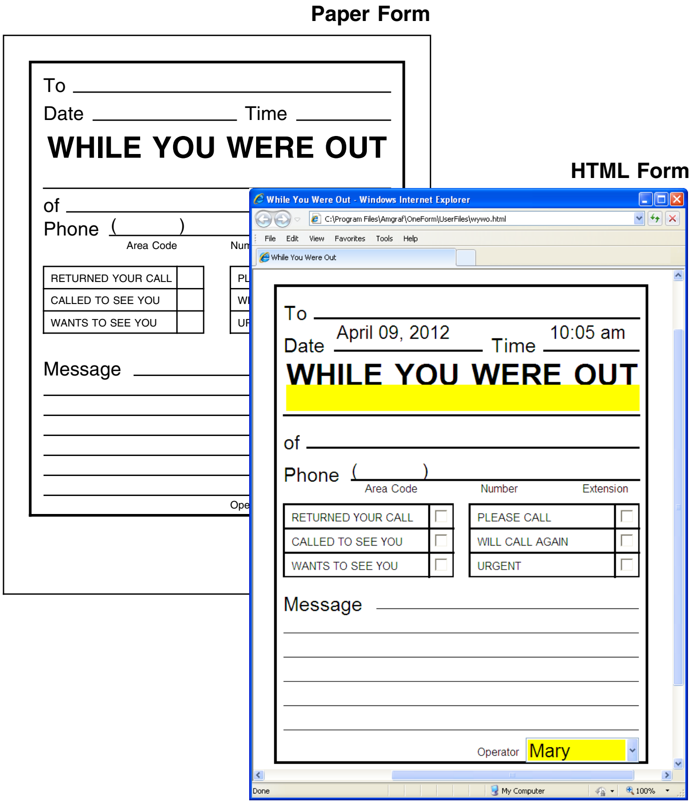
Step 3 – Finishing touches include editing the Form's Preferences for the browser's Title bar and setting tab ordering of each field.

Step 4 – Save, generate, and output the form.

After the form is fielded, it can be *generated* as an HTML JavaScript E-Form.

Sample File Used for the Tutorial

Our example will use the *While You Were Out* form. You can compose all of the rules, boxes, and text found on the form or load one of our examples supplied.


Paper Form
HTML Form
While You Were Out - Windows Internet Explorer
C:\Program Files\Amgraf\OneForm\UserFiles\wywo.html
File
Edit
View
Favorites
Tools
Help
While You Were Out
Done
My Computer
100%

To
Date April 09, 2012 Time 10:05 am

WHILE YOU WERE OUT

of
Phone ( )
Area Code Number Extension

RETURNED YOUR CALL	□	PLEASE CALL	□
CALLED TO SEE YOU	□	WILL CALL AGAIN	□
WANTS TO SEE YOU	□	URGENT	□

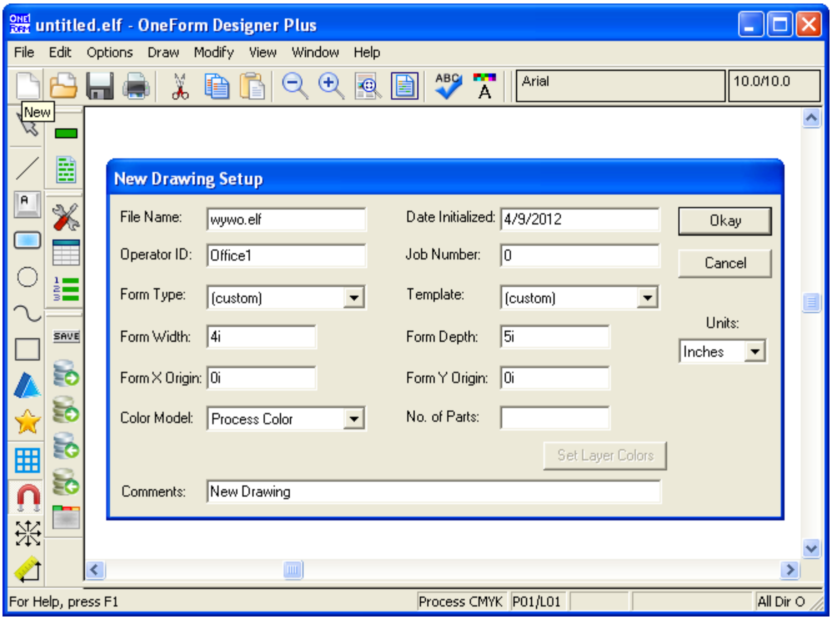
Message

Operator Mary

Figure 2-2: Paper Version and Electronic Version of the Form

Step 1A: Drawing a New Form

When you click the **New File** tool you are creating a brand new file and for electronic forms we need it to have an extension of **.elf**. Before selecting [Okay], you have an opportunity to type in a new form name and change the form size. We are going to draw a "While You Were Out" message form. Our new name is **wywo.elf** and the new size is **4" x 5".**



untitled.elf - OneForm Designer Plus

New Drawing Setup

File Name:	wywo.elf	Date Initialized:	4/9/2012
Operator ID:	Office1	Job Number:	0
Form Type:	(custom)	Template:	(custom)
Form Width:	4i	Form Depth:	5i
Form X Origin:	0i	Form Y Origin:	0i
Color Model:	Process Color	No. of Parts:	
Comments:	New Drawing		

Units: Inches

For Help, press F1 | Process CMYK | P01/L01 | All Dir O

Figure 2-3: Electronic Forms Must Use the File Extension .elf

As you are drawing, use the Save Tool to save your work into the wywo.elf file.

Note: If you find that the form field tools are not available, change the OneForm Software Profile under the Options Menu. The *EForms* option must be checked. If you make a change, re-open the software to activate the new profile.

Assistance in Drawing the Form

If you need assistance in drawing the form, the *Getting Started* reference manual has a step-by-step tutorial on drawing the *While You Were Out* form. It is *Chapter 4: Tutorial – Drawing a New Form.*

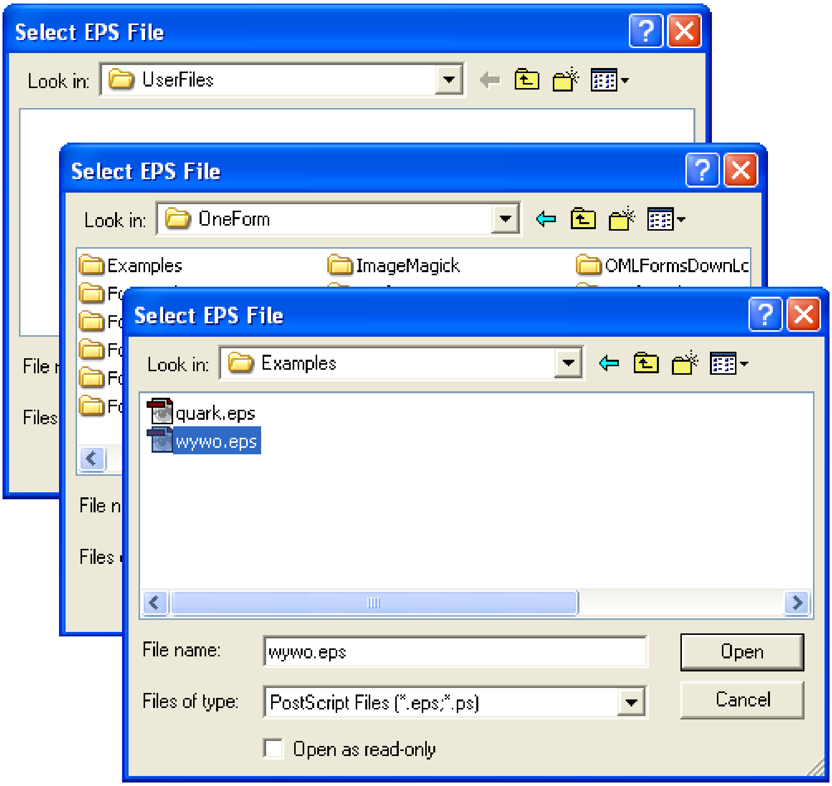
Once the artwork is complete, save, print, and proof the form. It is much easier to make changes before the fields have been placed on the form.

Step 1B: Importing EPS or PDF Files

You can import a file from another forms design system simply by using **File's Import,** menu. The choices are to import either an EPS file or PDF file.

Sample files for importing can be found under OneForm's **Examples** directory and are named **wywo.eps** or **wywo.pdf.**

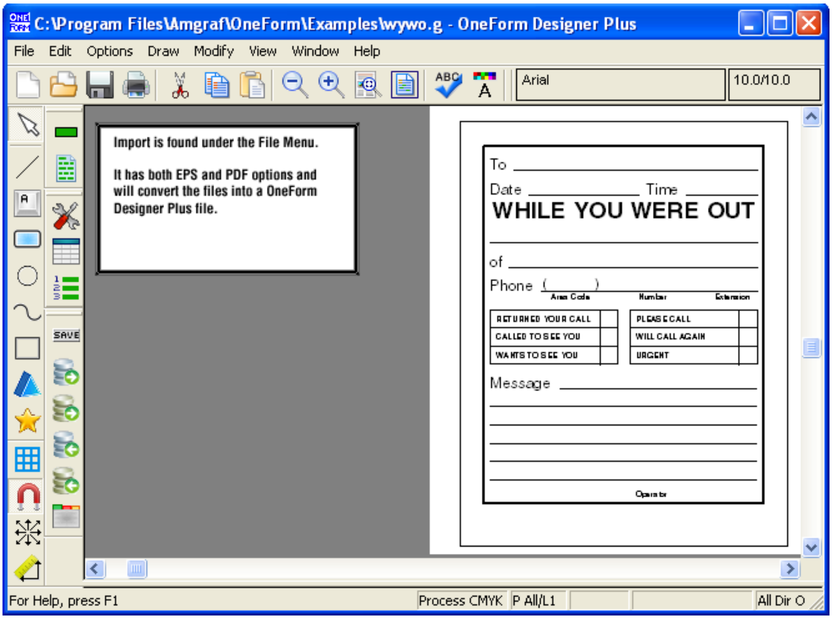
Import takes you to a UserFiles folder. Click onto the Close Files icon and you will see the Examples folder.



File name: wywo.eps

Files of type: PostScript Files (*.eps;*.ps)

Figure 2-4: Sample Files are Found in Examples Folder


C:\Program Files\Amgraf\OneForm\Examples\wywo.g - OneForm Designer Plus
File
Edit
Options
Draw
Modify
View
Window
Help
ABC
A
Arial
10.0/10.0
SAVE

Import is found under the File Menu.

It has both EPS and PDF options and will convert the files into a OneForm Designer Plus file.

To ____________

Date ____________ Time ____________

WHILE YOU WERE OUT

of ____________

Phone ( ) ____________

Area Code Number Extension

RETURNED YOUR CALL		PLEASE CALL	
CALLED TO SEE YOU		WILL CALL AGAIN	
WANTS TO SEE YOU		URGENT	

Message ____________

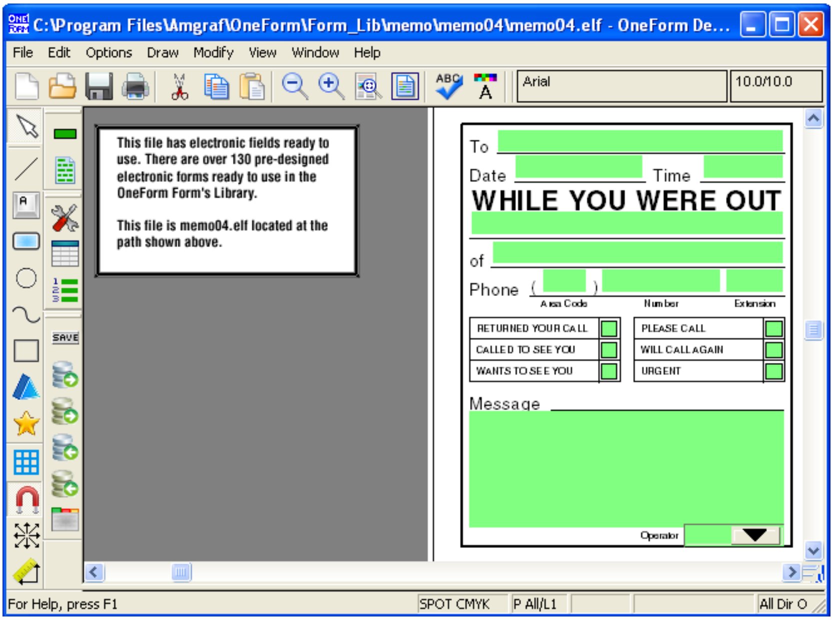
Operator

Figure 2-5: Looking at an Imported EPS File

The Import EPS option converts to a standard OneForm Designer Plus graphic file. Since we need it to have an extension of **.elf** for electronic forms, simply use **Save As** and rename it.

Step 1C: Using OneForm's Library

OneForm Designer has a large selection of pre-designed electronic fielded business forms ready to use and customize. You can start with any form in the Form_Lib directory simply by opening it.



C:\Program Files\Amgraf\OneForm\Form_Lib\memo\memo04\memo04.elf - OneForm De...

This file has electronic fields ready to use. There are over 130 pre-designed electronic forms ready to use in the OneForm Form's Library.

This file is memo04.elf located at the path shown above.

To

Date Time

WHILE YOU WERE OUT

of

Phone ( )

Area Code Number Extension

RETURNED YOUR CALL	🟩	PLEASE CALL	🟩
CALLED TO SEE YOU	🟩	WILL CALL AGAIN	🟩
WANTS TO SEE YOU	🟩	URGENT	🟩

Message

Operator

Figure 2-6: Opening a File

Clicking on the **Open File** tool will start you out at the *UserFiles* directory. We need to back out of there and go to the **Form_Lib** folder to find the **memo/memo04** directory. Open the **memo04.elf** file.

This form is complete with electronic form fill fields, ready to be generated into an electronic version.

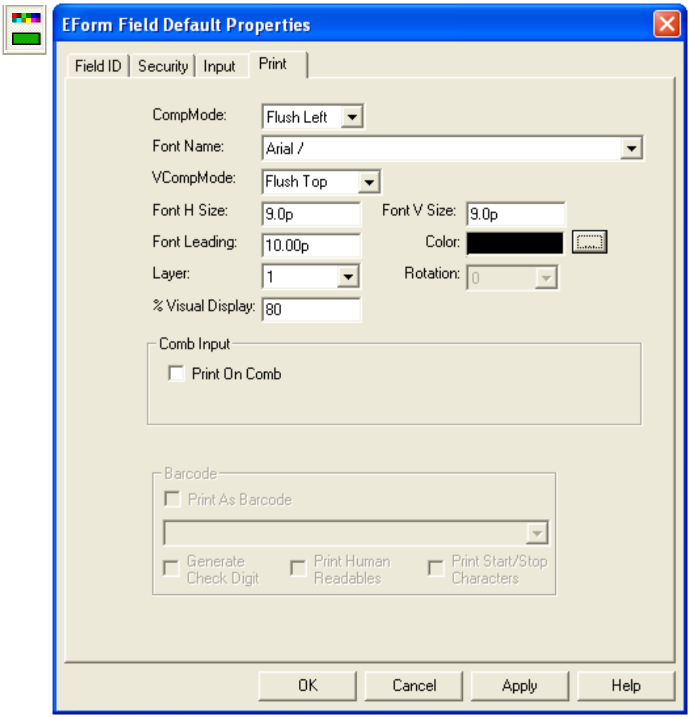
Go directly to Step 4 on page 24.

Step 2: Add Electronic Fill Fields

If you have drawn or imported your form, overlay the form with electronic fields for inputting data on the finished form.

Set Field Properties

Begin by setting up the field properties for the fields we are going to draw. This is done through the **Set Field Properties** Tool. Our fields will be TEXT field types to start. Lets make a change to the Print Tab, it controls how the user's input will look.



EForm Field Default Properties

CompMode: Flush Left

Font Name: Arial /

VCompMode: Flush Top

Font H Size: 9.0p

Font V Size: 9.0p

Font Leading: 10.00p

Color: ■

Layer: 1

Rotation: 0

% Visual Display: 80

Comb Input

□ Print On Comb

Barcode

□ Print As Barcode

□ Generate Check Digit

□ Print Human Readables

□ Print Start/Stop Characters

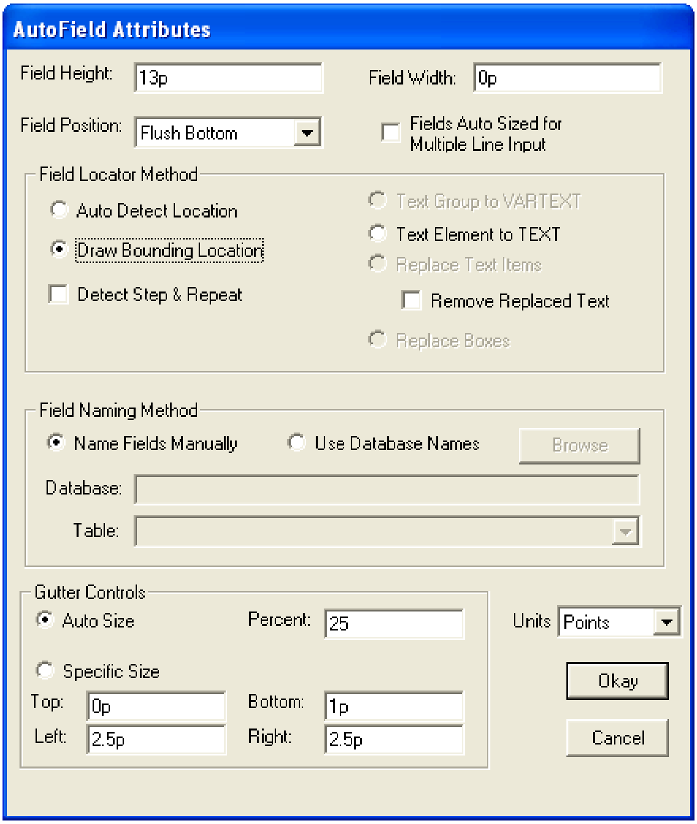
Figure 2-7: The Field Properties Tool Sets Up Print Attributes

Set the size of the text to 9p (9-point) in both places as shown. [Okay] your settings. This size will fit nicely on our form. Now, as we draw our fields, they will all be TEXT fields with 9-point user input.

Note: This exercise uses the *(Advanced) E-Form Field Properties* dialogs. If yours look different, turn on the **Advanced E-Form Field Properties.** This is done through the Options, Program Options, Use Advanced Elf Properties.

Using the Auto Field Tool

Overlay the form with electronic fields. These fields will collect data in the finished product. Lets begin placing fields on our form by choosing the **[Auto Field]** tool on the Tool bar. An Auto Field Attributes dialog will appear. Input the **Field Height** as 10-points. Select the Field Locator Method, **Draw Bounding Location** and **[Okay].**



Field Height: 13p

Field Width: 0p

Field Position: Flush Bottom

□ Fields Auto Sized for Multiple Line Input

○ Auto Detect Location

● Draw Bounding Location

□ Detect Step & Repeat

○ Text Group to VARTEXT

○ Text Element to TEXT

○ Replace Text Items

□ Remove Replaced Text

○ Replace Boxes

● Name Fields Manually

○ Use Database Names

Database:

Table:

● Auto Size

Percent: 25

○ Specific Size

Top: 0p

Bottom: 1p

Left: 2.5p

Right: 2.5p

Units Points

Figure 2-8: The Auto Field Tool Has an Attributes Dialog

Now when using the location method of Draw Bounding Location, you will **pick the two endpoints of each line** in which you want a field to be placed above. Since the attributes default is Flush Bottom and Gutters are 25% of the height, a perfect sized field will appear above each rule as you pick the two endpoints of each line.

Once the second pick has been made, the Field Properties will appear. **[Okay]** the field properties. The Field options will name each field consecutively *Field01, Field02, etc.* The Field Type will always be TEXT. This is fine for starters and allows us to focus only on placing the fields. Continue placing fields for the Date, Time, Name, Company, Phone, and Operator using the Auto Field tool.

For the CheckBoxes, we need to *reset the Auto Field tool.* To get back to the Auto Field Attributes Dialog, **toggle the tool off and back on again.** Here we need a Field Position of **Centered** and the Field Locator Method to be **Auto Detect Location.** Again, **[Okay]** the setups.

Auto Detect Location uses a dynamic rectangle to provide two corners of a region in which it looks for retangular box shapes.

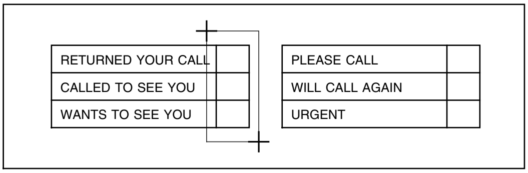
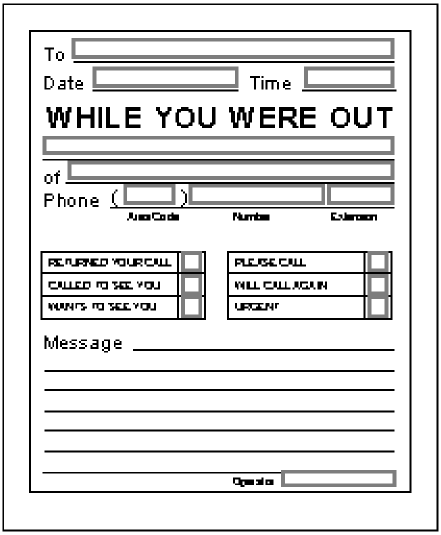


Figure 2-9: Pick Just Outside the First CheckBox Column

When picking the location for the first column of checkboxes, **pick outside the first checkbox area, then below the third checkbox.** As it finds each checkbox location, the Field Attributes Dialog will appear. **[Okay]** each one. Go over and **pick two locations** for checkboxes four through six and also **[Okay]** those.



To

Date Time

WHILE YOU WERE OUT

of

Phone ( )

Area Code Number Extension

RETURNED YOUR CALL	☐	PLEASE CALL	☐
CALLED TO SEE YOU	☐	WILL CALL AGAIN	☐
WANTS TO SEE YOU	☐	URGENT	☐

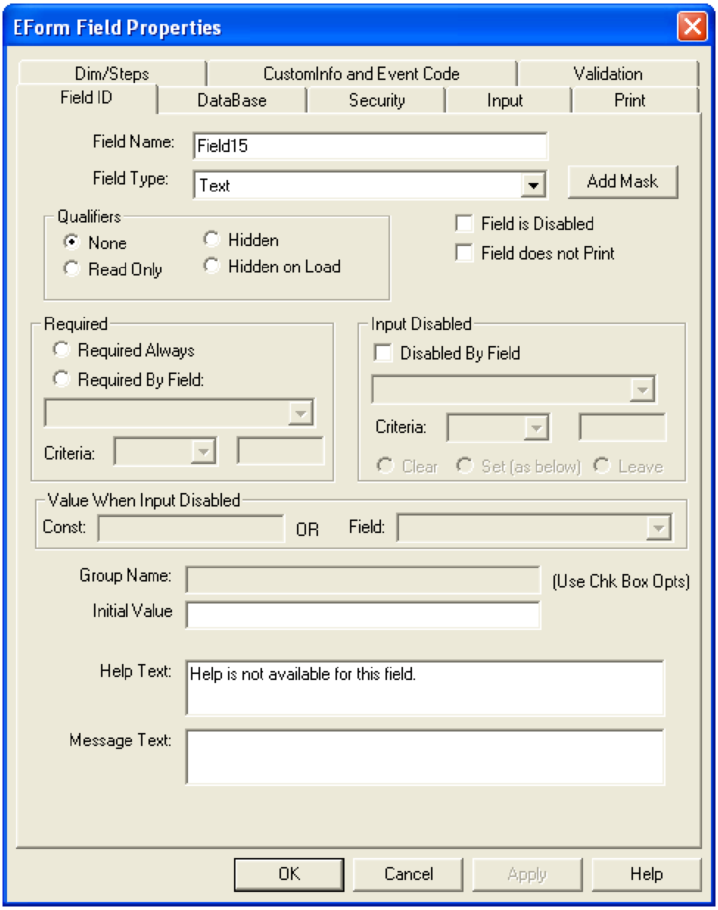
Message

Operator

Figure 2-10: Place All Fields with Auto Field Except the Message

Using the Add Field Tool

When you get to the *Message* field, it is best to use the **Add Field** tool. This tool places the edges of the field exactly on the **two diagonal picks.** It will then take you into the menu which allows you to assign a Field Name and Field Type. Just select **[Okay]** to accept the default.



EForm Field Properties

Dim/Steps | CustomInfo and Event Code | Validation
Field ID | DataBase | Security | Input | Print

Field Name: Field15

Field Type: Text

Qualifiers: ● None ○ Hidden ○ Read Only ○ Hidden on Load

□ Field is Disabled

□ Field does not Print

Required: ○ Required Always ○ Required By Field:

Criteria:

Input Disabled: □ Disabled By Field

Criteria:

○ Clear ○ Set (as below) ○ Leave

Value When Input Disabled: Const: OR Field:

Group Name: (Use Chk Box Opts)

Initial Value

Help Text: Help is not available for this field.

Message Text:

Figure 2-11: Adding the Message Field

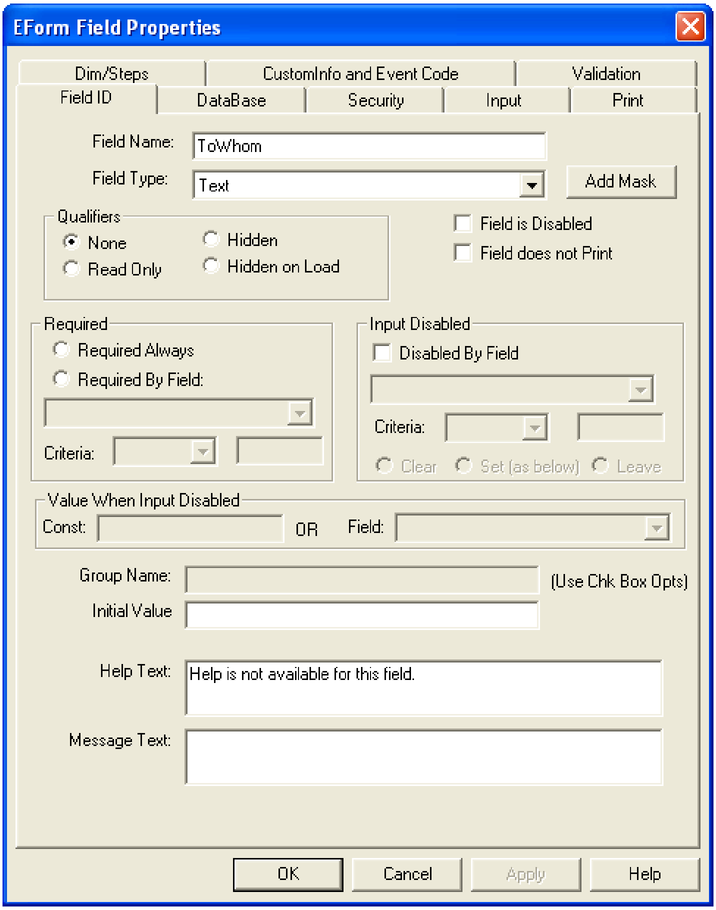
You can assign the unique Field Name, Field Type, and properties all within the properties dialog at the time of creation. But to make it easier to learn, each of these items have been broken down into individual Steps.

Deleting Fields

If you happen to draw a field you don't want, immediately use the Undo Tool. Or, use the Select Cursor to identify the field and press the [Delete] key.

Modifying Field Properties

The Field Properties allows you to specify the filling properties of the electronic fields. The Field Properties appear each time a field is drawn. To later access the properties, pick the field with the *Select* tool and right-click to select the *E-Form Field Properties* option.



Field Name: ToWhom

Field Type: Text

● None

Help Text: Help is not available for this field.

Figure 2-12: Modifying Field Properties

Every field must have its own unique name. The Field Name can stay *Field01, Field02, etc.,* but we suggest you give it an English name for easier identification. You may do this now if you want by using the Select tool to identify the field and going to the pop-up menu's **E-Form Field Properties.** For example, *Field01* could be named *ToWhom.* There are some reserved words. If you happen to use one of them, a message will appear.

Assigning the Field Type

The *Field Type* indicates the purpose of the field. There are many different types, each one doing a particular action. For example, Text allows the operator to simply cursor to the field and type (which is why it is the default). CheckBox wants a click from the mouse and places an X in the box.

The WYWO form uses Text, Date, Time (VB Only), CheckBox, and ComboBox Field Types. Changing a field type is as easy as **identifying the field** to change with the Select tool and opening its Field Properties and **selecting a type from the list.** The drawing below identifies the type of each field. Go ahead and change the field types which are not TEXT to their correct type. HTML does not recognize Time fields, so keep that one a Text field and we will apply a data Input Mask for the Time in our HTML E-Form.

Most of our fields are TEXT field types.

Telephone Numbers are considered TEXT fields.

NUMERIC field types are used only when calculations are necessary.

TIME field types are recognized only by VB E-Forms.

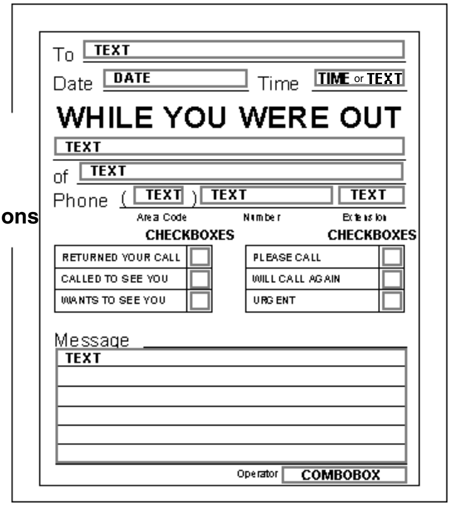


Figure 2-13: "While You Were Out" Field Types

Field ID Tab Options for Required or Read-Only

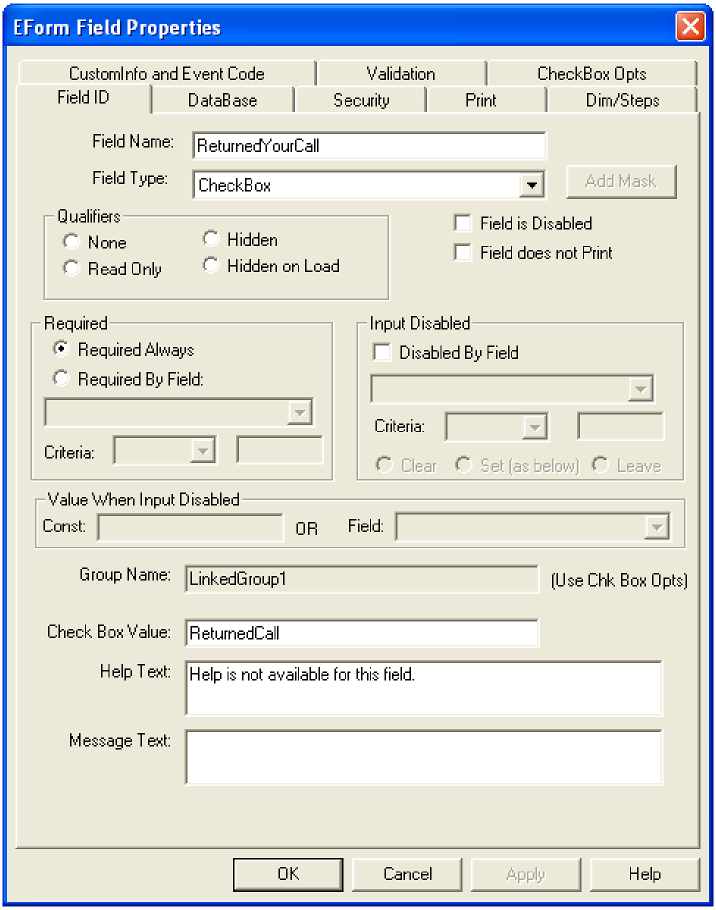
The following optional actions allow us to add intelligence to the form. Simple controls such as REQUIRED information or READ-ONLY make the form easier to use. These properties can be added under the *Field ID Tab* of the properties dialog.

- The *ToWhom, WhoCalled,* and *Operator* are all REQUIRED to be input before the form can be saved. Go to each one of these fields and under the **Field ID Tab** click on **Required.**
- In the case of the *DateReceived* and *TimeReceived* fields the information is being brought in from the computer’s date To get the tab to bypass these fields and not allow operator input, click on **Read Only.**

CheckBox Options

Only one entry should be selected for the three CheckBoxes *ReturnedYourCall, CalledToSeeYou,* and *WantToSeeYou*. To make them exclusive when chosen, apply a GROUP NAME and CHECKBOX VALUE.

The dialog below shows the CheckBox has a Group Name of *LinkedGroup1*.



Field Name: ReturnedYourCall

Field Type: CheckBox

Qualifiers
○ None ○ Hidden
○ Read Only ○ Hidden on Load

□ Field is Disabled
□ Field does not Print

Required
● Required Always
○ Required By Field:
Criteria:

Input Disabled
□ Disabled By Field
Criteria:
○ Clear ○ Set (as below) ○ Leave

Value When Input Disabled
Const: OR Field:

Group Name: LinkedGroup1 (Use Chk Box Opts)

Check Box Value: ReturnedCall

Help Text: Help is not available for this field.

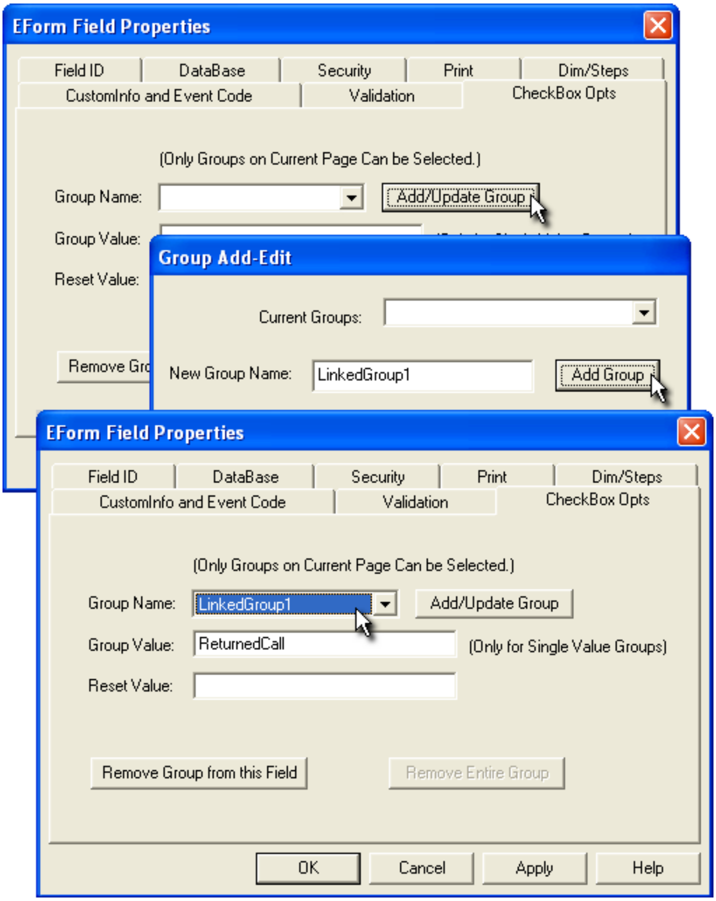
Message Text:

Figure 2-14: CheckBoxes Can Have a Group Name

Creating and associating a Group is found under the **CheckBox Option** Tab.

CheckBox Options Tab

First, pick **[Add/Update Group],** type in **LinkedGroup1** as your group name assignment, then **[Add Group], [OK].**



New Group Name: LinkedGroup1

Group Name: LinkedGroup1

Group Value: ReturnedCall

Figure 2-15: Adding a Group Name to CheckBoxes

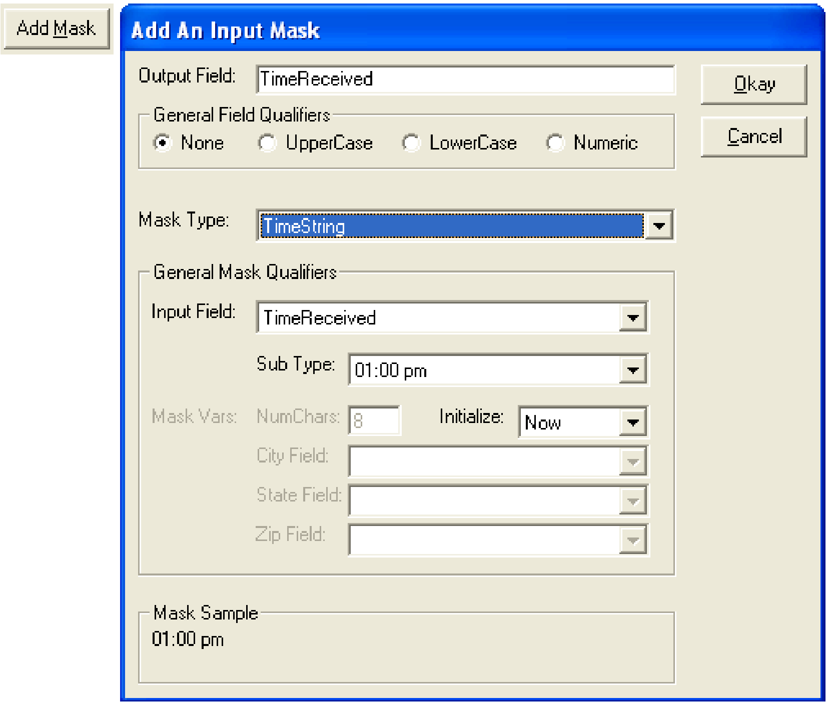
Now it is available for any CheckBox. For each of the three CheckBoxes, select **LinkedGroup1** from the list. This group name assignment associates multiple CheckBoxes together and then allows only one of that group to be selected.

Next assign a different value to each CheckBox. Ours is "ReturnedCall" for this CheckBox. You can make the value the same as the field name.

When the form is submitted to its data table, the Group Name will be the data table column name, and the value will be the data in the column.

Adding an Input Mask to a Text Field

For HTML E-Forms, displaying a time value is accomplished through an input mask of a text field. Go to the Field ID Tab of the *TimeReceived* field and click on the **[Add Mask]** button.



Output Field: TimeReceived

General Field Qualifiers: ● None ○ UpperCase ○ LowerCase ○ Numeric

Mask Type: TimeString

General Mask Qualifiers

Input Field: TimeReceived

Sub Type: 01:00 pm

Mask Vars: NumChars: 8 Initialize: Now

City Field:

State Field:

Zip Field:

Mask Sample

01:00 pm

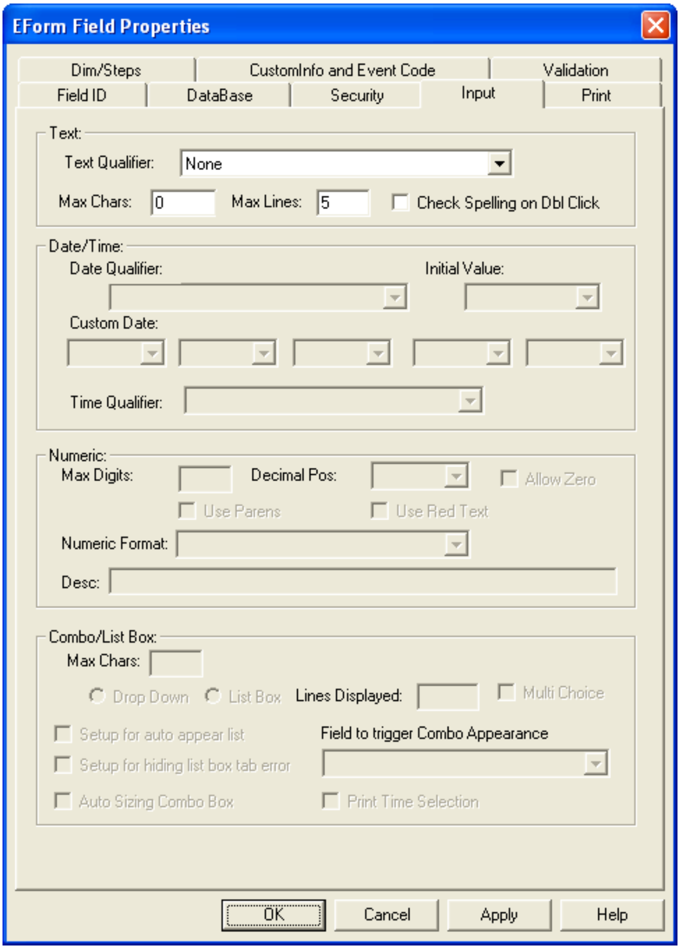
Figure 2-16: The Add Mask Dialog

From the Mask Type list select **TimeString** and choose a sub-type for the display (i.e, 01:00 pm) and the initialize type (i.e., Now). This will cause the computer's time to display in the text field.

The Input Properties Tab

This tab places controls on the data as it is input into a field, such as it will all be uppercase letters. Go to the following fields and make these changes to the Input Tab.

- The style of the *DateReceived* field is controlled here. Go to the **Input Tab** and select a **Date Qualifier.** Click on it and a list of choices will appear.



Text:

Text Qualifier: None

Max Chars: 0 Max Lines: 5 □ Check Spelling on Dbl Click

Date/Time:

Date Qualifier: Initial Value:

Custom Date:

Time Qualifier:

Numeric:

Max Digits: Decimal Pos: □ Allow Zero

□ Use Parens □ Use Red Text

Numeric Format:

Desc:

Combo/List Box:

Max Chars:

○ Drop Down ○ List Box Lines Displayed: □ Multi Choice

□ Setup for auto appear list Field to trigger Combo Appearance

□ Setup for hiding list box tab error

□ Auto Sizing Combo Box □ Print Time Selection

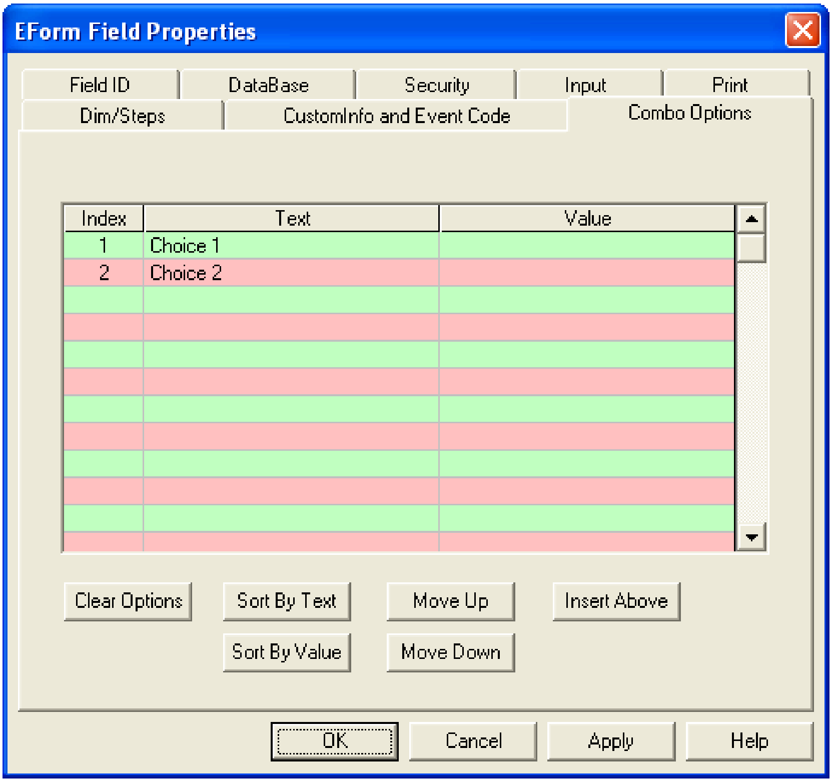
Figure 2-17: The Input Tab

- We used a maximum of five lines for the *Message* field on our WYWO form. You can type as much as you want with Max Chars of 0, but only 5 lines will print. Your electronic form can hold any amount of data, but when it comes to printing the form there is only so much physical room to print. Select the **Input Tab** and for **Max Lines** type in **5.**

- The *AreaCode* field is considered a text field with input of numbers only. Select the **Input Tab** and choose **Numeric Only.**

The Combo Options Tab

Here we can customize the ComboBox choices for the **Operator's** field on the form. It will always start out with *Choice 1* and *Choice 2*. Just click in the "Text" column and type in the name. Do this for each name needed in the list.



Index	Text	Value
1	Choice 1	
2	Choice 2	

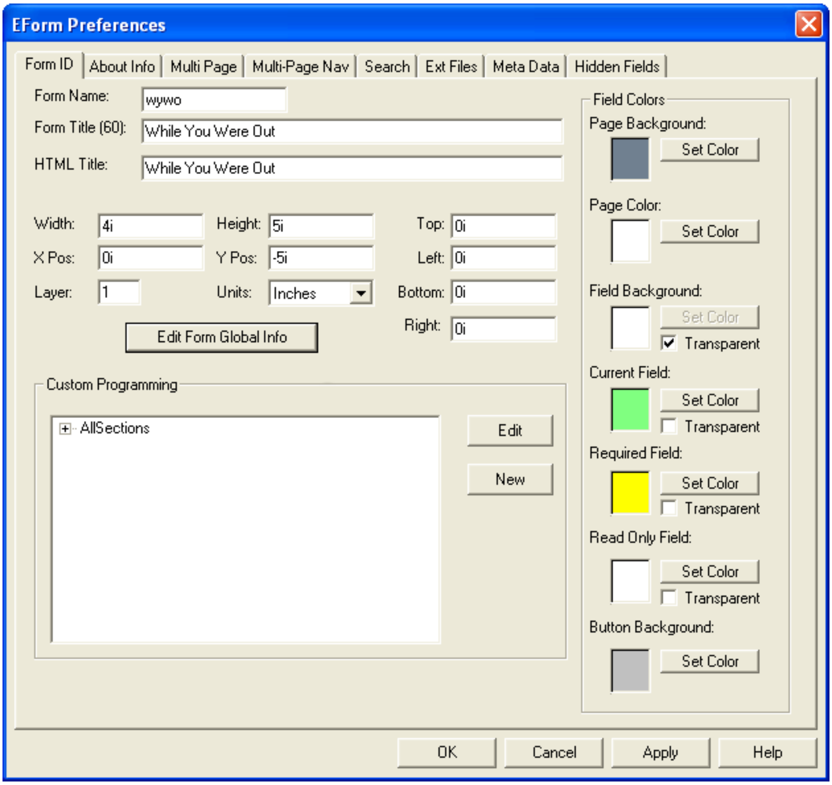
Figure 2-18: Input the ComboBox Choices

The Combo Option Tab will only be available if the Field Type is a ComboBox.

Step 3: Setting Form Preferences and Tab Ordering

Every form has its own set of preferences and are found under the Options, EForm Options, EForm Preferences menu. The following dialog assists in setting these preferences.

The Form ID Tab



Form Name: wywo
Form Title (60): While You Were Out
HTML Title: While You Were Out

Width:	4i	Height:	5i	Top:	0i
X Pos:	0i	Y Pos:	-5i	Left:	0i
Layer:	1	Units:	Inches	Bottom:	0i
				Right:	0i

Custom Programming: ⊞ AllSections

Field Colors:
- Page Background: ▨
- Page Color: □
- Field Background: □ ☑ Transparent
- Current Field: 🟩 ☐ Transparent
- Required Field: 🟨 ☐ Transparent
- Read Only Field: □ ☐ Transparent
- Button Background: ▨

Figure 2-19: The Form ID Tab

This tab starts out with fields for the Form's Project Name and the Title for the Helper Pages and the Browser Title Bar.

The Project Name is the basename of the ELF file. When the HTML is generated it will use *projectname.html.*

The first Title field is for the administration helper pages that are automatically created when made server ready.

The second Title field is for the name of the form within the HTML e-form's browser title bar.

Setting Tab Order

Tabbing order is the route the cursor takes when tabbing from one field to another. To set the Tab Order, go to the **Options Menu, EForm Options, Set Tab Order.**

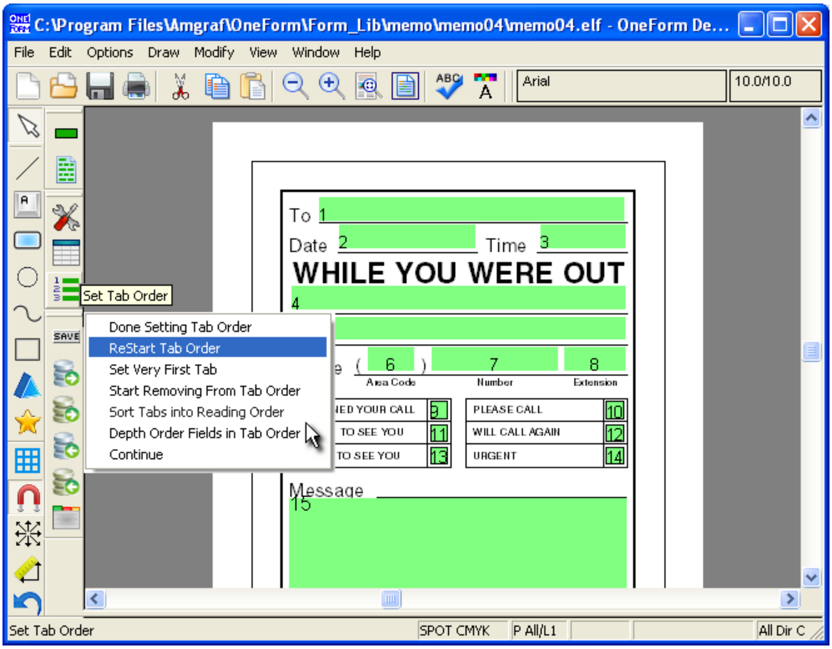


Figure 2-20: Setting Tab Order

Numbers will appear representing the current Tab Sequence. You can reset the tab order by right-clicking and choosing **ReStart Tab Order**. Choose the first field in the new Tab Sequence. This is where the cursor will begin when the form is opened. Continue picking on the fields in the order you want them to tab until all fields have been included. When you have finished, choose **Done Setting Tab Order**.

Read-only fields and CheckBoxes will show 0 in the tab sequence.

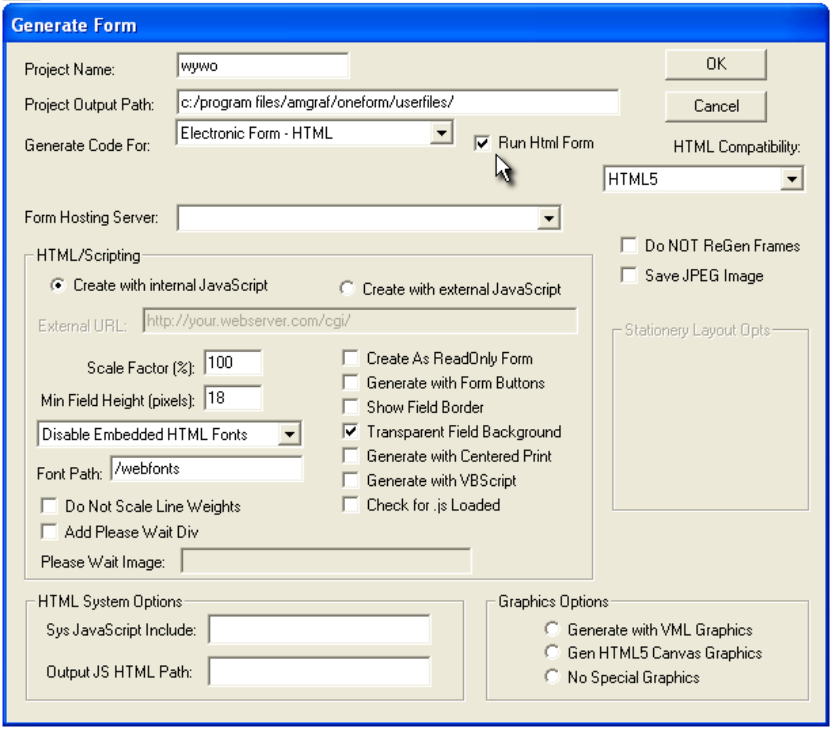
Note: CheckBoxes can be made tabable by going to E-Form Preferences, Form ID Tab, [Form Global Info], and turn on "Tabable CheckBoxes". This will allow resetting tab order to include CheckBoxes.

Step 4: The Generate Option

OneForm's Generate option allows you to take the same form file and generate it as an HTML E-Form.

Electronic Form – HTML

Click on the blue **Generate EForm** tool on the top tool bar. For the Generate Code, select within the drop-down list **Electronic Form - HTML.** Also click on the checkbox for **Run HTML Form** and select **[Okay].**



Generate Form

Project Name: wywo

Project Output Path: c:/program files/amgraf/oneform/userfiles/

Generate Code For: Electronic Form - HTML

☑ Run Html Form

HTML Compatibility: HTML5

Form Hosting Server:

□ Do NOT ReGen Frames

□ Save JPEG Image

HTML/Scripting

● Create with internal JavaScript ○ Create with external JavaScript

External URL: http://your.webserver.com/cgi/

Scale Factor (%): 100

Min Field Height (pixels): 18

Disable Embedded HTML Fonts

Font Path: /webfonts

□ Do Not Scale Line Weights

□ Add Please Wait Div

Please Wait Image:

□ Create As ReadOnly Form

□ Generate with Form Buttons

□ Show Field Border

☑ Transparent Field Background

□ Generate with Centered Print

□ Generate with VBScript

□ Check for .js Loaded

Stationery Layout Opts

HTML System Options

Sys JavaScript Include:

Output JS HTML Path:

Graphics Options

○ Generate with VML Graphics

○ Gen HTML5 Canvas Graphics

○ No Special Graphics

Figure 2-21: Generate E-Form

This creates an HTML file in the base directory. The "Run E-Form" option will start Windows Internet Explorer (or your default browser) and display the form ready for testing.

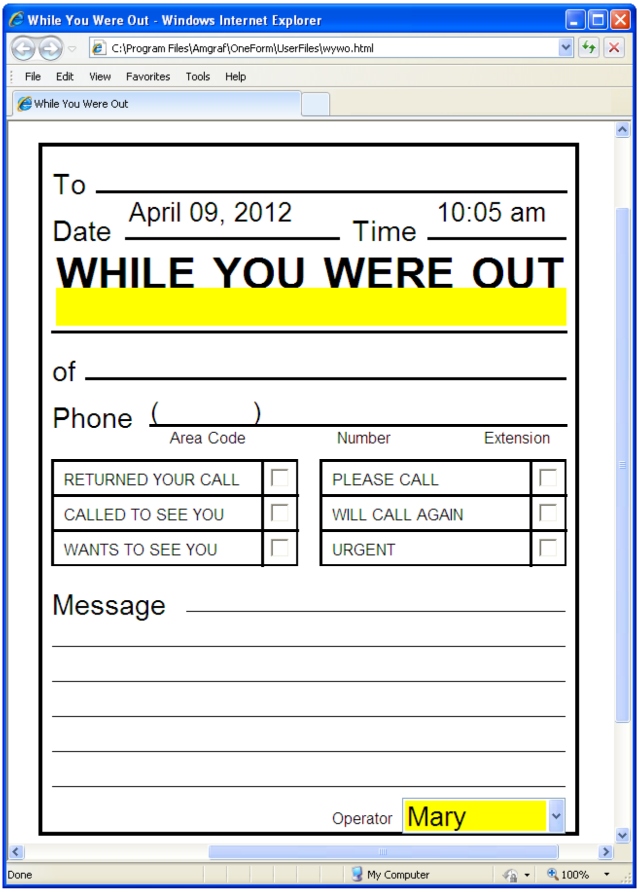
Note: The Generate EForm option is also found under the Options, EForm Options, **Generate EForm** menu and has a shortcut key **[Ctrl]-[G].**

The HTML E-Form will be displayed, ready to fill-in. Congratulations, you have created an electronic form.

Field Color Identifiers

The fields will display in one of three colors. Each color represents a property assigned to the field:

Green - Identifies where the cursor is located.
Yellow - Identifies that the field is required to be filled-in.
White - Read-only and optional fields.


While You Were Out - Windows Internet Explorer
C:\Program Files\Amgraf\OneForm\UserFiles\wywo.html
File
Edit
View
Favorites
Tools
Help
While You Were Out
Done
My Computer
100%

To

Date April 09, 2012 Time 10:05 am

WHILE YOU WERE OUT

of

Phone ( )
Area Code Number Extension

RETURNED YOUR CALL	□	PLEASE CALL	□
CALLED TO SEE YOU	□	WILL CALL AGAIN	□
WANTS TO SEE YOU	□	URGENT	□

Message

Operator Mary

Figure 2-22: The Electronic Form

Tab through the form and test field input. Closing your browser will take you back to OFDP.

CHAPTER 3

Drawing and Modifying Electronic Fields

There are several tools and menus which allow you to draw electronic fillable fields on your form. A fillable field is where the user can make selections or input data while using the form. In this chapter we will describe the best techniques for drawing these fields. They appear as a green rectangular area on the form while in OneForm. As soon as a field is defined, the E-Form Field Properties dialog will appear ready for you to give it a unique name. All of our examples will be working with the default names and TEXT field type.

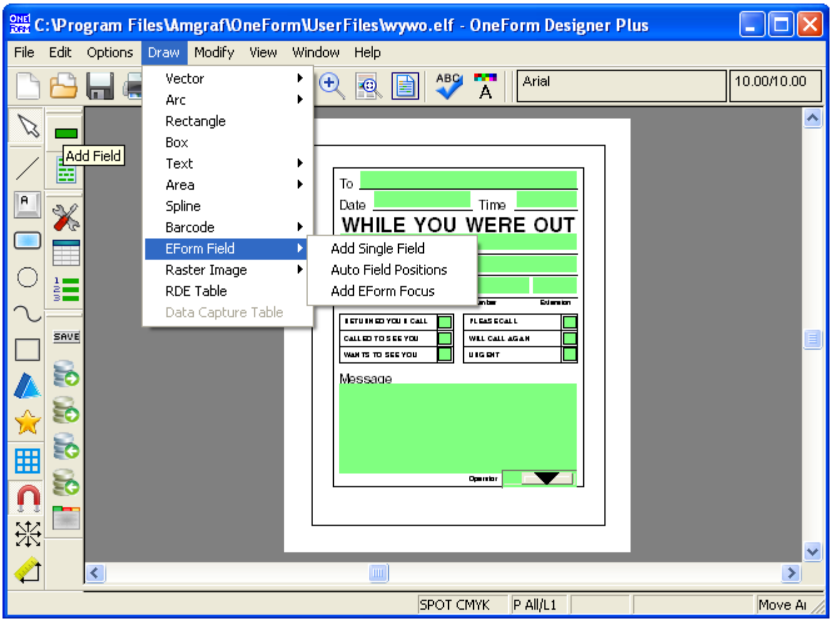


Figure 3-1: Field Menus and Tools

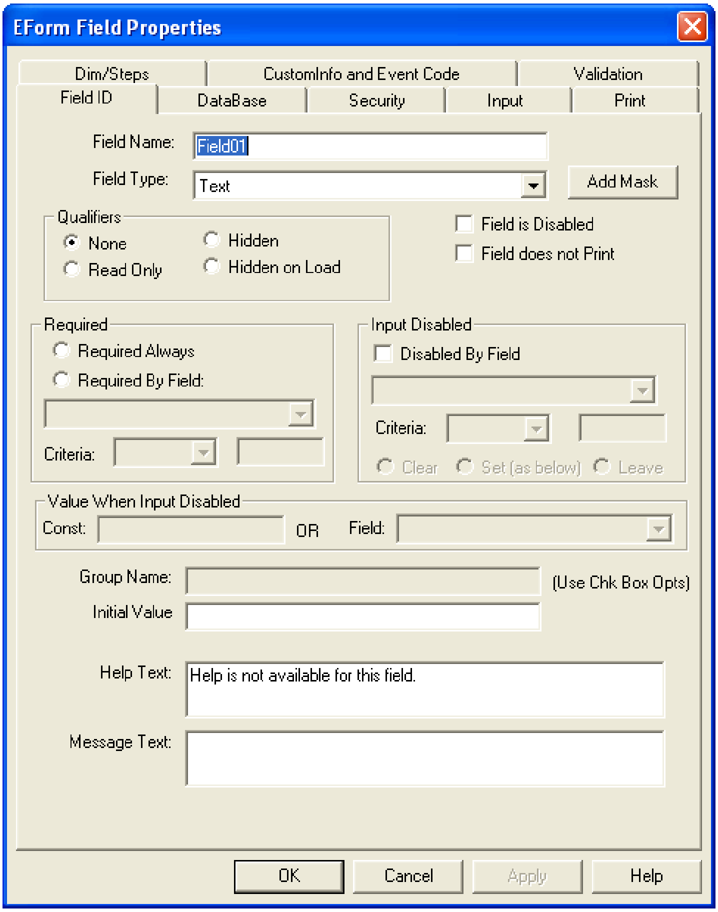
There are three Electronic Field Tools available. The first two, **Add Single Field** and **Auto Field Positions** allow you to add new fields to your form. The third tool is the **Set Electronic Field Properties.** This tool takes you into the setup dialogs and allows you to assign default properties of the fields before drawing. The drawing tools are also found in the Draw Menu.

Note: The first time you pick one of the E-Form Field tools, a prompt may appear asking you to approve a Form Name. Click [Yes] and [OK] the Form Name within the Form Preferences. It is used when generating the electronic form.

Add Single Field Tool

The Add Field tool is for placing a field exactly where you pick. There will be no pre-determined gutters, field dimensions or locations. It is placed exactly as you draw it.

To add a new field, choose the **Add Field** tool. Select a position for the **upper-left corner of your field.** As you move your cursor, you will see a dynamic rectangle representing the field size. When you **pick your second diagonal corner,** the Field Properties dialog will appear. **[Okay]** the properties and an electronic form field will be created within the two diagonal picks. Every time you draw a field, its E-Form Field Properties will open.



Field Name: Field01

Field Type: Text

Qualifiers
● None ○ Hidden
○ Read Only ○ Hidden on Load

□ Field is Disabled
□ Field does not Print

Required
○ Required Always
○ Required By Field:
Criteria:

Input Disabled
□ Disabled By Field
Criteria:
○ Clear ○ Set (as below) ○ Leave

Value When Input Disabled
Const: OR Field:

Group Name: (Use Chk Box Opts)

Initial Value

Help Text: Help is not available for this field.

Message Text:

Figure 3-2: Advanced E-Form Field Properties

Auto Field Tool

Another way to place fields is with the Auto Field tool. It has a dialog for defining the height, width, position and gutter of the field. It also has a Field Locator Method which uses the base artwork to position the field. As each field location is found, the Field Properties dialog will appear. [Okay] the properties and an electronic form field will be created.

To draw a field, click on the **Auto Field** tool. The auto field dialog box will appear. Notice that there are defaults for the field height, width, position, gutters and location method. After clicking **[Okay],** you will then be able to draw a field. The *Draw Bounding Location Method* is similar to Add Field, **just make two diagonal picks.** The Field Properties dialog will appear. **[Okay]** the properties and an electronic form field will be created. Notice that the size of the field is not exactly where you picked, but the height of the Auto Field Attributes.

The *Auto Detect Location Method* uses the dynamic rectangle to provide two corners of a region within which it looks for rectangular box shapes. The fields will be placed within the box shape according to the Auto Field properties.

Auto Field Dialog

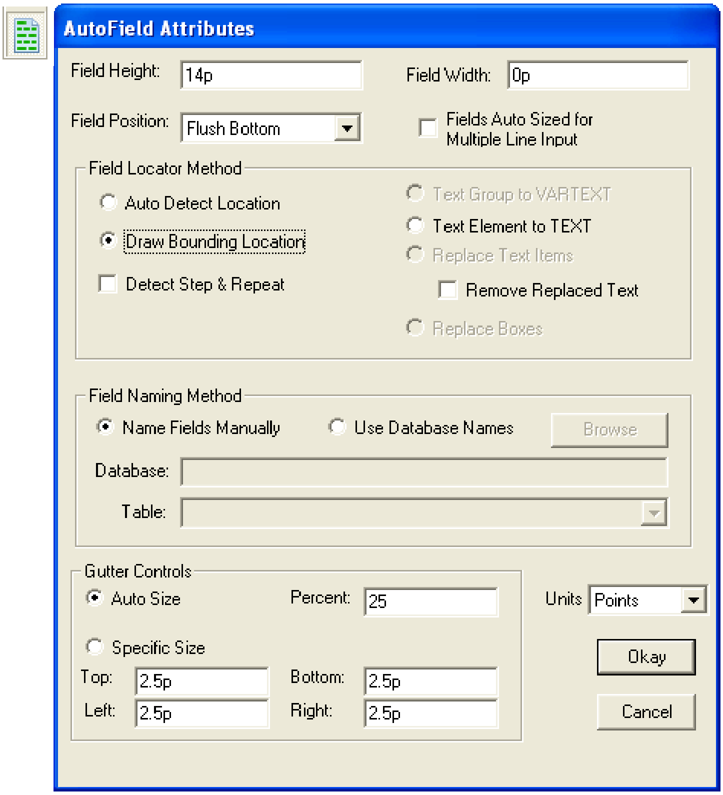
The Auto Field dialog box controls how the Auto Field tool will behave during the drawing process. When you click on the Auto Field tool, its dialog box will appear, giving you an opportunity to define location properties.

Field Height: This is the height of each field drawn. This will enable you to have consistent looking fields throughout your form. It will be ignored when *Field Auto Sized for Multiple Line Input, Replace Text Items,* or *Replace Boxes* are checked.

Field Width: This will lock in the width of the field only when using *Replace Text Item.*

Field Position: This controls the vertical position of the field within the retangular shape you are making. Your choices are *Flush Top*, *Centered*, and *Flush Bottom*. This option is ignored if *Field Auto Sized for Multiple Line Input* is checked.

Fields Auto Sized for Multiple Line Input: This is for making a text field that will allow more than one line of input. Fields Auto Sized for Multiple Line Input ignores the Field Height, Field Width and Field Position options. In short, the field size is determined by your two picks. The software then looks at the field height and determines how many lines of text can be input into the field based on the size of the font and leading. The maximum number of text lines that will fit into the field will be filled-in on the Input tab of Field properties.



AutoField Attributes

Field Height: 14p

Field Width: 0p

Field Position: Flush Bottom

□ Fields Auto Sized for Multiple Line Input

Field Locator Method

○ Auto Detect Location

● Draw Bounding Location

□ Detect Step & Repeat

○ Text Group to VARTEXT

○ Text Element to TEXT

○ Replace Text Items

□ Remove Replaced Text

○ Replace Boxes

Field Naming Method

● Name Fields Manually

○ Use Database Names

Database:

Table:

Gutter Controls

● Auto Size

Percent: 25

○ Specific Size

Top: 2.5p

Bottom: 2.5p

Left: 2.5p

Right: 2.5p

Units: Points

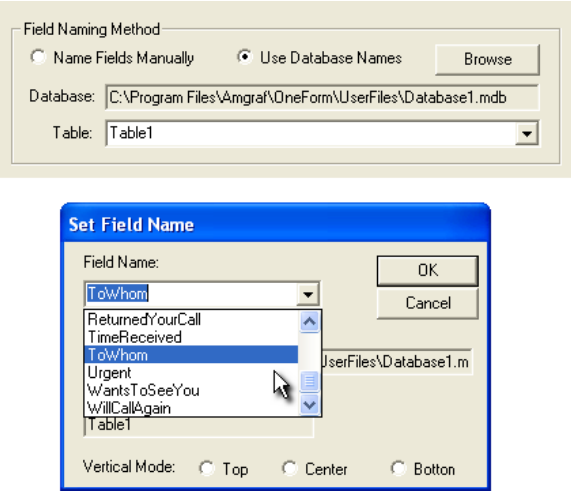
Figure 3-3: Auto Field Attributes

Gutter Controls: Gutters are seen when using *Auto Detect Location* or *Draw Bounding Location*. You may set gutter values for all 4 sides. The gutters are used in calculating the position of the field so that it will not touch the lines. Gutters can be specified by an amount or a percentage. *Auto Size* applies gutters by a percentage of the field height. *Specified* wants values for each gutter.

Field Naming Methods

Name Fields Manually: This opens the Field Properties dialog box each time a field is drawn, allowing you to replace the default field name.

Use Database Names: This option allows you to access the field names from a Access table within an .MDB database so that you can correspond the form fields to the table fields. **Browse** for the database location and **select a table.** The *Set Field Name* dialog will appear with a list of table field names to choose from, each time a field is drawn.



Field Naming Method

○ Name Fields Manually ● Use Database Names

Database: C:\Program Files\Amgraf\OneForm\UserFiles\Database1.mdb

Table: Table1

Set Field Name

Field Name: ToWhom

ReturnedYourCall
TimeReceived
ToWhom
Urgent
WantsToSeeYou
WillCallAgain

Vertical Mode: ○ Top ○ Center ○ Botton

Figure 3-4: Auto Field Tool - Use Database Names

Field Locator Methods

There are five choices for assistance in locating the position of your field: Auto Field Detect, Draw Bounding Location, Detect Step & Repeat, Replace Boxes, and Replace Text Item.

Auto Detect Location is looking for a dynamic rectangle to provide two corners of a region within which it looks for rectangular box shapes. This is used for placing fields within boxes, rectangles, or line components. *Height, Field Position,* and *Gutter Controls* can be used with this option. *Detect Step & Repeat* can also be toggled on for ease of handling multiple fields.

Detect Step & Repeat works only with *Auto Detect Location*. This uses the same location method as Auto Detect Location, but in addition, will "Step" the field when the artwork warrants. For example, if there is a column, a field will placed in every box in the column. Although there are multiple fields showing within the column, the fields are considered to be one field and are controlled through the first field’s Dim/Step Tab. When that field is modified, they are all modified.

Draw Bounding Location uses your two picks to determine the location of a field, allowing you to still have a *Height, Field Position,* and *Gutters* defined through the AutoField Attributes. This is for drawing fields of a specific height, but with no retangular shapes to help determine the placement. If you wanted to place a field on a line, you would use this locator method and pick the line endpoints.

Replace Boxes: This option will replace a box with a field, simply by picking upon a box component. It doesn’t get much easier than this. The field dimensions and location are all determined by the box. This gives you perfectly placed fields in an instant. This might be used for checkboxes or memo boxes. This option disregards all settings in the Auto Field Attributes and uses the box for width, depth and location. It really does ***replace*** the box. If the field were to be deleted, you would see that the box component no longer exists.

Text Group to VARTEXT and Replace Text Items: These options create VarText Field Types that are used with the OFDP Stationery Option. When checked, simply click on a text item (or group) to place a field. No other picks are involved in determining the location or dimensions of the field. The field will encompass the entire area of the text item. The field will also pick up the properties of the text. The Font Style, Font Size, and Composition Mode of the fill data will be the same as the text selected. Its properties can be viewed in the Print Tab of Field Properties. The Field Height and Width are determined by the text itself. The Field Width can be overwritten by specifying a Width in the Auto Field Attributes.

NOTE: The only difference between "Text Group" and "Replace Text" is the first one works with a group of Text Components. If you still want the text itself, toggle off "Remove Replaced Text".

Drawing with the Auto Field Tool

Examples of drawing with the Auto Field tool are described below. You will get better results if you understand how each location option looks at the picks made.

Auto Field For Multiple Line Input

Auto Field For Multiple Line Input is used for creating text fields that will allow more than one line of input into the field. This option will ignore the *Field Height* and *Position.* You have control over the field location and dimensions by using **Draw Bounding Location.** The software will then determine the number of lines of text that will fit into the field according to the field height (that you’ve picked) and the Font Leading specified in the Print Tab of Field Properties. For example, if the field height is *66-points* and the leading is *11-points*, the maximum number of lines for the field will be *6 lines*. Use this option when creating message or memo boxes.

Step 1: Choose the Auto Field tool on the Tool bar.

Step 2: Choose Auto Field for Multiple Line Input and Draw Bounding Location.

Step 3: Pick the upper-left corner of the field.

Step 4: A rectangle will follow the cursor position. Pick the lower-right corner of the field.

Step 5: The Field Properties will appear. View the maximum number of lines allowed in the Input tab of Field Properties.

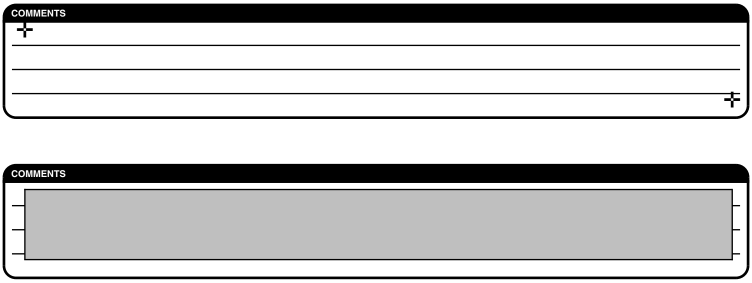


Figure 3-5: Auto Field For Multiple Line Input

Auto Detect/Step and Repeat

Auto Detect aids in placement of fields by looking at the base artwork's horizontal and vertical ruling to determine the positioning of the field. It uses the Field Height, Field Position, and Gutters as guides in determining the location and dimensions of your field.

Detect Step and Repeat can also be turned on with Auto Detect Location. When this is chosen, it will repeat your field, vertically or horizontally, according to the ruling.

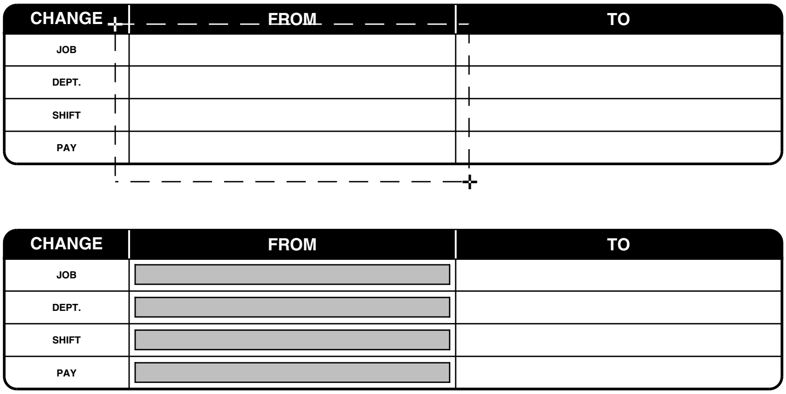


Figure 3-6: Auto Detect and Step & Repeat

Although the field will appear to be copied, the field is actually one field stepped X number of times. When clicked upon, they will appear to be one and the attributes will be identical. When Viewing the Field Names, the fields will be denoted as Field01(0), Field01(1), Field01(2), etc.

Step 1: Choose the Auto Field tool on the Tool bar.

Step 2: Input the Field Height, Field Position and Gutter values.

Step 3: Choose *Auto Detect*, and *Step and Repeat* in the Auto Field Attributes and [Okay].

Step 4: Pick the upper-left corner, picking outside of the column. The column must be surrounded in order for the software to see the ruling and therefore, determine the position.

Step 5: Completely surround the whole column with your dynamic rectangle. Make your second pick somewhere outside of the column.

Step 6: The Field Properties dialog box will appear. [Okay]. Watch as the computer steps your field vertically. View the properties of the steps under the **Dim/Steps** tab of Field Properties.

If you go through the same steps without *Detect Step & Repeat,* you will be making individual fields and will need to [Okay] the properties of each one as it is created.

Draw Bounding Location

Use *Draw Bounding Location* when you want the properties of the Auto Field Tool, but have no ruling for its placement. For example, when you want to place a field on a line.

Step 1: Choose the Auto Field tool on the Tool bar.

Step 2: Input the Field Height, Field Position and Gutter values.

Step 3: Choose *Draw Bounding Location* as the Field Locator Method [Okay].

Step 4: If this field were placed on a line, the first pick should be the left side of the line.

Step 5: Now, pick the right end-point of the line.

Step 6: The Field Properties dialog box will appear and [Okay].

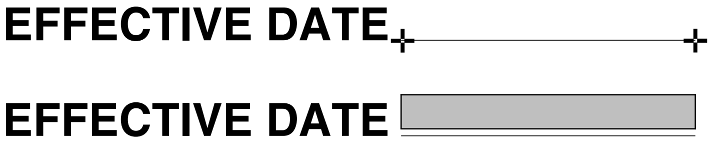


Figure 3-7: Draw Bounding Location on a Line

The field height and placement was all controlled by the Auto Field Attributes. The line was used as a guide. If you want the field to have the same length as the line make the Left and Right Gutters 0. The Bottom Gutter would be something like 3-points.

Modifying Field Properties, Location, and Dimensions

If you need to modify the properties of a field, you may do so through the Field's Pop-up menu. To access the Pop up menu, choose the **Select tool** and then **pick the Field.** You get to the Pop-Up menu by **right-clicking the mouse.** *Shortcut:* Double-clicking on a field will bring up its Modify Properties Dialog.

Note: If you do not see a Basic and Advanced selection, you are using the default "Advanced E-Form Field Properties" dialogs.

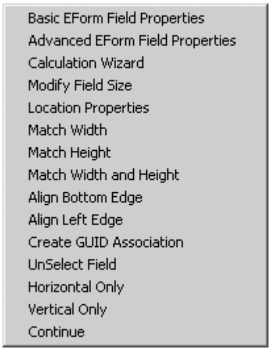


Figure 3-8: The Field Pop-Up Menu

EForm Field Properties (Basic): This will take you into that field's basic properties. At this time you can change the Field Name, Type, and any other basic properties shown.

Advanced EForm Field Properties: This will take you into that field's advanced properties. These dialogs have all options available and is usually made the default through the Options, Program Options menu.

Modify Field Size: Pick on the side of the field you want to modify and then pick its new location.

Location Properties: This will display the X and Y coordinates of the field and its size. Edits to location and size can be made here.

Match Width, Match Height, and **Match Height and Width** all work with Grouped fields only. They allow you to pick one size and have the others match it.

Group Select the fields you want to change, as well as, the field that will be your model (the field you are copying the dimensions from). Next, choose the Select tool and pick the model field, Right-click and choose *Match Height and Width*. All of the Grouped fields will be modified to match the height and width of the grouped model field.

Align Bottom Edge and **Align Left Edge** also work with Grouped fields only. This aligns the bottom or left edges of fields to another.

Group the fields that you want to align bottoms, as well as, the model field. With the Select tool, select the model field (the field you want to align the other fields to). Right-click to open the pop-up menu and choose *Align Bottom Edge*. The bottom edge of all the fields that have been grouped will align to the grouped model field.

Create GUID Association

There are times when you want a user input field and its label to stay together. This is accomplished by using Field's Pop-up Menu **"Create GUID Association"** option. GUID Association needs at least a field and piece of text.

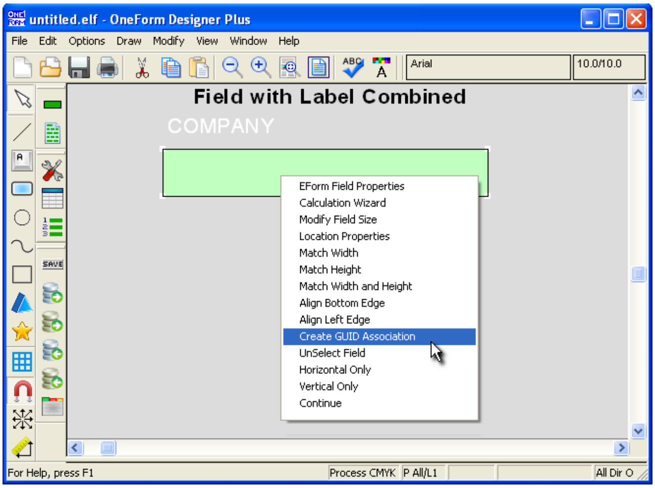


Figure 3-9: Combining Label and Field

Group the components that need linked and pick the field. Right-mouse and select the "Create GUID Association" menu and then pick a piece of text. This will link all components in the group to that field through its GUID number.

You can use this same technique to combine any objects, such as symbols or logos. Just "Disable" the field after it has been associated. It will still appear in your field list and will be editable, but will not be seen during generation.

Looking at Field Properties and Information

There are several other ways to look at the field properties, names, values, and information. The most obvious is the *Field List* tool for seeing a list of all fields on the form. It has a choice of informational columns that can be displayed for the fields, such as stepping, font, or group name. With a right-click, the field's properties will open for viewing or modification.

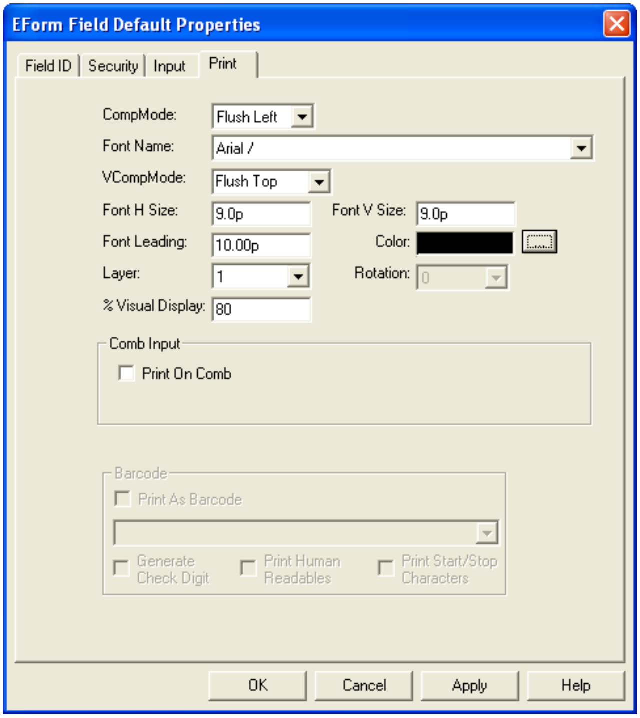
The *Options, EForm Options* menu has many options that are already tool picks, but one that is only found here is *Eform Field Report.* This option generates a report of all fields and their properties. It will save it in your project folder as a text file and then open Notepad to display the form information. It starts with the information about the form including custom programming to your form. Then has information concerning each field on the form. At the very bottom of the report is a field summary listing field types and how many are on the form.

In *View, EForm Fields* there are options to display field names, tabbing order, data table column names, etc on top of the fields. Picking "Normal Fields" will turn off the displays, and "No Fields" will visually hide the fields if you need to see just the form artwork.

Set E-Form Field Properties

The Set E-Form Field Properties tool sets the properties of fields before the field is drawn. In most cases, it is not necessary to set the properties before drawing since the properties appears each time a field is drawn. It is helpful, for example, to set the properties if you need the same print size or numeric formatting on the form.

See *Chapters 5 on Field Properties* for complete details on all of the properties and tabs available.



CompMode: Flush Left
Font Name: Arial /
VCompMode: Flush Top
Font H Size: 9.0p
Font V Size: 9.0p
Font Leading: 10.00p
Color: ■
Layer: 1
Rotation: 0
% Visual Display: 80

Comb Input
□ Print On Comb

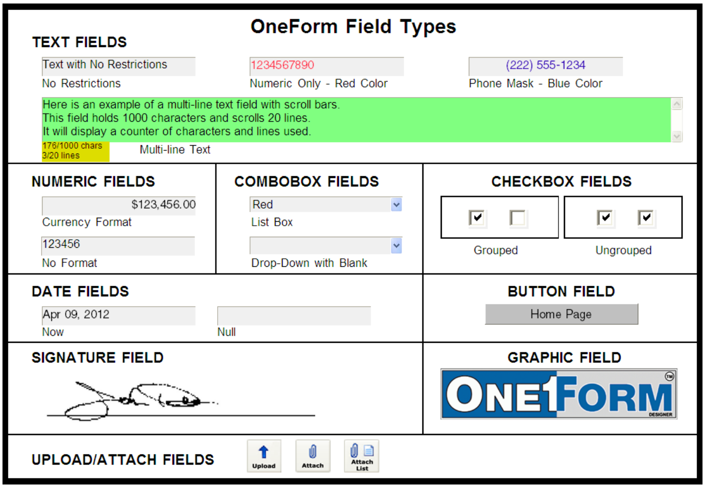
Barcode
□ Print As Barcode
□ Generate Check Digit
□ Print Human Readables
□ Print Start/Stop Characters

Figure 3-10: Set E-Form Field Properties Dialog

CHAPTER 4

Field Types

The Field Type is an identifier as to what is going to happen with that field when displayed on the form. There are eleven basic types, each one doing a particular action. For example, Text allows the operator to simply cursor to the field and type, which is why it is the default. CheckBox wants a click from the mouse to place a check in the box.



OneForm Field Types

TEXT FIELDS

Text with No Restrictions
No Restrictions

1234567890
Numeric Only - Red Color

(222) 555-1234
Phone Mask - Blue Color

Here is an example of a multi-line text field with scroll bars.
This field holds 1000 characters and scrolls 20 lines.
It will display a counter of characters and lines used.

176/1000 chars
3/20 lines
Multi-line Text

NUMERIC FIELDS

$123,456.00
Currency Format

123456
No Format

COMBOBOX FIELDS

Red
List Box

Drop-Down with Blank

CHECKBOX FIELDS

☑ □
Grouped

☑ ☑
Ungrouped

DATE FIELDS

Apr 09, 2012
Now

Null

BUTTON FIELD

Home Page

SIGNATURE FIELD

GRAPHIC FIELD

UPLOAD/ATTACH FIELDS

Figure 4-1: Different Types of Fields

The Field Type can be selected when an electronic field is created, or it can be changed later through the Field's Properties found on the selected component's Pop-up menu.

The Field Type identifier is found on the Field ID tab. You simply select one of the types from the drop-down list. *See Figure 4-2.*

OneForm's Basic Field Types:

TEXT *(Default)*
NUMERIC
COMBOBOX
CHECKBOX
CHECKBOX ARRAY

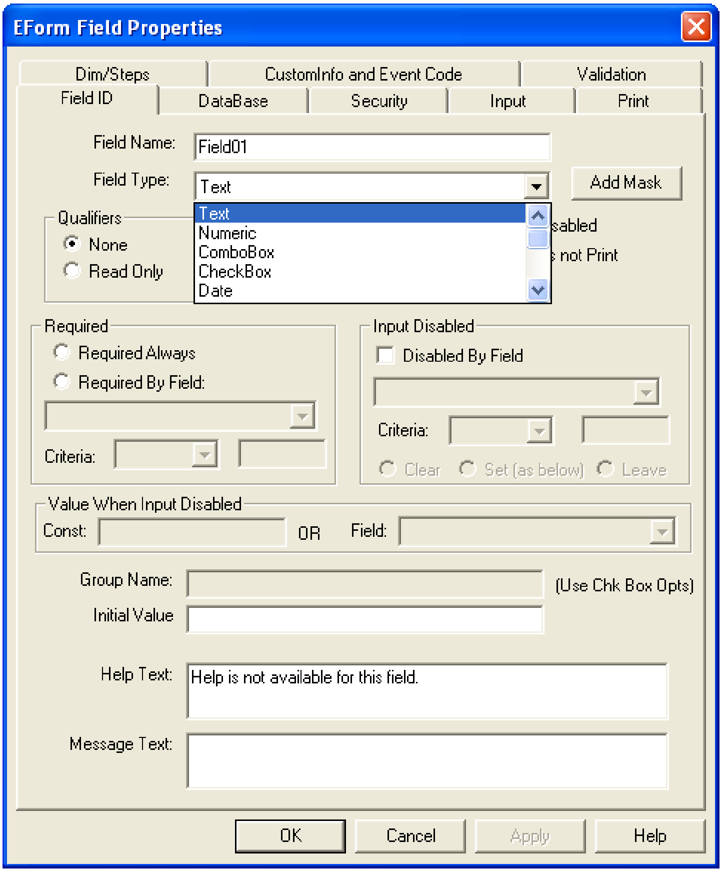
DATE/DATETIME
BUTTON
GRAPHIC/LAYOUT/OBJECT
SIGNATURE *(See Chapter 12)*
UPLOAD/ATTACH

Types used in Optional Software:

DATAMATRIX BARCODE
PDF417 BARCODE
EXCLUSION AREA

INCLUSION AREA
FOCUS
VARTEXT

As a field type is chosen, additional options become available that further define the characteristics of the field.



Field Name: Field01

Field Type: Text

Text
Numeric
ComboBox
CheckBox
Date

Qualifiers: ● None ○ Read Only

○ Required Always ○ Required By Field:

□ Disabled By Field

○ Clear ○ Set (as below) ○ Leave

Help Text: Help is not available for this field.

Figure 4-2: Field Type Drop-Down List

Every field has default properties to control its actions. Be sure to check out all of these settings in *Chapters 5: Set Field Properties.*

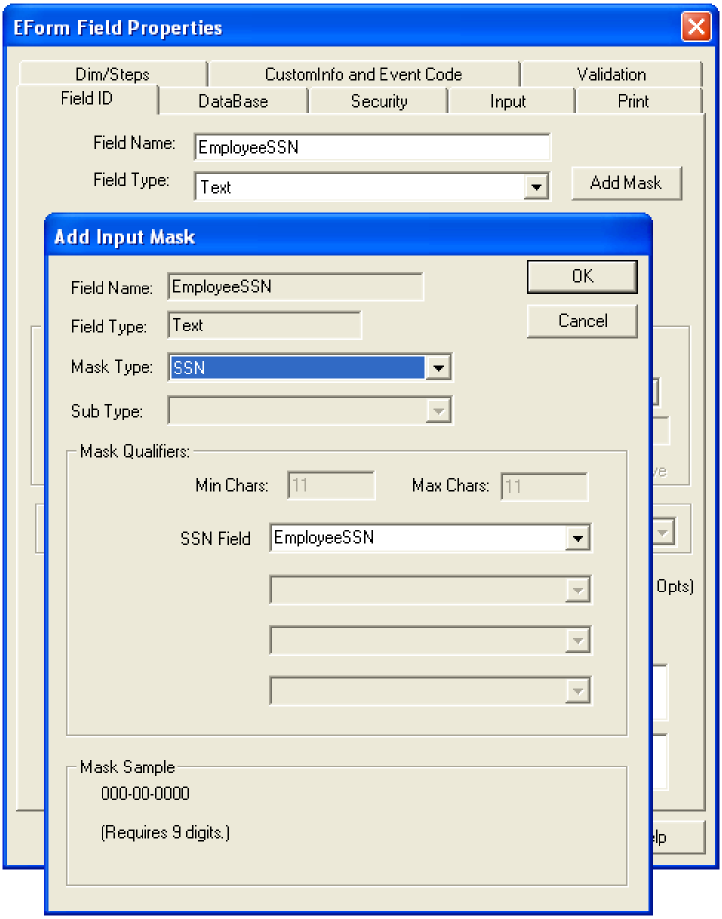
TEXT

This can be any keyboard input. Default is 50 characters on 1 line, but it can be changed on the *Input* tab. Input qualifiers include *Upper Case, Lower Case, or Numeric (numbers only).* Character limitations may be placed on the field by specifying the total *Maximum Characters* and *Maximum Lines* allowed in the field. A "0" in Maximum Characters, means unlimited.

[Add Mask]

A formatting mask can be added to text fields. This controls the look of the data after the user tabs out of the field. Common masks are phone numbers and Social Security Number. OFDP has a wide variety of input masks: Bank Number, City State Zipcode, Federal ID, Pad Zero, Percent, Social Security Number, TimeString, and many others.

Our example puts hyphens into a Social Security Number. Click the [Add Mask] button, followed by selecting the Mask Type: SSN, as shown.



Field Name: EmployeeSSN

Field Type: Text

Field Name: EmployeeSSN

Field Type: Text

Mask Type: SSN

Sub Type:

Min Chars: 11 Max Chars: 11

SSN Field EmployeeSSN

Mask Sample

000-00-0000

(Requires 9 digits.)

Figure 4-3: Click the [Add Mask] Button and Select SSN

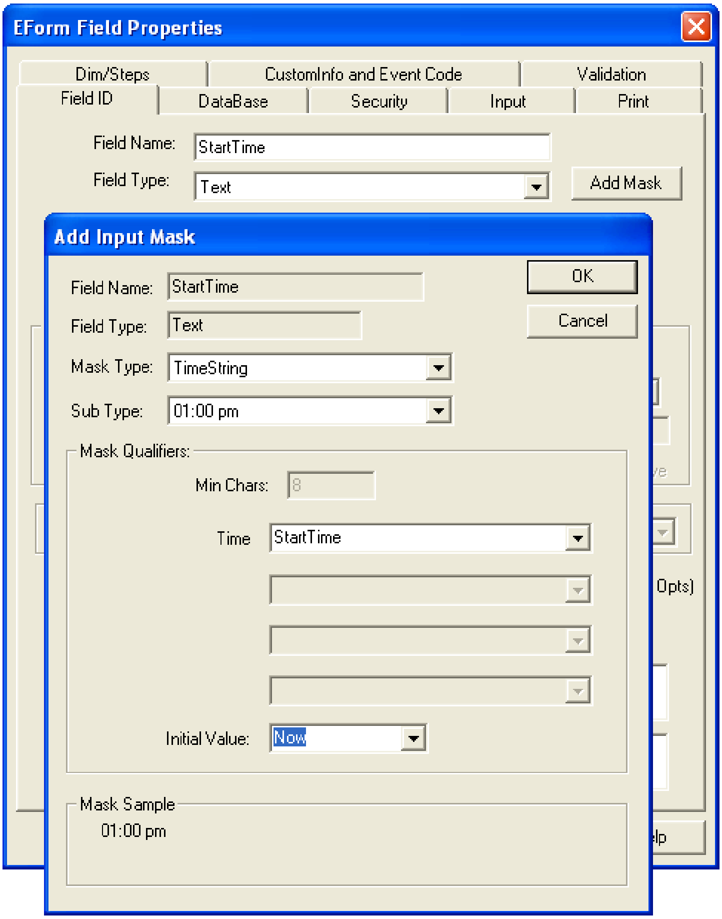
Needing Numbers Only

If you have a field that should be restricted to numbers only, such as "Number of Weeks Needed?", keep it a TEXT field type with a *Numeric Only* qualifier under the Input tab. You will want to specify numeric input for TEXT fields, when there is no currency formatting or no computations involved.

Needing a Time Format

Use TEXT field with a TimeString mask to get a time format.

After Text has been chosen as the field type, click the [Add Mask] button. From the Mask Type list select TimeString and choose a sub-type for the display (i.e, 01:00 pm) and the initialize type (i.e., Now). This will cause the computer's time to display in the text field.



EForm Field Properties

Field Name: StartTime

Field Type: Text

Add Input Mask

Field Name: StartTime

Field Type: Text

Mask Type: TimeString

Sub Type: 01:00 pm

Mask Qualifiers:

Min Chars: 8

Time StartTime

Initial Value: Now

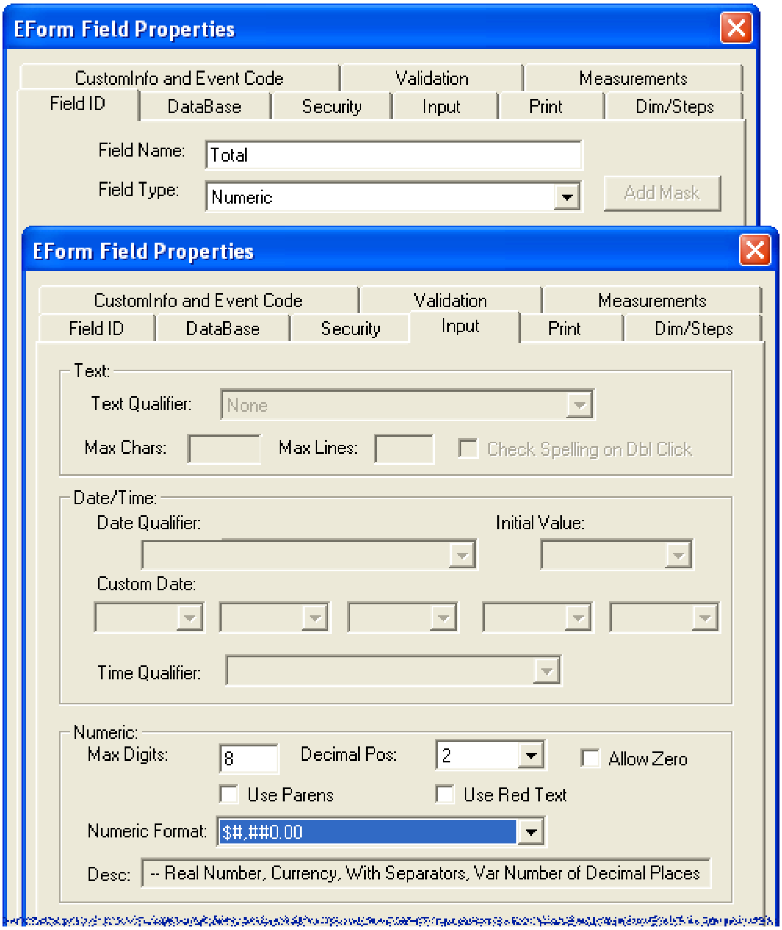
Mask Sample

01:00 pm

Figure 4-4: Using the Timestring Mask Type

NUMERIC

This will only allow numeric input and is normally used for fields involving currency formats or computations.



Field Name: Total

Field Type: Numeric

Text Qualifier: None

Max Chars: Max Lines: □ Check Spelling on Dbl Click

Max Digits: 8 Decimal Pos: 2 □ Allow Zero

□ Use Parens □ Use Red Text

Numeric Format: $#,##0.00

Desc: -- Real Number, Currency, With Separators, Var Number of Decimal Places

Figure 4-5: Numeric Field with Formatting

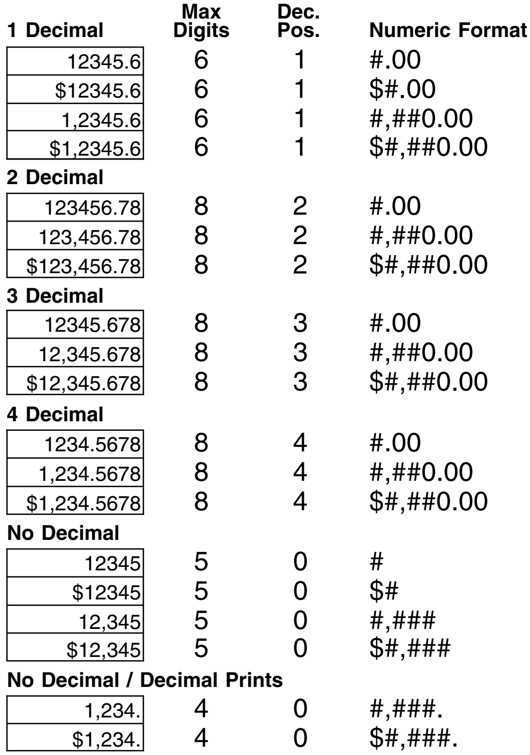
Character input limitations and formats may be applied to Numeric fields through the Input tab using **Maximum Digits, Decimal Positions,** and **Numeric Format** for currency or percentages.

The Advanced Field Properties Input tab has a drop-down of choices for ease of use.

See Figure 4-6 for a variety of examples using numeric formats with the maximum digits and decimals.

Numeric Format Examples for E-Forms

Max Digits: is how many keystrokes are allowed (0=unlimited).
Decimal Position: controls how many digits allowed after the decimal.
Numeric Format: is the format shown exactly regardless of Maximum Digits needed.



1 Decimal	Max Digits	Dec. Pos.	Numeric Format
12345.6	6	1	#.00
$12345.6	6	1	$#.00
1,2345.6	6	1	#,##0.00
$1,2345.6	6	1	$#,##0.00
2 Decimal			
123456.78	8	2	#.00
123,456.78	8	2	#,##0.00
$123,456.78	8	2	$#,##0.00
3 Decimal			
12345.678	8	3	#.00
12,345.678	8	3	#,##0.00
$12,345.678	8	3	$#,##0.00
4 Decimal			
1234.5678	8	4	#.00
1,234.5678	8	4	#,##0.00
$1,234.5678	8	4	$#,##0.00
No Decimal			
12345	5	0	#
$12345	5	0	$#
12,345	5	0	#,###
$12,345	5	0	$#,###
No Decimal / Decimal Prints			
1,234.	4	0	#,###.
$1,234.	4	0	$#,###.

Figure 4-6: Examples of Numeric Formats

Suppressing zeros for total fields requires coding to be done within **Form Preferences.** For more examples see the *Forms Library.*

HTML Calculations

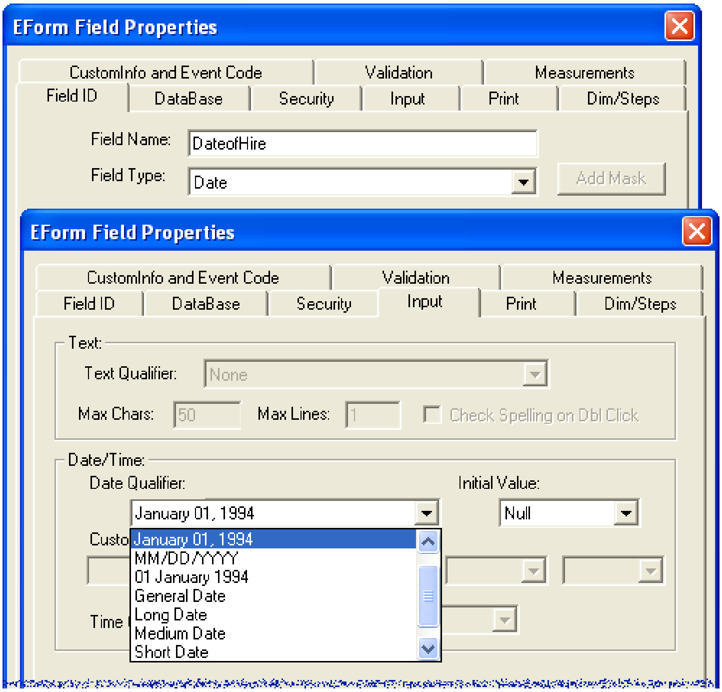
```
// suppress zeros
   _ThisForm["tot10"].value = FPNFormatZS
   (_ThisForm["tot10"],
   _fldsFmt["tot1"]);
```

PDF Calculations

```
// suppress zeros
   if (_ThisForm["tot1"].value == 0)
   {_ThisForm["tot1"].value =  "";}
```

DATE/DATETIME

When the date field is double-clicked in the HTML, a calendar will appear. The user can click on the calendar to select a date or simply input one of their own as in the PDF E-Form. The field can also pull the computer's date.



Field Name: DateofHire

Field Type: Date

Text Qualifier: None

Max Chars: 50 Max Lines: 1 ☐ Check Spelling on Dbl Click

Date Qualifier: January 01, 1994

Initial Value: Null

January 01, 1994
MM/DD/YYYY
01 January 1994
General Date
Long Date
Medium Date
Short Date

Figure 4-7: Date Field Qualifier

Under the Input tab, the **Date Qualifier** sets the look of the date, while the **Initial Value** indicates whether the computer's date is to populate the field or not.

Select **Now** to input the computer's date.

Select **Null** to prohibit the computer's input.

Getting a Time

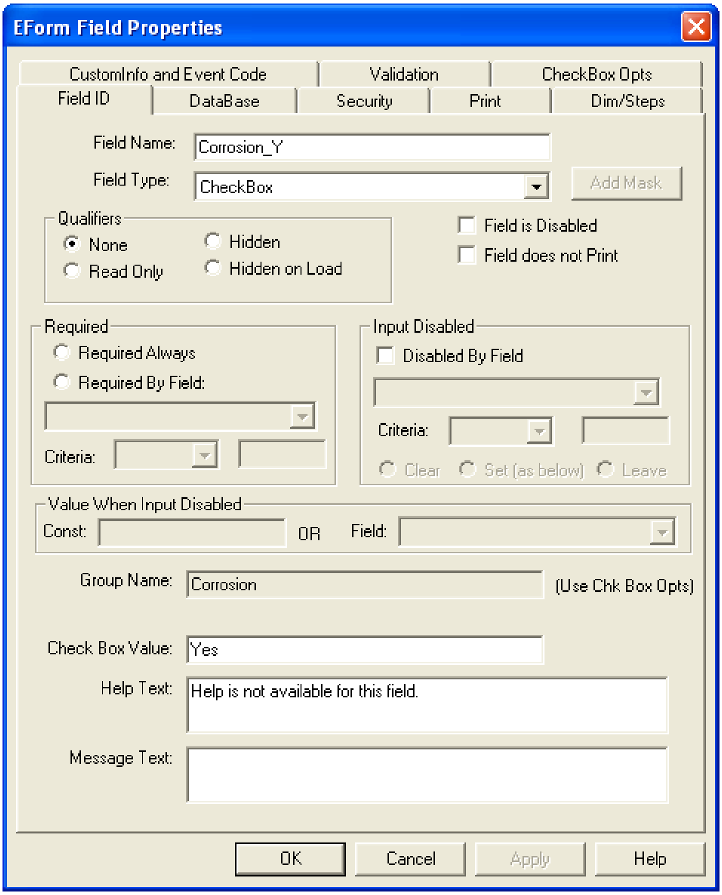
The TEXT field type has a mask option that can display time, so use a TEXT field instead with a TimeString mask as described with Figure 4-4.

DateTime Field

The DATETIME field type has a specific use for forms using custom intelligence. It is a read-only field that can be hidden or visible, but was developed to capture a programmed date and time, such as when a form was submitted. It has database DATETIME field properties, and records both the date and time, where the OFDP DATE field type only records the date.

CHECKBOX

This field type checks the box when clicked with the mouse or spacebar, if tabbing to it. They can be independent or assigned to groups where only one of the group can be checked.



Field Name: Corrosion_Y

Field Type: CheckBox

Qualifiers

● None
○ Read Only
○ Hidden
○ Hidden on Load

□ Field is Disabled
□ Field does not Print

Required

○ Required Always
○ Required By Field:

Input Disabled

□ Disabled By Field

Value When Input Disabled

Const: OR Field:

Group Name: Corrosion (Use Chk Box Opts)

Check Box Value: Yes

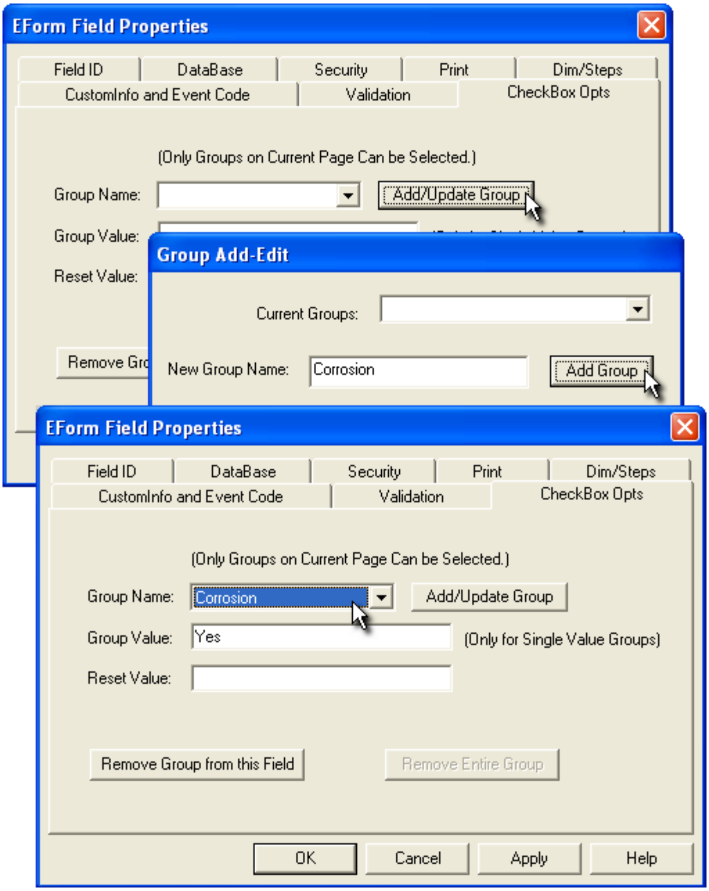
Help Text: Help is not available for this field.

Message Text:

Figure 4-8: Basic CheckBox Field

To group checkboxes, select the same **Group Name** for each field. This makes them behave like radio buttons. For example, when a field in a group series is checked, all other checkboxes in the group will become unchecked. Only one selection can be made for the group of fields.

You can have many checkbox groups, providing the Group Names are unique. The Group Name must start with an alpha character, be at least 3 alpha/numeric characters, and cannot include spaces. If you want to know what fields are grouped, you can display a column for group names under the Field List Dialog.



EForm Field Properties

(Only Groups on Current Page Can be Selected.)

Group Add-Edit

Current Groups:

New Group Name: Corrosion

EForm Field Properties

(Only Groups on Current Page Can be Selected.)

Group Name: Corrosion

Group Value: Yes (Only for Single Value Groups)

Reset Value:

Figure 4-9: Creating a Group Name

Create a group by picking **[Add/Update Group],** type in the group name assignment, then **[Add Group], [OK].** Now it is available for any checkbox when you look at the **Group Name** list.

Values Associated with CheckBoxes

The Form Global Option "Single Database Value for Groups" in the Form Preferences, will generate a single database table output column (Corrosion) per Checkbox Group, and give you the ability to designate the values (Yes or No) associated with each checkbox.

Drawing Your CheckBoxes

HTML E-Forms: Checkbox fields are not the size drawn and the type of browser along with HTML Compatibility Type will determine what you get.

Using Windows IE with Old, Loose, or Strict Compatibility types 10-point through 15-point look about the same, 16-point through 20-point start showing up as larger. They will display with a border, and use the checkmark (✔).

Using Windows IE with HTML5 Compatibility type 10-point through 12-point look about the same, but then each size starts showing up as larger. They will display with a border, and use the checkmark (✔).

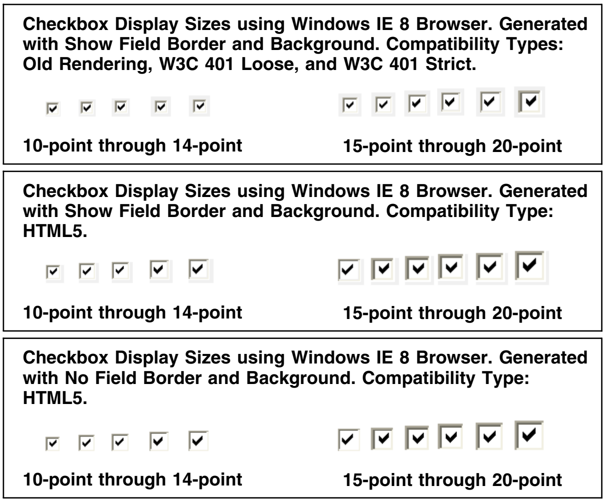

Checkbox Display Sizes using Windows IE 8 Browser. Generated with Show Field Border and Background. Compatibility Types: Old Rendering, W3C 401 Loose, and W3C 401 Strict.
10-point through 14-point
15-point through 20-point
Checkbox Display Sizes using Windows IE 8 Browser. Generated with Show Field Border and Background. Compatibility Type: HTML5.
10-point through 14-point
15-point through 20-point
Checkbox Display Sizes using Windows IE 8 Browser. Generated with No Field Border and Background. Compatibility Type: HTML5.
10-point through 14-point
15-point through 20-point

Figure 4-10: HTML Display of CheckBox Fields

If you are going to use Windows Print for the HTML, you can associate a different print character (such as an X) through EForm Preferences, [Edit Form Global Information], HTML Forms, Checkbox Print Character. The the field's Print Tab controls the character's font/size.

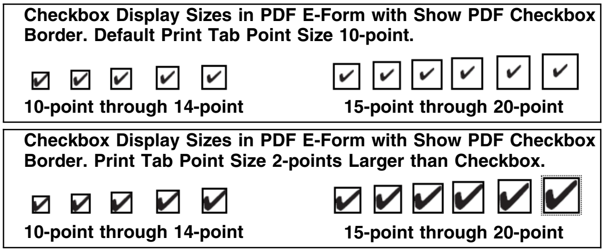
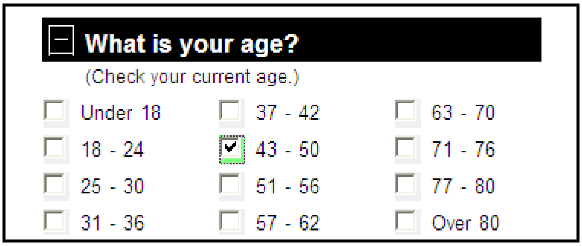

Checkbox Display Sizes in PDF E-Form with Show PDF Checkbox Border. Default Print Tab Point Size 10-point.
10-point through 14-point
15-point through 20-point
Checkbox Display Sizes in PDF E-Form with Show PDF Checkbox Border. Print Tab Point Size 2-points Larger than Checkbox.
10-point through 14-point
15-point through 20-point

Figure 4-11: PDF Display of CheckBox Fields

PDF E-Forms: Checkbox field is size drawn, 10-point through 14-point is best. Character size is controlled by the Print Tab. The Show PDF Checkbox Border option gives you a 1-point rule. If you need a thinner one, you can draw a box component instead. The character by default is the checkmark (✔).

CHECKBOX ARRAY

The Checkbox Array Field Type simplifies the layout of forms with choice list sections. OFDP creates a labeled checkbox for each record in a data table column. This function is used to automatically place the checkbox fields, labels, and JavaScript code within the E-Form source file to support the PDF printed layout and the E-Form expand/compress action.



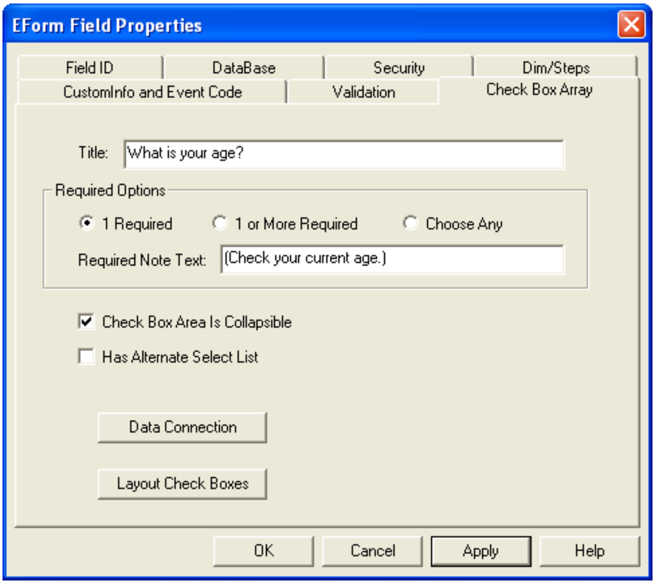
⊟ **What is your age?**

(Check your current age.)

□ Under 18	□ 37 - 42	□ 63 - 70
□ 18 - 24	☑ 43 - 50	□ 71 - 76
□ 25 - 30	□ 51 - 56	□ 77 - 80
□ 31 - 36	□ 57 - 62	□ Over 80

Figure 4-12: Example of CheckBox Array HTML

To start, create a CHECKBOX ARRAY field on the form that is the overall size of the section containing the check boxes. The following CheckBox Array panel will appear allowing you to create the list with the desired attributes.



Title: What is your age?

Required Options

● 1 Required ○ 1 or More Required ○ Choose Any

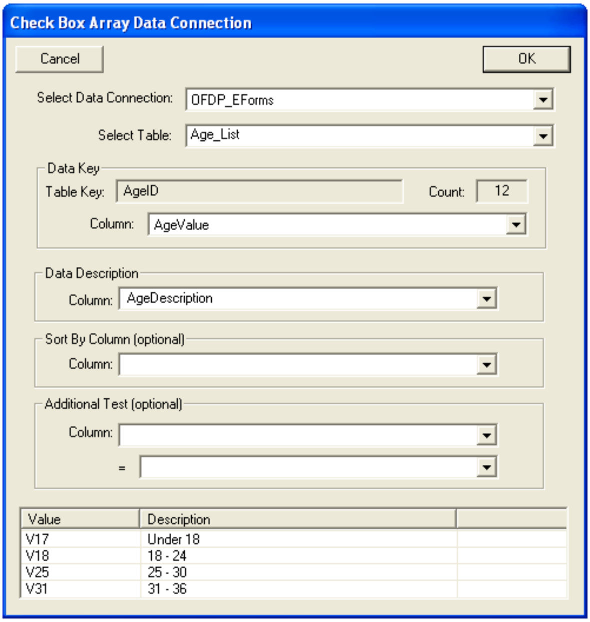
Required Note Text: (Check your current age.)

☑ Check Box Area Is Collapsible

□ Has Alternate Select List

Figure 4-13: CheckBox Array Panel

You can indicate if the check box array is collapsible, and if an associated dropdown list is to be created along with the array of checkboxes. Clicking the [Data Connection] button will connect the check box array to a data table of choices. The following input panel will be displayed:



Check Box Array Data Connection

Select Data Connection: OFDP_EForms

Select Table: Age_List

Data Key
Table Key: AgeID   Count: 12
Column: AgeValue

Data Description
Column: AgeDescription

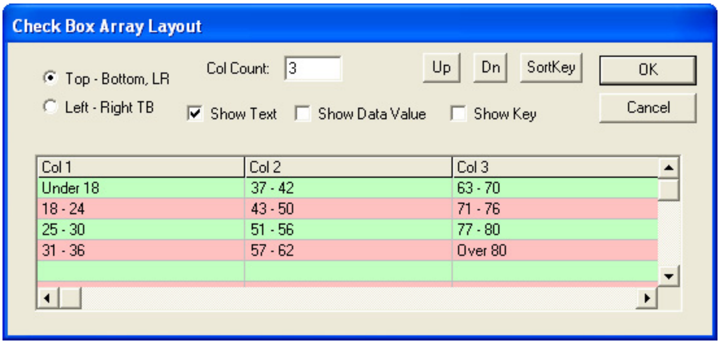
Sort By Column (optional)
Column:

Additional Test (optional)
Column:
=

Value	Description	
V17	Under 18	
V18	18 - 24	
V25	25 - 30	
V31	31 - 36	

Figure 4-14: Associating a Data Table to the CheckBox Array Field

Clicking the [Layout CheckBoxes] button brings up the following panel for arranging the data table list names in the desired order. The software will automatically space the checkboxes into rows and columns that fit within the size of the field.



Check Box Array Layout

● Top - Bottom, LR
○ Left - Right TB

Col Count: 3

☑ Show Text □ Show Data Value □ Show Key

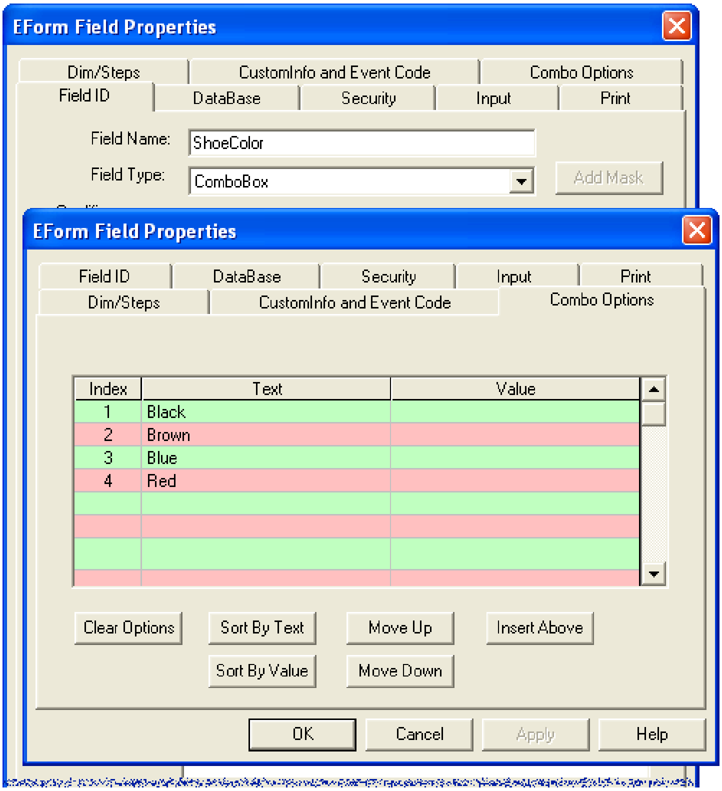
Col 1	Col 2	Col 3
Under 18	37 - 42	63 - 70
18 - 24	43 - 50	71 - 76
25 - 30	51 - 56	77 - 80
31 - 36	57 - 62	Over 80

Figure 4-15: Arranging the Data Table Names into Columns

Printing is always done using the PDF generated copy, that is not a collapsed form.

COMBOBOX

When there is a multiple choice, use a ComboBox field. This type of field creates a drop-down box with a list of choices. These choices can be typed into the field's Combo Options Tab, or pulled from a database.



Field Name: ShoeColor

Field Type: ComboBox

Index	Text	Value
1	Black	
2	Brown	
3	Blue	
4	Red	

Figure 4-16: Setting the Choice List within a Field

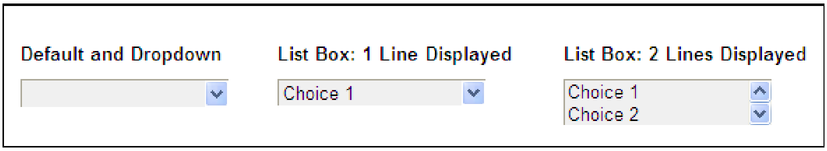
The Combo Options Tab will only be available if the Field Type is a ComboBox. It will always start out with *Choice 1* and *Choice 2*. Just click in the "Text" column and type in the text needed. Do this for each item needed in the list.

The Value column is used, if you need to store a different value in the data table, other than the "Text" column.

Getting the Choice List from a Database

The ComboBox list can be pulled from a data table column. See *Chapter 8: Associating Databases* for an example. When this is done, the Combo Options Tab will need to be cleared of all choices.

Different ComboBox Looks



Default and Dropdown

List Box: 1 Line Displayed

Choice 1

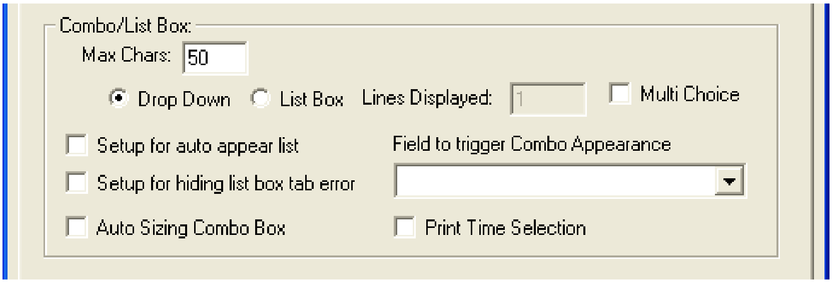
List Box: 2 Lines Displayed

Choice 1
Choice 2

Figure 4-17: Default is DropDown

The default is a "DropDown" select list, that holds 50 characters in the data table field. It starts with a blank and you must select from the dropdown list to make a choice. "List Box" has options for number of lines displayed and how many the user can highlight. It does not have a blank and by default has the first item selected.

Optional Features on the Input Tab



Combo/List Box:

Max Chars: 50

● Drop Down ○ List Box Lines Displayed: 1 □ Multi Choice

□ Setup for auto appear list

Field to trigger Combo Appearance

□ Setup for hiding list box tab error

□ Auto Sizing Combo Box

□ Print Time Selection

Figure 4-18: Optional Features on the Field Properties Input Tab

Max Chars: The ComboBox field creates a text field type (varchar) in a data table. The default is 50 characters, just like the TEXT fields.

The *Auto Sizing ComboBox* will adjust the width of the field in the HTML to fit the list. This is recommended if the list is coming from a data table.

Field to Trigger Combobox Appearance: The ComboBox field can be hidden and made visible when focusing off of the selected field.

With the PDF Option, *Print Time Selection* causes the text column to print, not the values stored in the data table.

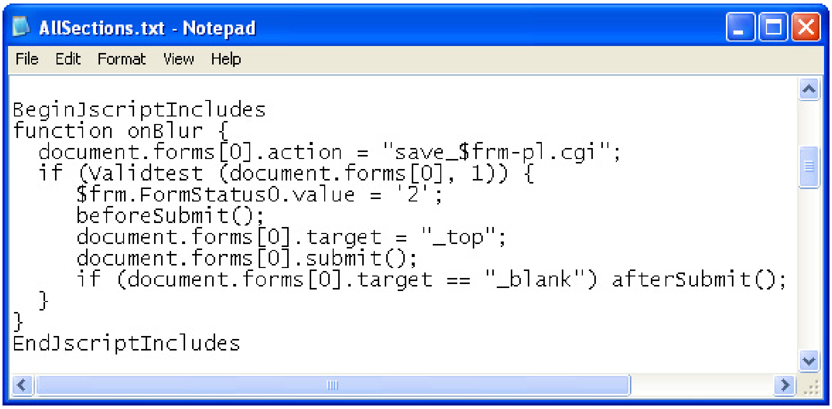
Getting the Choice List from a Database

The ComboBox list can be pulled from a data table column. See *Chapter 8: Associating Databases* for an example. When this is done, the Combo Options Tab will need to be cleared of all choices.

BUTTON

This is a field type that when clicked upon will react according to its programming. It can bring up a new program or send the user to another page.

The default of the Button is to **Use Button to Execute Code.** This has the button execute the code within the JscriptInclude strings. Below is an example of the code in OFDP's SubmitData button.



```
BeginJscriptIncludes
function onBlur {
  document.forms[0].action = "save_$frm-pl.cgi";
  if (Validtest (document.forms[0], 1)) {
     $frm.FormStatus0.value = '2';
     beforeSubmit();
     document.forms[0].target = "_top";
     document.forms[0].submit();
     if (document.forms[0].target == "_blank") afterSubmit();
  }
}
EndJscriptIncludes
```

Figure 4-19: CustomInfo and Event Tab

Button Caption

As soon as you choose the field type of Button, a *Caption* field will appear allowing you to type in the label for the Button. Default is to use the field name. You can size this caption through the Print dialog.

Insert JavaScript code into the Form Preferences to generate custom button code if you need it on two lines.

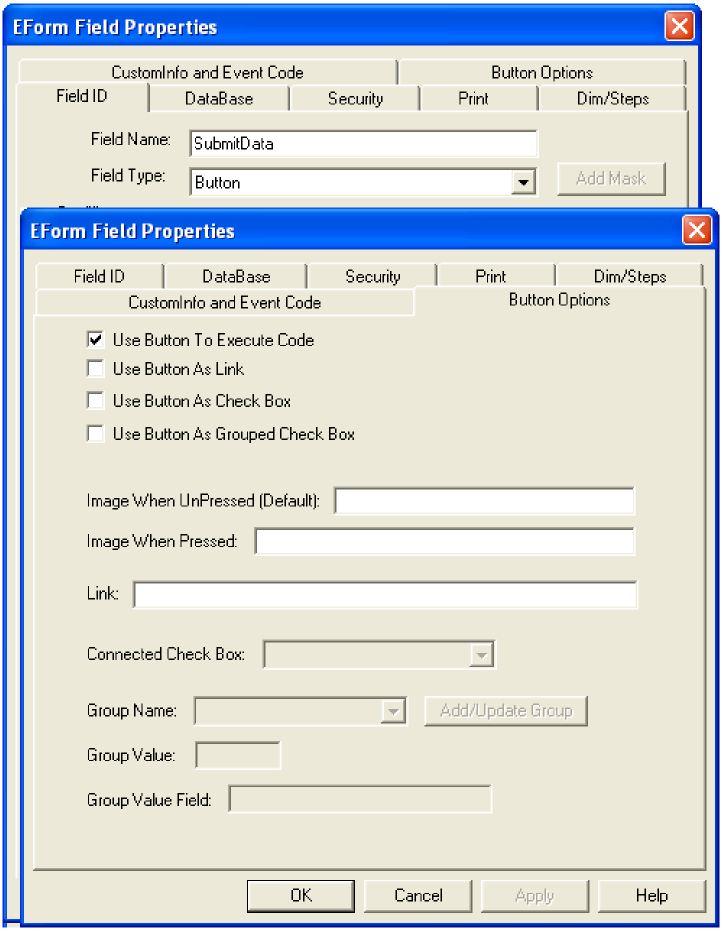
```
BeginJscriptIncludes
function initialize {
    $frm["Field01"].value = "caption 1\rcaption 2";
}
EndJscriptIncludes
```

The \r is a "return" code. You will not see the two line caption until the HTML is generated.

Button Options Tab

Use Button to Execute Code: This is the default action. A newly drawn button does not have any code. You will need to go the the Custom Information Tab and place JavaScript HTML code for its actions.

Use Button as Link: Type in the full path of the **Link** to be accessed.



Field Name: SubmitData

Field Type: Button

☑ Use Button To Execute Code

☐ Use Button As Link

☐ Use Button As Check Box

☐ Use Button As Grouped Check Box

Image When UnPressed (Default):

Image When Pressed:

Link:

Connected Check Box:

Group Name:

Group Value:

Group Value Field:

Figure 4-20: Button Options Tab

Use Button as Checkbox: This allows you to link images (pressed and unpressed) to your checkbox through the button. It requires that you have a hidden connected checkbox associated to store the "checked/unchecked" value.

Use Button as Grouped Checkbox: This allows you to link images (pressed and unpressed) to your grouped checkboxes through buttons. See "Advanced Checkbox Groups" for more information, since it behaves the same.

GRAPHIC

The Graphic field type is for pulling images into a form field position. It will pull in a .GIF or .JPG image into the position and size of the field. The image will be resized to the size of the field so take this into consideration. The Height and Width value is found on the Dims/Steps Tab.

The image to be referenced is placed into the CustomInfo of the Field Properties as shown:

```
BeginIformGraphicFiles
   "ImageName.jpg"
EndIformGraphicFiles
```

Be sure to surround the image name with quotes. It can be pathed if needed.

PDF E-Forms

The Graphic field type will become a Button field type in the PDF E-Form. The image file will need to be located within Acrobat through their field properties.

LAYOUT

The LAYOUT field defines an HTML Iframe within the HTML document. It has a Layout Tab to name the IFrame and apply properties.

OBJECT

The OBJECT field defines an embedded object within an HTML document. Use this element to embed multimedia (like audio, video, Java applets, ActiveX, PDF, and Flash) in your web pages.

You can also use the this to embed another webpage into your HTML document.

SIGNATURE

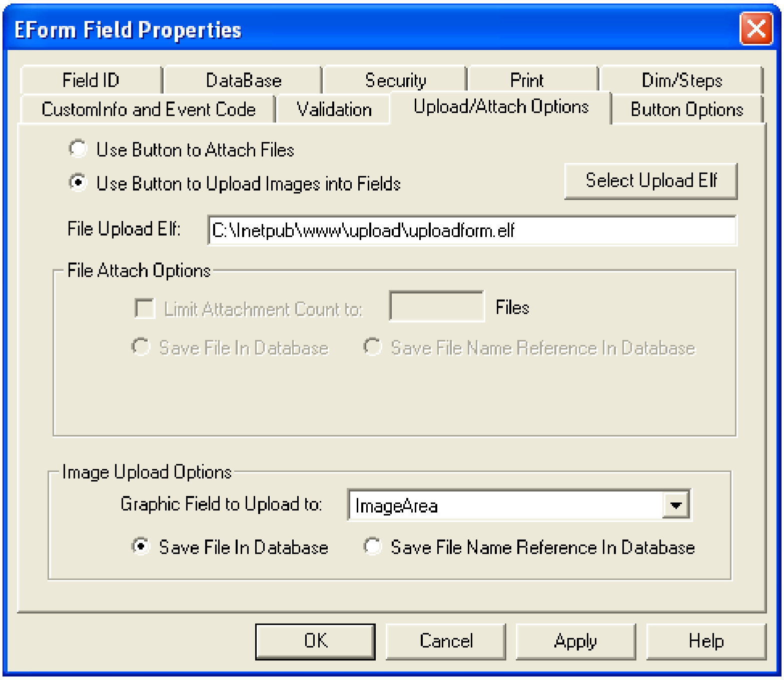
The Signature Field is a very unique type of field used for locking a group of visible fields on a form. Signatures are discussed in detail within *Chapter 12: Forms with Signature Fields.*

UPLOAD/ATTACH BUTTONS

This field type is used to build the fields, buttons, and programming for two different actions, both through the ease of its dialog shown below:

- Uploading of images into the E-Form allows the user to upload a photograph into a designated field. You can have several upload fields on one form if needed.
- Attaching files to the E-Form is similar to any standard email attachment. It can be very useful if you have Word documents or WinZip files that need saved with the E-Form.

When the field type "Upload/Attach" is chosen through the Advanced Field Properties, the Upload/Attach Tab will appear allowing you to choose which action you need.



○ Use Button to Attach Files

● Use Button to Upload Images into Fields

File Upload Elf: C:\Inetpub\www\upload\uploadform.elf

File Attach Options

□ Limit Attachment Count to: Files

○ Save File In Database ○ Save File Name Reference In Database

Image Upload Options

Graphic Field to Upload to: ImageArea

● Save File In Database ○ Save File Name Reference In Database

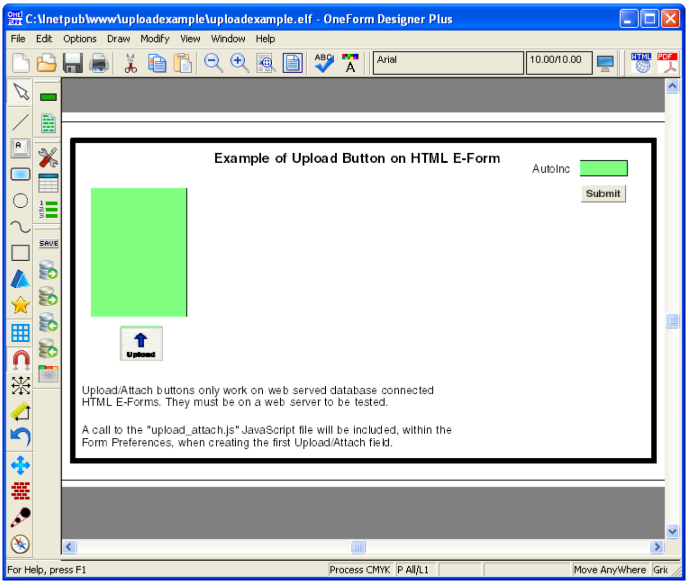
Figure 21: Upload/Attach Options Tab

Note: Upload/Attach buttons only work on web served database connected HTML E-Forms. They must be on a web server to be tested.

Creating the Upload Image Action

Placing an upload image action on your form requires two fields: a graphic field to hold the image and an upload button field.

When you draw the graphic field, it must be a Graphic Field Type. The size and placement of the Graphic Field is static and images will always upload into the field's upper-left corner, sized proportionally according to its largest side.


C:\Inetpub\www\uploadexample\uploadexample.elf - OneForm Designer Plus
File Edit Options Draw Modify View Window Help
Arial
10.00/10.00
SAVE
AutoInc
Submit
Upload
For Help, press F1
Process CMYK
P All/L1
Move AnyWhere
Gri

Example of Upload Button on HTML E-Form

Upload/Attach buttons only work on web served database connected HTML E-Forms. They must be on a web server to be tested.

A call to the "upload_attach.js" JavaScript file will be included, within the Form Preferences, when creating the first Upload/Attach field.

Figure 22: Drawing the Two Upload Fields

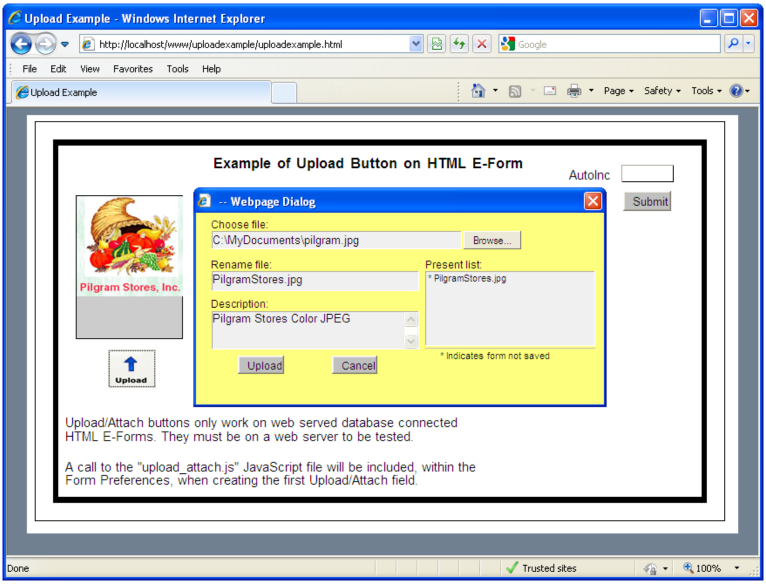
Create the Upload Button Field through the Advanced Field Properties and select Upload/Attach as the Field Type. Go to the Upload/Attach tab and select "Use Button to Upload Images into a Fields". The upload action references an ELF file that generates the dialog for the upload browsing. The default ELF is ..\upload\uploadform.elf.

Now, go down to the Image Upload Options and choose from the list a corresponding Graphic field for the "graphic field to upload to". Apply your selections and OK to save and close the dialog. Reloading your ELF will display our upload button image.

The Upload button will appear ready to be tested when your E-Form HTML is placed on your web server. A call to the server's "upload_attach.js" RefJavaScript file will be included within the Form Preferences, when creating the first Upload/Attach field.

Using the Upload Image Button

Clicking the Upload button will bring up the Browse Web Page Dialog as shown on the next page:



Upload Example - Windows Internet Explorer
http://localhost/www/uploadexample/uploadexample.html
Google
File Edit View Favorites Tools Help
Upload Example
Page Safety Tools
-- Webpage Dialog
Browse...
Upload
Cancel
Submit
Pilgram Stores, Inc.
Upload
Done
Trusted sites
100%

Example of Upload Button on HTML E-Form

AutoInc

Choose file: C:\MyDocuments\pilgram.jpg

Rename file: PilgramStores.jpg

Present list: * PilgramStores.jpg

Description: Pilgram Stores Color JPEG

* Indicates form not saved

Upload/Attach buttons only work on web served database connected HTML E-Forms. They must be on a web server to be tested.

A call to the "upload_attach.js" JavaScript file will be included, within the Form Preferences, when creating the first Upload/Attach field.

Figure 23: Browse for Image and click [Upload]

Here you can choose the image file to upload. These can be either GIF or JPEG images.

There are two optional fields for renaming the file before it is stored into the upload data table and for decribing the file within the data table. Clicking the [Upload] button will bring the image into the graphic field for display.

When the form is submitted to the server, a sub-table is created within the database to hold the upload image and its description.

Creating the Attach Files Action

Placing the attach files action on your form requires two button fields, but both are being drawn as you create the first attach button field.

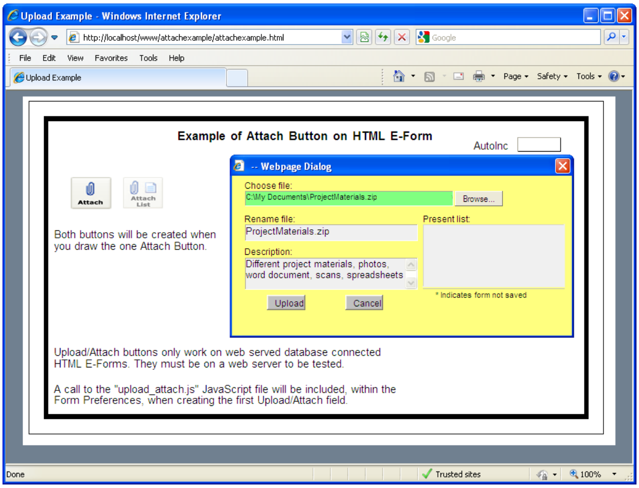
Create the Attach Button Field through the Advanced Field Properties and select Upload/Attach as the Field Type. Go to the Upload/Attach tab and select "Use Button to Attach Files". The attach action also references an ELF file that generates the dialog for the attach upload browsing. The default ELF is ..\upload\uploadform.elf.

Apply your selection and OK to save and close the dialog. Reloading your ELF will display our attach button images.

The Attach buttons will appear ready to be tested when your E-Form HTML is placed on your web server. A call to the server’s "upload_attach.js" RefJavaScript file will be included within the Form Preferences, when creating the first Upload/Attach field.

Using the Attach Buttons

Clicking the Attach button will bring up the Browse Web Page Dialog as shown below:


Upload Example - Windows Internet Explorer
http://localhost/www/attachexample/attachexample.html
Google
File Edit View Favorites Tools Help
Upload Example
Page Safety Tools
Attach
Attach List
-- Webpage Dialog
Browse...
Upload
Cancel
Done
Trusted sites
100%

Example of Attach Button on HTML E-Form

AutoInc

Choose file: C:\My Documents\ProjectMaterials.zip

Rename file: ProjectMaterials.zip

Present list:

Description: Different project materials, photos, word document, scans, spreadsheets

* Indicates form not saved

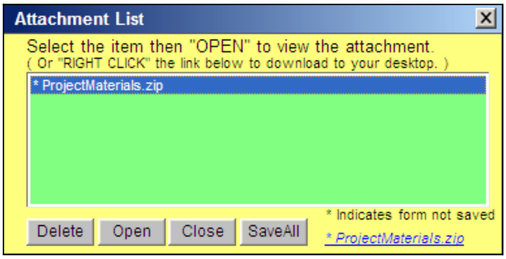
Both buttons will be created when you draw the one Attach Button.

Upload/Attach buttons only work on web served database connected HTML E-Forms. They must be on a web server to be tested.

A call to the "upload_attach.js" JavaScript file will be included, within the Form Preferences, when creating the first Upload/Attach field.

Figure 24: Browse for the File and click [Upload]

There are two optional fields for renaming the file before it is stored into the upload data table and for decribing the file within the data table. Clicking the [Upload] button will save the file into the "Attach List".



Select the item then "OPEN" to view the attachment.
(Or "RIGHT CLICK" the link below to download to your desktop.)

* ProjectMaterials.zip

* Indicates form not saved

* ProjectMaterials.zip

Figure 25: Attach List Button Shows a List of Files Attached

Attaching files to the E-Form is similar to any standard email attachment. It can be very useful if you have Word documents or WinZip files that need saved with the E-Form.

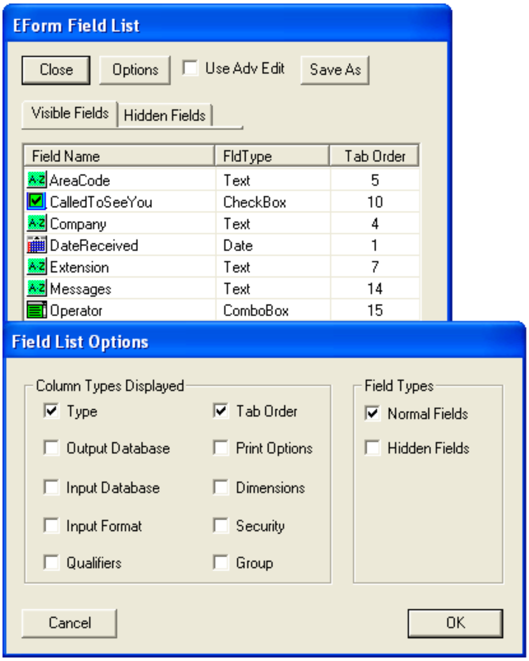
Clicking on the Link button will create a link associated to the file name highlighted. This link will allow you to open the file if you have the appropriate program on your local machine. In our example it would be WinZip.

When the form is submitted to the server, a sub-table is created within the database to hold the attached files and their descriptions.

FIELD LIST

The EForm Field List dialog can be found in EForm Options under the Options menu and as a tool on the fielding toolbar. This brings up a dialog box that lists all fields on your form. You can choose the field list display by clicking on the [Options] button. Sort your fields by clicking on the top of the column. You can sort by Field Type as shown.

When a field in the Field list is selected, the field on the form will become selected and display blue. This allows you to easily identify where the field is on the form.



□ Use Adv Edit

Field Name	FldType	Tab Order
AreaCode	Text	5
CalledToSeeYou	CheckBox	10
Company	Text	4
DateReceived	Date	1
Extension	Text	7
Messages	Text	14
Operator	ComboBox	15

Column Types Displayed

☑ Type
□ Output Database
□ Input Database
□ Input Format
□ Qualifiers
☑ Tab Order
□ Print Options
□ Dimensions
□ Security
□ Group

Field Types

☑ Normal Fields
□ Hidden Fields

Figure 4-26: Field List

Left-button click on a field in the list to rename it. Click it once and it will be highlighted, click it again and the rename border will appear. Type in the new name and then press [Enter] or click off of it.

Right-button click on a field in the list to bring up its Modify Properties dialog ready for you to review and make changes.

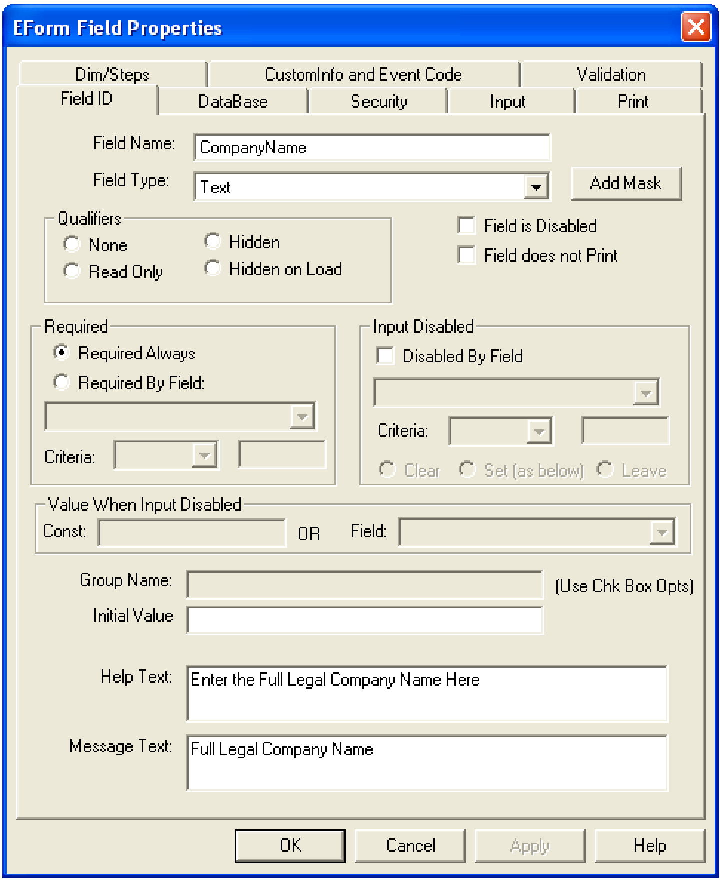
There are also options to always use the Advanced Field Properties dialog and to [Save] the listing information to a .csv file type.

CHAPTER 5

E-Form Field Properties

The E-Form Field Properties dialog appears each time a field is drawn. This gives the designer an opportunity to name the field and set the properties at the time the field is created.

This chapter describes in detail the standard Tabs available in the E-Form Field Properties.



EForm Field Properties

Field Name: CompanyName

Field Type: Text

Qualifiers: ○ None ○ Read Only ○ Hidden ○ Hidden on Load

□ Field is Disabled

□ Field does not Print

Required: ● Required Always ○ Required By Field:

Input Disabled: □ Disabled By Field

Group Name:

Initial Value

Help Text: Enter the Full Legal Company Name Here

Message Text: Full Legal Company Name

Figure 5-1: Field Properties Opens to the Field ID Tab

Double-clicking on a field with the Select cursor will bring up the field's properties for modifying. The option is found in the pop-up menu. Also, a right-mouse in the field list will bring up field properties.

Field Property Tabs

The Field Properties control the attributes of the field. It sets properties such as the type of input allowed into the field. There are tabs for the size and location, the print properties, and whether it is associated to a database table. The tabs within most fields are below.

- **Field ID:** This holds the main information concerning the field, such as the unique field name, the field type, and any standard qualifiers.
- **Database:** This shows the database tables that have been assigned to the field, both Input and Output.
- **Security:** This holds the security level value of the field for use with Signatures.
- **Input:** The input characteristics of the field are items such as how many characters a field can hold, or date formats, numeric formats, etc.
- **Print:** This controls the user input font display and print properties of the field’s filled-in text.
- **Dim/Steps:** This tab shows the width and height of the field, along with its location. It also has controls for stepping the field either horizontally or vertically to produce rows or columns of fields.
- **CustomInfo:** This is provided for those who need to add custom intelligence to a field through HTML JavaScript code. Also, it is where some of the field specific information gets stored. Such as Help Messages for the field and the list for a ComboBox field.
- **Validation:** The type of validation available is determined by the field type. A Required Field’s "Message Text" is a common use of this tab.

Field Type Specific Tabs

The Field Type specific tabs are described in *Chapter 4,* under their field types.

- **Button Options:** This tab defines the type of action for the button.
- **CheckBox Options:** This tab defines the CheckBox group list and values.
- **CheckBox Array Options:** This tab holds the CheckBox Array columns and their values.
- **Combo Options:** This tab holds the ComboBox list of choices and their values.
- **Measurement:** Numeric fields can have an Imperial to Metric conversion applied to a value and Date fields can have American/European conversion applied.
- **Object Properties:** This tab defines the multimedia Object properties.
- **Upload/Attach Options:** This tab holds the type of action for the upload and attach buttons.
- **Layout Properties:** This tab defines the Layout IFrame properties.

Field ID Tab

When drawing or modifying a field, it will always display this tab showing main information: Field Name, Field Type and Qualifiers.

Field Name: Every field on the ELF must have its own unique name. Field names must begin with an alpha and be at least 3 characters. The only punctuation allowed is an underscore (_). Fields are given names when they are created: Field01, Field02, etc., but we suggest you give it an English name for easier identification. For example, if the field is going to be an address, name the field *address*. This will make finding the field easier.

There is a "reserved" list of words that cannot be used as field names. If a name is used that is on the reserved list, a warning will appear stating that the name used is a reserved word. The name will immediately be changed to Field## at which time you may rename the field. For example, *Date* cannot be used, but *Date1* is acceptable.

Field Type: Each field must be assigned a field type. The default Field Type is TEXT. The Field Type is an identifier as to what is going to happen with that field. Each type behaves differently, doing a particular action. For example, Text allows the operator to simply cursor to the field and type. CheckBox wants a click from the mouse and places a checkmark (✔) in the box.

Select a field type from the list and press the [Tab] key. This will activate the other appropriate fields and Field Type Tabs for that type.

The Different Standard Field Types:

- **Text:** This allows any keyboard input.
- **Numeric:** This only allows numeric input and characters needed for values representing currency. This should be used when calculations are needed.
- **CheckBox:** This inputs a checkmark (✔) when clicked upon. If tabbing to the CheckBox, the [SpaceBar] will toggle it.
- **CheckBox Array:** Creates CheckBoxes from a list of choices.
- **ComboBox:** This creates a drop-down list of choices.
- **Date:** Inputs acceptable date formats.
- **DateTime:** Reads in a date and time for programming purposes.
- **Button:** Creates a field that when clicked on, an action takes place.
- **Graphic:** Places a graphical image on on the form.
- **Layout:** Places and IFrame in the HTML.
- **Object:** Use this to embed multimedia files into the HTML form.
- **Signature:** A unique type of field which is used for locking groups of visible fields through a security level and password.
- **Upload/Attach:** A unique type of field which is used to build the fields and actions for an upload photo button and an attach document button.

Qualifiers:

Qualifiers further define or limit users input. Qualifiers include: None, Read Only, Hidden, and Hidden on Load.

None allows the user to input in a field or leave it blank without any consequence. This is the default.

Read Only prohibits the user from inputting into the field. For example, Date fields may be Read Only if you only want the date input to come from the computer. Fields that calculate totals are often Read Only.

Hidden: This hides the field. The field is still usable, but is hidden from the user. For example, the field can hold data values, but does not display.

Hidden on Load: This hides the field when the E-Form is first opened. It then can be a conditional field turned visible through custom programming.

Other Options:

Field is Disabled: This makes a field completely ignored. As far as the HTML is concerned, the field does not exist. If you have a field that you don't want to delete without testing first, use disable.

Field Does Not Print: When checked, the field's data will not print, but does not effect input into the field.

Required checks the field for data, prior to saving the form when Submit Data is clicked, A message will appear if the required field is left blank. A field can be required always or required by the actions of another field.

Input Disabled: Makes a field read-only by the actions of another field. The field can be cleared of data, or can be set to equal the value of another field.

What usefulness is the feature? When you have a case when a persons' name is to appear in another spot on the form, this works great. Since we still have a restriction that a duplicate field name is not allowed on the same page, you now have a way to get the duplicate value set in another field.

Group Name: This is only enabled for a CheckBox field type. This links CheckBoxes and makes them behave like radio buttons. For example, when a field in a group series is checked, all other CheckBoxes in the group will become unchecked.

Caption: This is only enabled for Buttons. This text will appear on the button as the caption. Font size is controlled by the Print Tab.

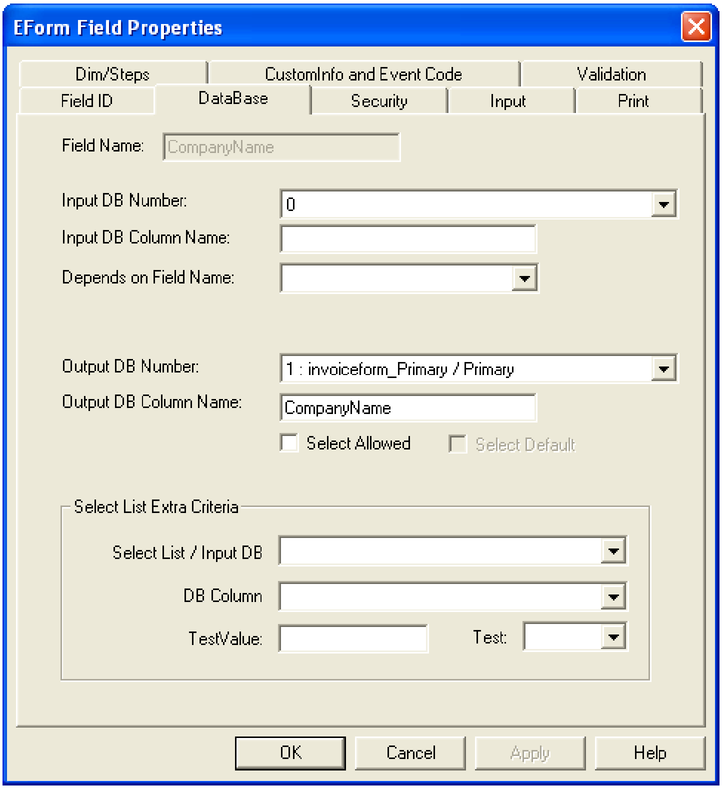
Initial Value: This is only enabled for Text fields. This text will appear in the field, but can be replaced, unless the field is Read-Only.

Help Text: This is a textual help message. When the cursor hovers over the field, the help text for that field will appear.

Message Text: When a field is required, by default the message will read: "You MUST enter (fieldname)". This allows you to customize the required message for the field. This text will replace the (fieldname) portion.

Database Tab

This tab shows the database table connections for the fill-field for both input and output. Changes can be made here, but are usually made within the main EForm Options Database dialogs.



Field Name: CompanyName

Input DB Number: 0

Input DB Column Name:

Depends on Field Name:

Output DB Number: 1 : invoiceform_Primary / Primary

Output DB Column Name: CompanyName

□ Select Allowed □ Select Default

Select List Extra Criteria

Select List / Input DB

DB Column

TestValue: Test:

Figure 5-2: Database Tab

Input DB Number: This will list the input databases which have been made available to your form through the EForm Input Database dialogs. Here you can choose one of them to be associated with this fill-field. The default number *0* indicates that there is no database table associated with this field. Associating an input table, indicates that this field READS data from the database table. The table field (column) which is read is stated in the *Input DB Field Name.*

Input DB Field Name: This is the column name of the Database Table Field Name that is being read into the field. (This will be the same as the form's corresponding field name if the table was created with the OFDP create_formname.cgi script.)

Depends on Trigger Name: This is for controlling the current field's information from the input table by making it dependent on another field. For example, when a customer is input into the "Customer" field, the address will populate. The Address Field is dependent and triggered upon the Customer being selected. Within Address's Database Tab the "Depends on Trigger Name" would be "Customer".

Output DB Number: This will list the output database table which has been made available to your form through the EForm Output Database dialogs. Here you can choose the table to be associated with this field. The default number *0* indicates that there is no table associated to the field. An output database table name indicates that this field WRITES to that database table.

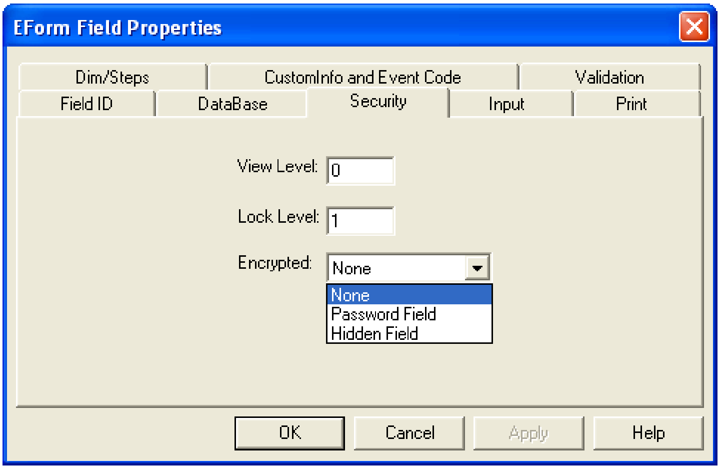
Output DB Field Name: This is the Database Table Field Name associated to the fill-field and to be used to store data from this field.

Select List Extra Criteria: This is for paring down the initial Input List. If the list contains more items than would be useful to display at once, you can identify another field to assist in testing the values needed for the list. Such as a checkbox for each alphabet or weather zones, etc. Identify the field and column for the test.

For more detailed information, see *Chapter 9: Associating Data Tables to the E-Form.*

Security Tab

Security Lock Levels can be applied to any field through this tab. Security lock levels are used when Signature Fields are applied to the form. All fields with a security lock level ***equal to or lower than*** the Signature field will be locked and un-editable, once a password has been applied. See *Chapter 18: Forms with Signature Fields* for more information.



View Level: 0

Lock Level: 1

Encrypted: None

Figure 5-3: Security Tab

Lock Level: Signature Fields use this value to control when the field is changeable or locked. This Lock Level value is 0 through 15, with 0 being the lowest. Use lock level 1 for all fields being locked when the first signature is applied, 2 for the second, and so on. Fields left with a 0 will be locked when the first signature is applied. Fields with a number higher than all signatures on the form will never be locked.

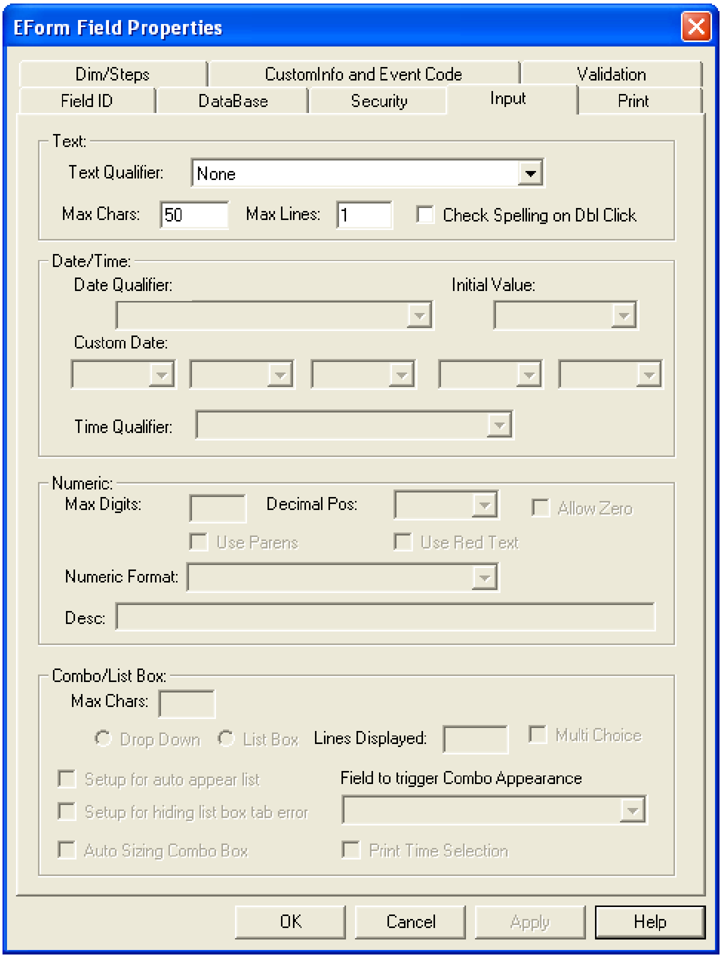
You can use the Group option to select all fields for a particular lock level and change them all at once.

View Level: This option allows a View Level to be assigned so that the field can be hidden through custom programming. This View Level value is 0 through 15.

Encrypted: A field can be encrypted in the HTML to not show keystrokes. This is only used on the Password field of a signature.

Input Tab

The Input Tab holds qualifiers for some of the field types and they become available only with their corresponding field type. These qualifiers place controls on the data being keyed into the fill-field, such as character limits.



EForm Field Properties

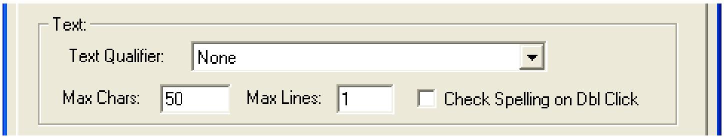
Text Qualifier: None

Max Chars: 50 Max Lines: 1 □ Check Spelling on Dbl Click

Figure 5-4: Input Tab

Text Qualifiers

Text Qualifiers are only active when Text is chosen as the field type. Through these qualifiers you can apply other controls to the text: Upper Case, Lower Case, or Numeric Only.



Text:

Text Qualifier: None

Max Chars: 50 Max Lines: 1 □ Check Spelling on Dbl Click

Figure 5-5: Text — Input Defaults

None: Places no restrictions on the user's input and toggles off the other selections.

Uppercase: Inputs all data as uppercase letters.

Lowercase: Inputs all data as lowercase letters.

Numeric: Allows input of 0 through 9, and a few characters, such as the decimal (.) and hyphen/minus (-) sign. Use a Text field with a Numeric qualifier when there is a numeric field that does not need calculations. For example, a phone number and social security number do not require any calculations, but do need to be numeric only with hyphens.

Maximum Characters: This controls the number of characters input into the entire field. The default is 50 characters, regardless of the width of the field. You will want to test your fields after you generate the HTML and make sure that the needed user input will fit. Only the characters that can fit in the physical space provided will print. However, all characters will display and be placed into the output database table.

Use 0 to have unlimited characters. Be aware that by giving the field an unlimited character count, the field will always be considered a memo field in a database.

Maximum Lines: This is how many lines of data are allowed in the field. A paragraph can hold more than 1 line of text as the user is typing, they may press [Enter] to go down to the next line. If the field is not deep enough to hold the amount of lines specified, a scroll bar will be displayed. When printing, it will only print the data that can be displayed, since there is only so much physical room on the paper.

To find the number of lines able to fit into a field, take the Field Height (Dim/Steps Tab) and divide it by the Leading (Print Tab). Round the number down and this will give you the maximum number of lines that can fit into the field.

Check Spelling: A spelling checker can be made available upon a double-click in a Text field.

Date/Time Qualifiers

The Date and Time Qualifiers become active only when their respective field types are chosen. The qualifier is the format data will be converted to when the user tabs out of the field.

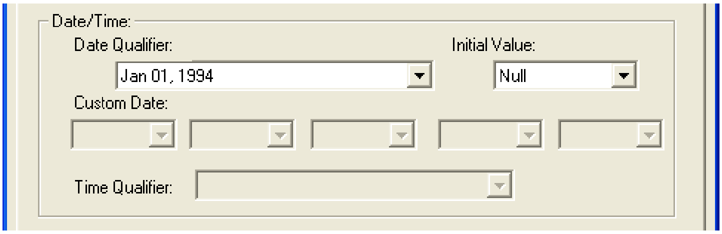
Date Qualifier: This will input the computer's date or convert the user's input into a variety of formats. Choose from the drop-down list. When choosing the Custom Date format, a selection is available for the day, month, year, and type of dividers. Standard choices include *Now* or *Null*. Use **Now** when you wish the computer's internal clock to input the date.

Date Format - Sample of Format
Jan 01, 1994 - Jan 31, 2012
January 01, 1994 - January 31, 2012
MM/DD/YYYY - 01/31/2012
01 January 1994 - 31 January 2012
General Date* - 01 January 2012
Long Date* - 01 Jan 2012
Medium Date* - 01 Jan 2012
Short Date* - 01-31-12
Custom Date - Jan 2012

***Follows Windows International Settings**

Figure 5-6: Date Formats

Time Qualifier: This is available for DateTime Field Type. It will input the computers time or convert the user's input into a variety of formats. Choose from the drop-down list.

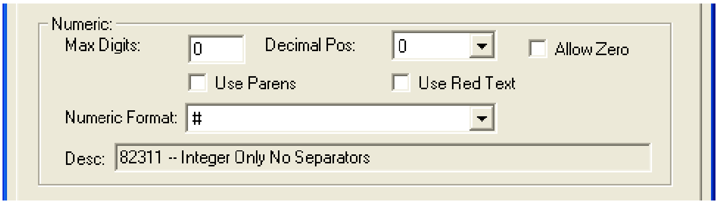


Date/Time:
Date Qualifier: Jan 01, 1994
Initial Value: Null
Custom Date:
Time Qualifier:

Figure 5-7: Date/Time Qualifiers

Numeric Format:

There are many Numeric Formats for E-Forms. The patterns are used to describe with or without decimals, and with or without the $ character.



Numeric:
Max Digits: 0
Decimal Pos: 0
□ Allow Zero
□ Use Parens
□ Use Red Text
Numeric Format: #
Desc: 82311 -- Integer Only No Separators

Figure 5-8: Numeric Qualifiers

Max Digits: is how many numbers are allowed including the decimal positions. Use 0 for unlimited.

Decimal Position: controls how many digits allowed after the decimal.

Numeric Format: is the format shown regardless of Maximum Digits needed.

1 Decimal	**Max Digits**	**Dec. Pos.**	**Numeric Format**
12345.6	6	1	#.00
$12345.6	6	1	$#.00
1,2345.6	6	1	#,##0.00
$1,2345.6	6	1	$#,##0.00
2 Decimal			
123456.78	8	2	#.00
123,456.78	8	2	#,##0.00
$123,456.78	8	2	$#,##0.00
3 Decimal			
12345.678	8	3	#.00
12,345.678	8	3	#,##0.00
$12,345.678	8	3	$#,##0.00
4 Decimal			
1234.5678	8	4	#.00
1,234.5678	8	4	#,##0.00
$1,234.5678	8	4	$#,##0.00
No Decimal			
12345	5	0	#
$12345	5	0	$#
12,345	5	0	#,###
$12,345	5	0	$#,###
No Decimal / Decimal Prints			
1,234.	4	0	#,###.
$1,234.	4	0	$#,###.

Figure 5-9: Examples of Numeric Formats

Allow Zeros: By default, if the value of a field is zero, the field is left blank. This will display a zero instead and save a zero in the data table.

Use Parentheses: If it is a negative value, replace the minus with parentheses.

Use Red Text: If it is a negative value, display in a Red color.

Suppressing Zeros for Total Fields

If using calculations and you do not want zeros in the totals, they must be suppressed. This requires JavaScript coding to suppress zeros for total fields.

HTML Calculations

```
// suppress zeros
   _ThisForm["tot10"].value = FPNFormatZS
   (_ThisForm["tot10"],
   _fldsFmt["tot1"]);
```

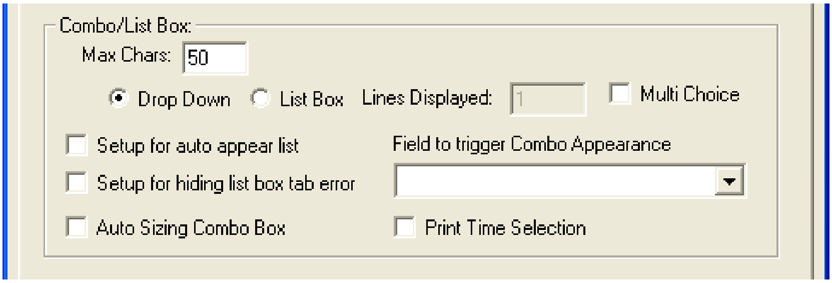
PDF Calculations

```
// suppress zeros
   if (_ThisForm["tot1"].value == 0)
   {_ThisForm["tot1"].value =  "";}
```

For examples of JavaScript calculations and suppressing of zeros in the totals, see the *Forms Library.*

ComboBox Qualifiers

The following qualifiers apply only to the ComboBox field types.



Combo/List Box:

Max Chars: 50

● Drop Down ○ List Box Lines Displayed: 1 □ Multi Choice

□ Setup for auto appear list

□ Setup for hiding list box tab error

Field to trigger Combo Appearance

□ Auto Sizing Combo Box

□ Print Time Selection

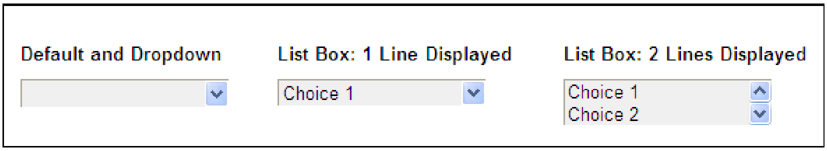
Figure 5-10: ComboBox Features on the Field Properties Input Tab

Max Chars: The ComboBox field creates a text field type (varchar) in a data table. The default is 50 characters, just like the TEXT fields.

The *Auto Sizing ComboBox* will adjust the width of the field in the HTML to fit the list. This is recommended if the list is coming from a data table.

Field to Trigger ComboBox Appearance: The ComboBox field can be hidden and made visible when focusing off of the selected field.

With the PDF Option, *Print Time Selection* causes the text column to print, not the values stored in the data table.



Default and Dropdown

List Box: 1 Line Displayed

Choice 1

List Box: 2 Lines Displayed

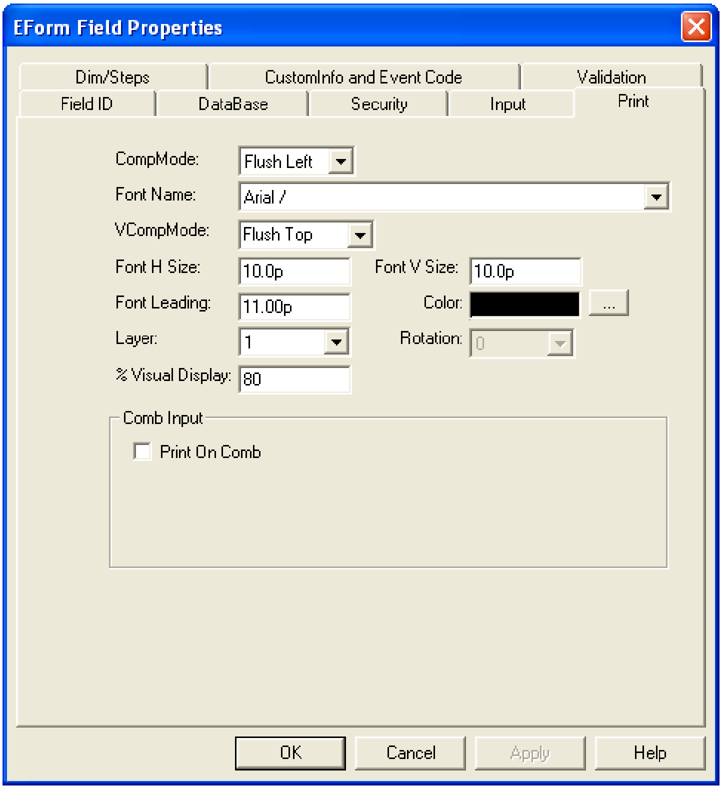
Choice 1
Choice 2

Figure 5-11: ComboBox has Different Looks and Behaviors

The default is a "DropDown" select list, that holds 50 characters in the data table field. It starts with a blank and you must select from the dropdown list to make a choice. "List Box" has options for number of lines displayed and how many the user can highlight. It does not have a blank and by default has the first item selected.

Print Tab

The print tab controls the look of the field's data for viewing and printing the electronic form. There are standard controls for font name, size, and leading, along with several others:



CompMode: Flush Left

Font Name: Arial /

VCompMode: Flush Top

Font H Size: 10.0p

Font V Size: 10.0p

Font Leading: 11.00p

Color: ■ ...

Layer: 1

Rotation: 0

% Visual Display: 80

Comb Input

□ Print On Comb

Figure 5-12: Print Tab

The **Comp Mode** (Composition Mode) controls the position of the text within the field. Choices include: Flush Left, Centered, Flush Right and Justified.

Flush Left aligns all left edges, *Flush Right* aligns right edges, *Centered* will center all lines over each other and *Justified* will adjust each line's spacing to align both the left and right edges of text.

Print on Comb: This looks at the Maximum characters allowed during input and prorates the field to achieve equal spacing between characters. Maximum Characters/Digits is required in order to achieve the effect. For example, if the field is to be a social security number, check *Print on Comb* and input *9* as the maximum number of characters. This option requires the CompMode to be "Centered".

The **Font** field shows what font is currently selected. Clicking on the [...] button next to the Font Name field will bring up the font dialog box. This box contains a list of all fonts accessible to OneForm Designer Plus Font Setup. Select the font and style needed and then click the Okay button.

The **VComp Mode** (Vertical Composition Mode) controls the position of the text vertically within the field. Choices include: Flush Top, Centered, and Flush Bottom.

Flush Top aligns the top of text to the top of the field. *Centered* will center all lines within the field height, and *Flush Bottom* will put the last line's baseline on the bottom of the field.

The **Font H Size** and **Font V Size** option control the size of your text. The HSize field controls the width of the text and the VSize field controls the height. These values can be entered in 1/10th point increments for either TrueType or PostScript font families. When they have two different values, you are requesting it to expand or condense the font. If the field is a Button, this controls the size of the caption text.

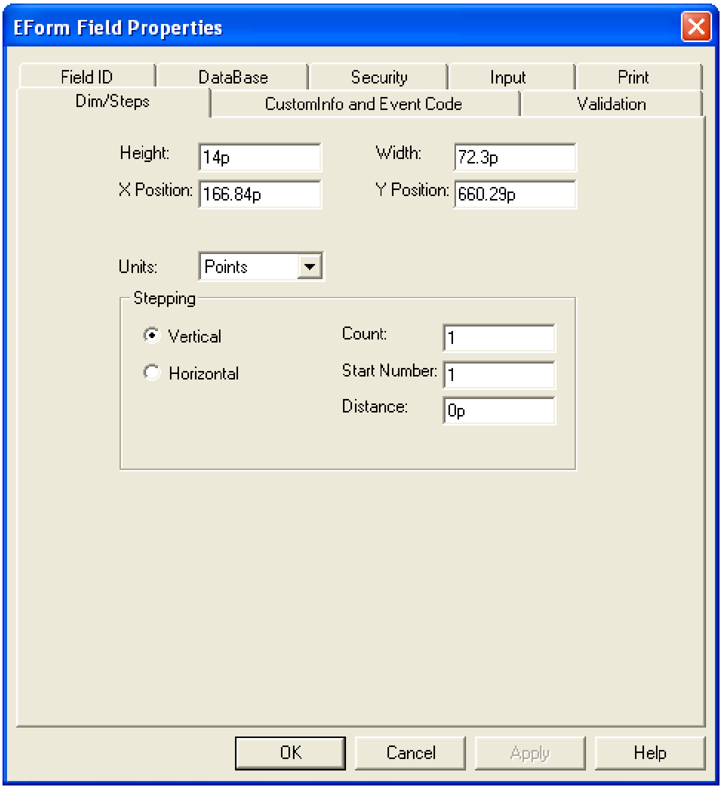
Leading controls the spacing from the baseline of the first text line to the baseline of the second text line. The values can be input in 1/10th point increments. The leading should be equal to or larger than the vertical size of the text to keep lines from oversetting.

Layer and Color controls the color of the user input text. It will pick up the color of the Drawing Properties, Layer Color. *See Chapter 8: Finishing Touches for more information.*

% **Visual Display** is for the size of the visual display of the text as you are entering it. Normally 80% of the printed point size gives a readable visual display when entering text on your monitor. If you are entering a size smaller than 9-point, you may want to change the visual display to be 100%. If the visual display is set too high for the point size and field height, the ascenders and descenders of the text may be cut off when printed.

Dim/Steps Tab

The size of the field is found here. The X and Y Positions are recording the lower-left corner of the field. You can also set the characteristics of Stepped Fields: number of steps, direction of steps, and the distance between them.



Height: 14p

Width: 72.3p

X Position: 166.84p

Y Position: 660.29p

Units: Points

Stepping

● Vertical

○ Horizontal

Count: 1

Start Number: 1

Distance: 0p

Figure 5-13: Dim/Steps Tab

Height, Width: This will report the height and width of the field. The size can be modified by editing these values.

X-Position, Y-Position: This is the X/Y coordinates of the component.

X is the horizontal position and Y is the vertical position of the lower-left corner of the field. The location of the field may be modified by editing these values.

Stepping

When electronic form fields are drawn they can be stepped allowing the operator to associate copies with one another. These copies are identical to the original field and are equally spaced horizontally or vertically. Since the steps are part of the original, they can only be accessed through the original.

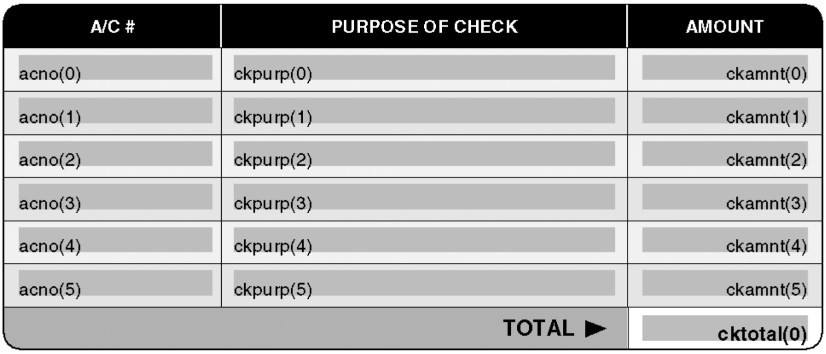
Through OneForm Designer Plus, the fields will be viewed as one. When the properties of one is altered, they are all altered. The fields are indexed and identified with a (#) attached to the field name. The original is always (0). For example if Field01 is stepped the names will be: Field01(0), Field01(1), Field01(2), etc.

Stepping: Select a direction for the copies: Horizontal or Vertical.

Count: This is the number of steps needed including the original.

Start Number: This is the starting number associated with the name. The fields are identified as Field01(0), Field01(1), Field01(2), etc. For example, normally a stepped field will begin with 1. If a stepped field were copied to the second page of a multiple page form, the copy could be named the same, but would begin with the next available number. If the field has 15 steps, the field on the second page would have a Start Number of 16.

Distance: This is the distance between each field. Distance is "left edge to left edge" for horizontal and "top to top" with vertical copy.



A/C #	PURPOSE OF CHECK	AMOUNT
acno(0)	ckpurp(0)	ckamnt(0)
acno(1)	ckpurp(1)	ckamnt(1)
acno(2)	ckpurp(2)	ckamnt(2)
acno(3)	ckpurp(3)	ckamnt(3)
acno(4)	ckpurp(4)	ckamnt(4)
acno(5)	ckpurp(5)	ckamnt(5)
	TOTAL ▶	cktotal(0)

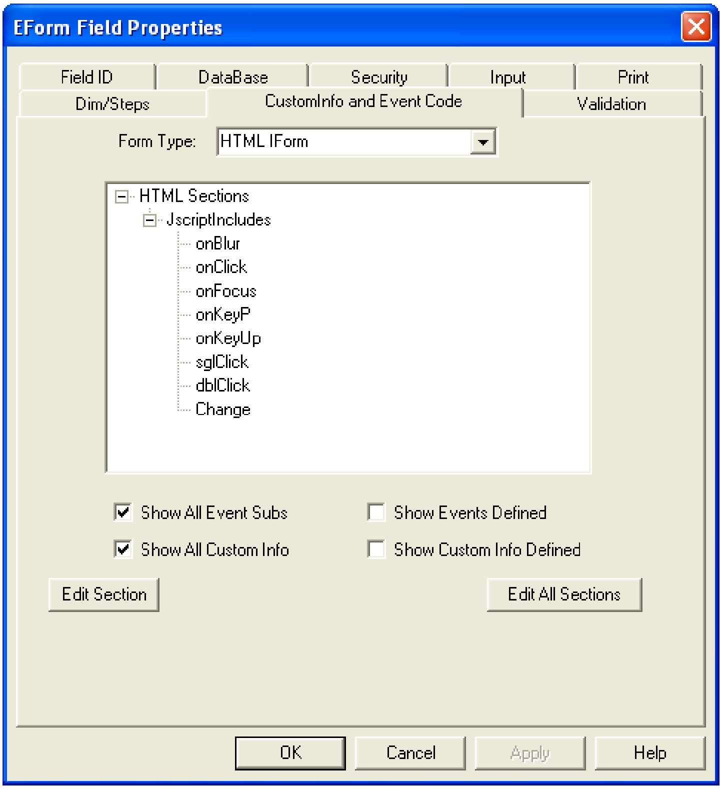
Figure 5-14: Stepping Works Well for Columns of Fields

CustomInfo Tab

Each field can have Custom Information to allow for control over its actions. This tab views the Custom Information. To make edits, select the appropriate section and click on the [Edit] button. This will open Notepad for editing the field's Custom Information.

BeginHelpInfo: This stores the help information for the field.

BeginInitVars: Put Custom VB command initial code here.



Form Type: HTML IForm

- HTML Sections
  - JscriptIncludes
    - onBlur
    - onClick
    - onFocus
    - onKeyP
    - onKeyUp
    - sglClick
    - dblClick
    - Change

☑ Show All Event Subs
☐ Show Events Defined
☑ Show All Custom Info
☐ Show Custom Info Defined

Figure 5-15: CustomInfo Tab

BeginJscriptIncludes: Put Custom HTML JavaScript commands here.

BeginCustomIncludes: Put Custom VB command event code here.

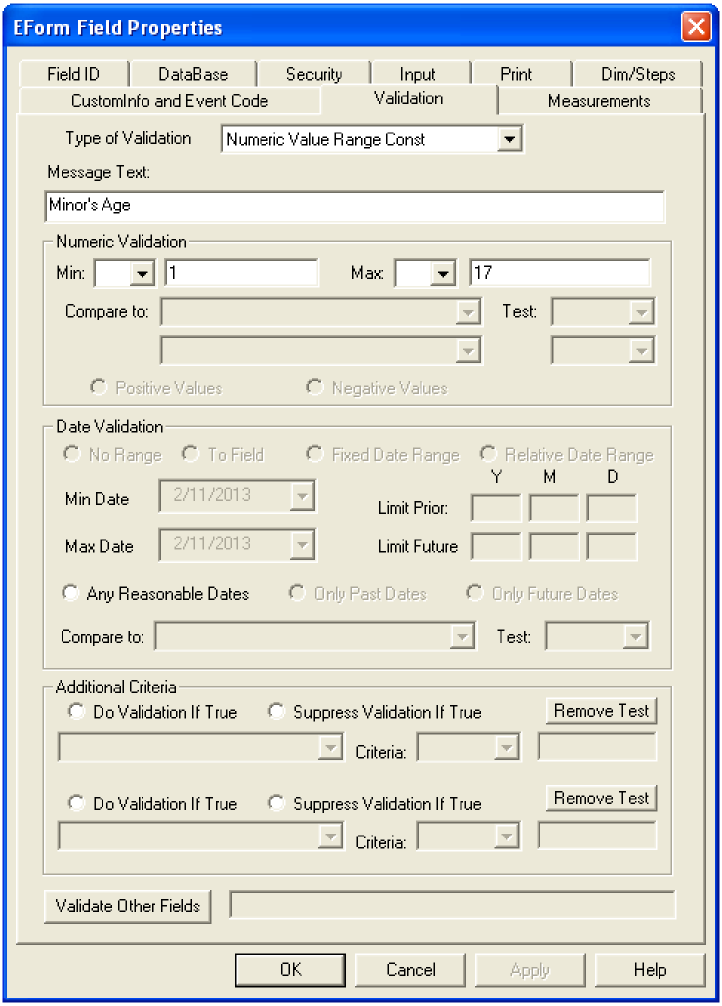
ComboBoxChoices: This is the stored choices on the Combo Options Tab.

Edit All Sections Button: Edit the custom information and open in Notepad.

Note: If you receive an error when trying to open Notepad, the problem lies in the location of the Notepad.exe in Windows. You will need to look at the software register and see if it matches NotePad's location.

Validation Tab

This tab allows you to set some basic validation tests for the user input. The tab is very similar to the Input Tab in that sections of the tab become available according to the field type.



Type of Validation: Numeric Value Range Const

Message Text: Minor's Age

Min: 1

Max: 17

Figure 5-16: Validation Tab

The example above is for a numeric field validation. It is comparing the user input to be within a constant range of numbers.

Validation needs "AutoFieldValidation=Y" in Form Preferences Genform Options to generate the validate.js files.

Modifying Field Properties

If you need to make changes to the properties of a field, you may do so through its Pop-up menu. Choose the Select tool and click on the field to modify and right-click to open the Pop-up menu. Choose either **Basic** or **Advanced E-Form Field Properties**. The Modify Field Properties dialog will appear. Make any corrections and [Okay]. If you do not see a choice of field properties, you are defaulting to Advanced E-Form Field properties always.

You may Group fields and alter the properties of many fields at one time. Only the properties that have been modified will be changed for all fields, leaving the other properties untouched.

Shortcut: Double-clicking on a field will bring up its Modify Properties Dialog.

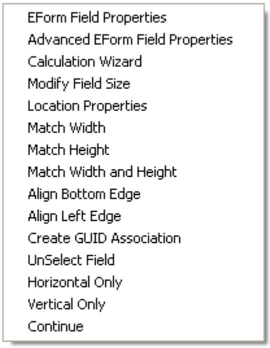


Figure 5-17: E-Form Field Pop-Up Menu

Basic EForm Field Properties: This will take you into that field's basic properties. At this time you can change the Field Name, Type, and some basic properties such as Qualifiers.

Advanced EForm Field Properties: This will take you into that field's advanced properties. These dialogs have all options available and can be made the default through the Options, Program Options menu. When this is set, only one properties selection will show up in the Pop-up menu.

Modify Field Size: Pick on the side of the field you want to modify and then pick its new location.

Location Properties: This will display the X and Y coordinates of the field and its size. Edits to location and size can be made here.

Match Width, Match Height, and **Match Height and Width** all work with Grouped fields only. Pick one field and then choose "Match".

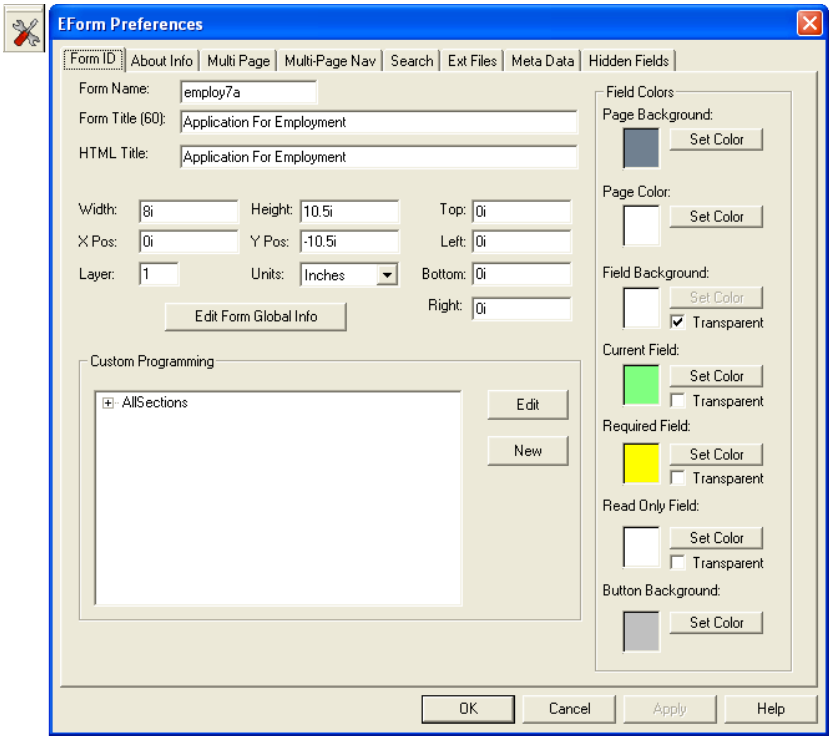
Align Bottom Edge and **Align Left Edge** also work with Grouped fields only. Pick one field and then choose "Align Edge".

CHAPTER 6

E-Form Preferences Tabs

The **EForm Preferences** can be found on the toolbar or in **EForm Options** under the Options menu. Setting the EForm Preferences is necessary in order to generate HTML code. Every form has its own set of preferences, with the most basic being the project's Form Name.

A shortcut to the EForm Preferences is **[Ctrl]-[F].**



Form Name: employ7a

Form Title (60): Application For Employment

HTML Title: Application For Employment

Width: 8i
Height: 10.5i
Top: 0i

X Pos: 0i
Y Pos: -10.5i
Left: 0i

Layer: 1
Units: Inches
Bottom: 0i

Right: 0i

Custom Programming

⊞ AllSections

Field Colors

Page Background:

Page Color:

Field Background:
☑ Transparent

Current Field:
□ Transparent

Required Field:
□ Transparent

Read Only Field:
□ Transparent

Button Background:

Figure 6-1: EForm Preferences Tool and the Form ID Tab

The E-Form Preferences are organized by tabs:

- **Form ID: –** Identifies the Form and its basic properties and holds any custom programming associated with the form. This tab also controls the form's color attributes for the Page Background Color and the Page itself, just by selecting a color. This tab also controls the form's field colors.
- **About Info: –** Designs the Splash Screen and tracks the Version for E-Forms and holds PDF E-Form document information.

- **Multi-Page and Multi-Page Navigation: –** Links additional pages (ELFs) together and controls the top navigation frame for HTML E-Forms.
- **Search: –** Searches Custom Info code in fields and eform preferences.
- **Meta Data: –** Meta Data is information about the form.
- **Hidden Fields: –** Lists hidden fields.

There are optional tabs which may appear such as Managed Forms or OneForm Manager. These require OFDP profile settings and optional software.

The following pages discuss the standard E-Form Preference's tabs.

Form ID Tab

When you click on the EForm Preferences tool, it always opens to this tab. *See Figure 6-1.* It starts out with fields for the Form Name and Title. The **Form Name** is the base name to be used when the electronic form is generated. The **Form Title** is displayed in the Window's Title bar when the electronic form is loaded. The Page Width and Page Height controls the form size displayed during use and when printing.

Form Name: For E-Forms, this is the project name and therefore must meet its standards: start with an alpha character, be between 3 and 25 characters, and include no punctuation. The default picks up the File Name. If it does not meet the naming standards, change it here.

It controls the basename of the HTML file and all corresponding CGI files when generating for an HTML E-Form.

For VB 6.0 E-Forms, the project name must meet its standards: start with an alpha character, be between 3 and 8 characters, and include no punctuation.

Form Title: This is the name that will appear on the Title bar when the form is in use. It will also appear on the Title bar of the Splash Screen. The default is to pick up the File Name from Drawing Properties.

Page Width and **Depth:** These are the dimensions of the page and populate from the Drawing Properties.

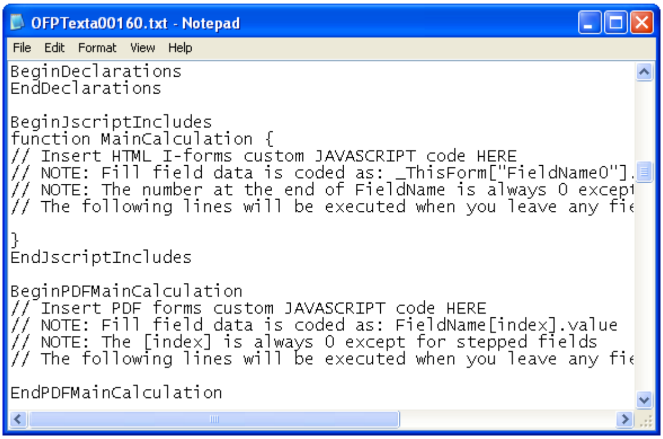
X and Y Position: This is the lower-left corner of your form for display and printing purposes.

Page Margins: Page margins are used to avoid scaling of the page when printing. Your form components should not spill out into the margins. For printing purposes the image size is determined by subtracting the margins from the overall page size. If your form is larger than your printer page parameters, then it uses the image size to determine how much to reduce the form to get a proper fit on the paper.

For this reason, we suggest putting in a ¼" margin all the way around, which will automatically make the Page Width and Page Height 8" x 10.5". It will still be an 8.5" x 11" form when printed and will be centered on the paper. You will also notice as you input the margins that the X and Y Positions will be adjusted to move the lower-left corner of the form to be inside the margin.

Custom Programming Section: The ELF file can hold Custom Programming. The **[Edit]** button will take you into Notepad allowing you to make edits to *All Programming Sections* at once.

Or, you can pick a specific section before selecting [Edit] to direct Notepad to the correct section.



```
BeginDeclarations
EndDeclarations

BeginJscriptIncludes
function MainCalculation {
// Insert HTML I-forms custom JAVASCRIPT code HERE
// NOTE: Fill field data is coded as: _ThisForm["FieldName0"].
// NOTE: The number at the end of FieldName is always 0 except
// The following lines will be executed when you leave any fie

}
EndJscriptIncludes

BeginPDFMainCalculation
// Insert PDF forms custom JAVASCRIPT code HERE
// NOTE: Fill field data is coded as: FieldName[index].value
// NOTE: The [index] is always 0 except for stepped fields
// The following lines will be executed when you leave any fie

EndPDFMainCalculation
```

Figure 6-2: NotePad will Open Displaying the EForm Preferences Code

The following categories within the Programming Section are commonly used:

GenformOptions: Displays information set through the [Edit Form Global Info] dialog.

HelpInfo: This displays information regarding the Splash screen and Help About section and populates from the EForm About Info Tab.

MultiPages: Displays information retrieved from the Multi-Page tab.

Database0: This information comes from setting the output database.

MailSelection: Displays information about the E-Mail options. Information is pulled from the [Edit Form Global Info] dialog.

BeginJscriptIncludes: Input HTML E-forms custom JavaScript here.

Within the BeginJscriptIncludes, you can add custom JavaScript for adding intelligence to the form. JavaScript placed inside of the Form Preferences, Programming is always seen first, before any code within an individual field.

Go to Options, Eform/Iform Options, Eform/IForm Preferences and pick the [Edit] button. This will open NotePad and take you to the programming section of the form. Be sure to place all functions between one set of "BeginJscriptIncludes" and "EndJscriptIncludes".

BeginPDFMainCalculation: Input PDF E-forms custom JavaScript here.

Be sure to place all PDF JavaScript calculations between one set of "BeginPDFMainCalculation" and "EndPDFMainCalculation". Examples can be found in the OFDP Forms Library.

Information on Custom JavaScript for HTML

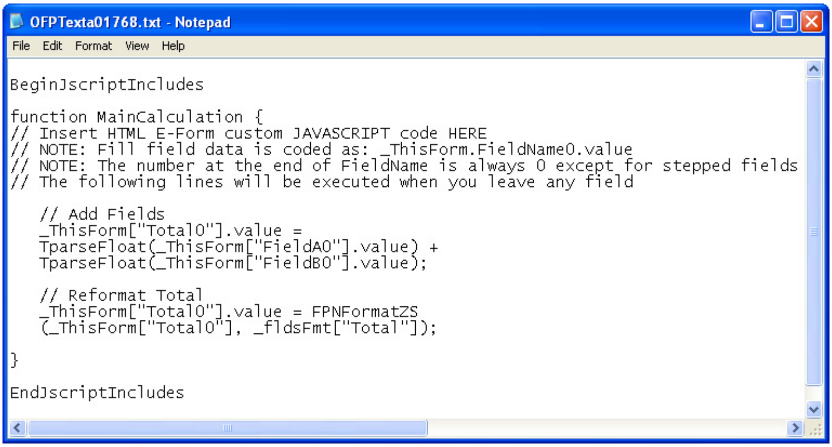
Simple calculations can be added to E-Forms through the Calculation Wizard. But, if you know JavaScript programming, you can place your own custom calculations into the form's programming section in EForm Preferences. When working with E-Forms, the calculations inserted should be written in JavaScript code. JavaScript is widely used with web browsers and there are many resources to learn JavaScript coding.

Custom calculations for E-Forms are added within **JscriptIncludes** in a function called **MainCalculation** as shown in *Figure 6-3*.

Example of Custom E-Form Calculations

Many examples can be found in the OFDP Forms Library. The following is an example of a simple custom calculation of two fields:

Total = FieldA + FieldB



```
BeginJscriptIncludes

function MainCalculation {
// Insert HTML E-Form custom JAVASCRIPT code HERE
// NOTE: Fill field data is coded as: _ThisForm.FieldName0.value
// NOTE: The number at the end of FieldName is always 0 except for stepped fields
// The following lines will be executed when you leave any field

   // Add Fields
   _ThisForm["Total0"].value =
   TparseFloat(_ThisForm["FieldA0"].value) +
   TparseFloat(_ThisForm["FieldB0"].value);

   // Reformat Total
   _ThisForm["Total0"].value = FPNFormatZS
   (_ThisForm["Total0"], _fldsFmt["Total"]);

}

EndJscriptIncludes
```

Figure 6-3: Custom Calculations with JavaScript

The calculation must begin with an open brace and end with an end brace. Each line of code must end with a semi-colon, as if putting a period at the end of a sentence to symbolize the end of a thought.

A heading must be placed before each component within a calculation. There are four acceptable headings: **document.forms[0], $frm, the formname** or **_ThisForm.**

An additional statement must be added in the string representing the type of number being referenced. All field types are referenced as "value", but must be further defined as either being an integer or floating decimal. **ParsInt** refers to an integer. An integer is any whole number either positive or negative. Decimal values will be discarded when using ParsInt. The **TparsFloat** is a floating decimal and will accept any number, including a decimal.

[Edit Form Global Info] Button

The [Edit Form Global Information] Button takes you into a dialog for assigning preferences to the form. *See Figure 6-3.*

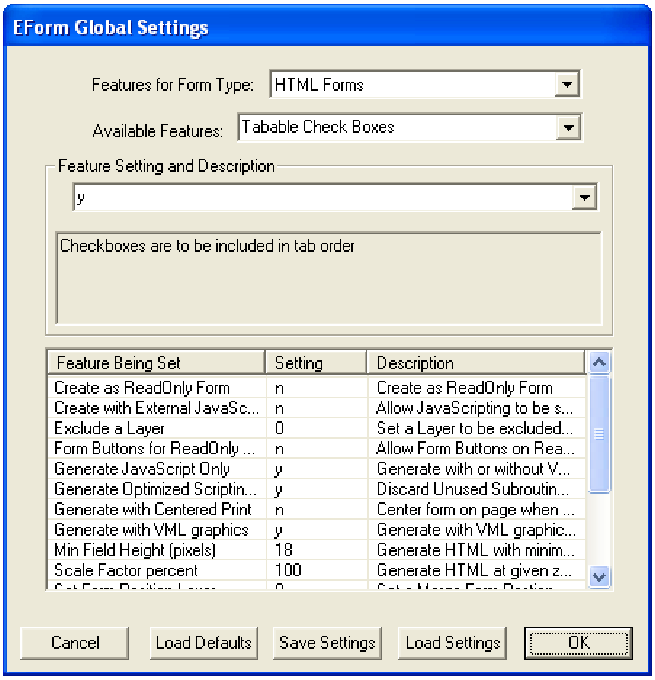
Our example shows how to enable tabbing through checkboxes for electronic forms. Select the feature **Tabable Checkbox** and choose a setting. **[OK]** the choice.

Form Type: The features available are catagorized by form type: *HTML Forms, PDF Forms, Stationery Forms* or *Visual Basic Forms.* Make a selection from the drop-down list. This will display the current settings for that form type for you to scroll through.

Available Features: If you wish to add or make a change to any settings, click on the feature from the drop-down list. This will display the setting choices in the *Feature Setting and Description* area.

Feature Settings: Make a selection from the list of settings available. Usually it is a Yes or No. At this time you should see the main window update with the choice. Select **[OK]** to save the settings.

[Save Settings] and **[Load Settings]** allow you to save and load preferences for all newly created ELF files. The **[Load Defaults]** will get the software preferences back to its original settings.



EForm Global Settings

Features for Form Type: HTML Forms

Available Features: Tabable Check Boxes

Feature Setting and Description

y

Checkboxes are to be included in tab order

Feature Being Set	Setting	Description
Create as ReadOnly Form	n	Create as ReadOnly Form
Create with External JavaSc...	n	Allow JavaScripting to be s...
Exclude a Layer	0	Set a Layer to be excluded...
Form Buttons for ReadOnly ...	n	Allow Form Buttons on Rea...
Generate JavaScript Only	y	Generate with or without V...
Generate Optimized Scriptin...	y	Discard Unused Subroutin...
Generate with Centered Print	n	Center form on page when ...
Generate with VML graphics	y	Generate with VML graphic...
Min Field Height (pixels)	18	Generate HTML with minim...
Scale Factor percent	100	Generate HTML at given z...
Set Form Position Layer	0	Set a Mouse Form Position

Figure 6-4: Form Global Settings

Global Features Available for HTML E-Forms

Auto Field Validation – Allow field validation.
AutoSave.fil Name – Specify name fully pathed. Leave blank to disable autosave.
AutoTab 1 Character Fields – Automatically tab out of 1 character fields. – Y/N
CGI Executables Path on JSP Server(Rel to ROOT) – For JSP Servers Only – Type in CGI Executables Path
CGI HTMLs Path on JSP Server(Rel to ROOT) – For JSP Servers Only – Type in CGI HTMLs Path
CGI Scripting for JSP Server – Define CGI Scripting for JSP Server – Y/N
Checkbox Print Character – Type in Alternate Printable Character for Checked fields.
Create as ReadOnly Form – Create a ReadOnly Form. – Y/N
Create with External JavaScripting – Allow JavaScripting in an external file and pathed by URL.
Disable Edits – Generate switchable code to disable field edits. – Y/N
Exclude a Layer – Set a Layer to be excluded from processing. – 0=No Layer Excluded, 1-32=Layer to be Excluded
Field Color - Current Field – Set a Color to represent the current field. – 80FF80 = (Green) RGB Color value for Current Field
Field Color - Field Background Color – Color to represent field backgrounds. – -1 = (Transparent) RGB Color value for All Field Backgrounds. Default 6% gray is RGB = F0F0F0.
Field Color - Read Only Fields – Set a Color to represent the read only fields. – -1 = (Transparent) RGB Color value for Read Only Fields. Default white is RGB = FFFFFF.
Field Color - Required Fields – Set a Color to represent the required fields. – FFFF00 = (Yellow) RGB Color value for Required Fields
Field Comments – Allow per Field Comments. – Y/N (Default N - Requires Custom Programming)
Form Buttons for ReadOnly Form – Allow Form Buttons on ReadOnly Form. – Y/N
Generate for No Signature Pad Attached – Generate HTML with no EPad warnings. – Y/N
Generate JavaScript Only – Generate with or without VBScript. – N=JavaScript with VBScript, Y=JavaScript Only
Generate Optimized Scripting Code – Discard Unused Subroutines from JavaScript Code. – Y/N
Generate with Centered HTML – Center HTML form in browser. – Y/N
Generate Plain HTML – Generate HTML without fill fields. – Y/N
Generate with Centered HTML – Center HTML form in Browser. (Default Y) – Y/N
Generate with Centered Print – Center form on page when printing. (Default upper-left corner.)
Generate with VML Graphics – Generate with VML graphics for IE Browsers. – Y/N
Ignore Custom JavaScript – Generate HTML without Custom JavaScript.
Local File Save Path – Default Save Path for Local File functions. – Type in full path.
Macintosh DB Init Compatible – Generate Macintosh Database Compat HTML.
Manual Initialization of Select Lists – Do NOT Invoke the InitSele Functions OnLoad, Scripting will Invoke at Appropriate time.
Min Field Height (pixels) – Generate HTML with minimum Field Height. – Between 10 and 50.
Multi-Line Counter Display – Display Character Counter During Input in Multi-Line Text Fields.
Netscape Compatible – Generate Netscape V6 Compatible HTML. – Y/N
Page Background Color – RGB Color Value for Form Background. – Type in RGB value.
Page Color – RGB Color Value for Form Background. – FFFFFF=(White)RGB Color value for Form Itself.
Print ComboBox Icon – Print the ComboBox Arrow Icon and Border.
Print ComboBox Text – Print the ComboBox Text, else the Value.
Print Image on Empty Fields – Specify an Image to print on empty fields except: CheckBox, Button, Graphic, VarText, Layout.
Print Local File Message – Print the message To locally save or open files.
Raster Target Directory – Specify Path For Inline Rasters to be Found At.
Remove Field Name Parens in HTML – Remove (0) from field names in HTML. – Y/N
Scale Factor Percentage – Generate HTML at given zoom percentage. – Between 50 and 250.
SendAllParens – Add (0) to field names when building the send_all function. – Y/N
Set Form Position Layer – Set a Merge Form Data Positions Layer. – 0=No Layer Excluded, 1-32=Layer to be Excluded
Show Field Border – Show border on all fields (display and print). – Y/N
Single Database Value for Groups – Generate a Single Database Output Column per Group – Y/N
Stationery Layout Gen Hidden VARTEXT – Generate Code for Hidden VARTEXT Fields.
Tabable Check Boxes – Checkboxes are to be included in tab order. – Y/N
Table Field Flags – Display Navigation Flags for fields in large tables. – Y/N (Default N - Requires Custom Programming)
Transparent Field Background – Make a regular field background color transparent. – Y/N
Unchecked Checkbox Print Character – Select an Alternate Printable Character for Unchecked checkboxes. – Type in Alternate Character

URL to External JavaScript – URL path to External JavaScript files. – Type in full path.
URL to JavaScript CkSpell – URL path to JavaScript Check Spell files.
Use HTML Fonts – Downloadable Typefaces: N=Disabled, PFR=Bitstream Fonts, WEFT= Microsoft Fonts
Use HTML Web Font Path – Absolute or Relative Path to downloadable fonts. – Type in full path.
Use Navigation Bar for Single Pages – Force use of Navigation Bar on single page forms. – Y/N
Validate On Print – Force Validation of Required fields on Print – Y/N
Validate on Save – Force Validation of Required fields on Save – Y/N
XML Access – Generate functions to access XML Document processing. – Y/N

Global Features Available for PDF E-Forms

Create PDF Fields as ReadOnly – Create a Read Only Form – Y/N
PDF Image on Empty Fields – Specify an Image to place on empty fields except: CheckBox, Button, Graphic, VarText, Layout
Stationery Display Layout Code – Generate Code for Display of Field Data in Standalone Layout – Y/N
Stationery Layout Composition Code – Generate Code for Composition of Data in Standalone Layout – Y/N
Stationery Layout Init Code – Generate Initialization code for Standalone Layout – Y/N
Stationery Previous Layout Code – Generate Code for Previous Data in Standalone Layout – Y/N
Tabable Check Boxes – Checkboxes are to be included in tab order. – Y/N
Use_ThisForm Notation – Generate PDF with _ThisForm Notation – Y/N
Use ADBC – Generate PDF with Acrobat ADBC database functions – Y/N

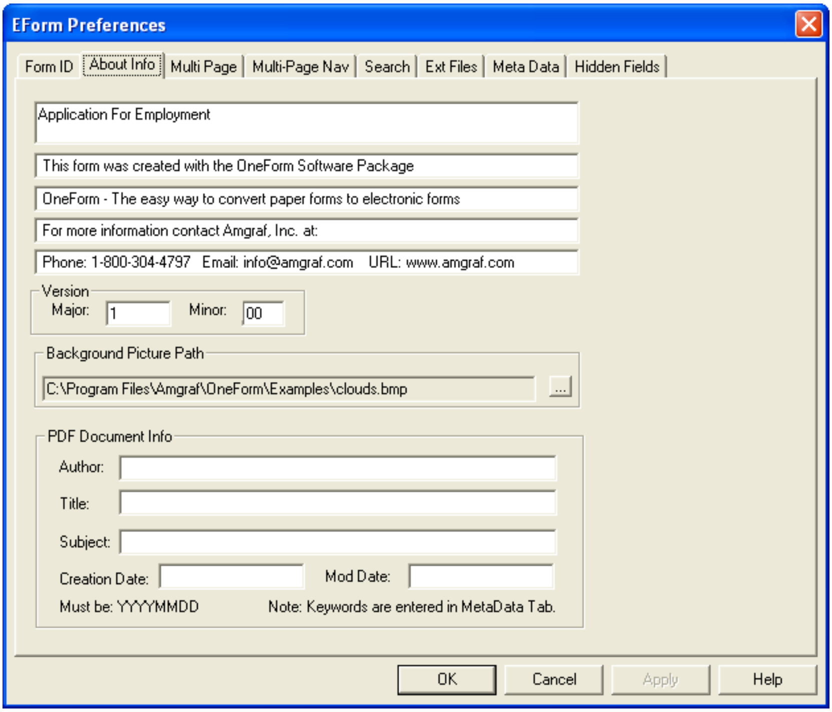
Global Features Available for Stationery Forms

Refocus to Imprint Fields – Focus to last imprint fields after click – Y/N
Variable Depth Imprint Fields – Allow imprint fields to contract/expand – Y/N

Global Features Available for Visual Basic Electronic Forms

Add GetDirectory Frm – Add 'Get Directory Dialog' to VB project. – Y/N
Add GetPassword Frm – Add 'Get Password Dialog' for signatures to VB project. – Y/N
Add SendFile Frm – Add 'Send File Dialog' to VB project. – Y/N
Add Sign and Send Frm – Add 'Sign and Send Dialog' to VB project. – Y/N
Add Visual Speller – Add Visual Speller to VB project. – Y/N
Allow Delete Record – Allow Database record deleting. – Y/N
Allow Rich Text – Allow Rich Text Boxes for Style Changes. – Y/N
AutoSave.fil Name – Specify an fully pathed AutoSave.fil name. Leave blank to disable autosave.
Confirm EXE Exit – Ask to Confirm that it is OK to Exit. – Y/N
Confirm FIL File Open – Confirm if invalid FIL file open. – Y/N
Default Printer – Form has a default printer. – Y/N
Display Internet Submit URL – Distributed E-Forms Allow Submit URL to be displayed.
E-Mail Type – Supported E-Mail Types – 0=No Mail, 1=MS Mail, 2=VIM Mail
Enable Maximize Button – Enable Maximize Button on Form – Y/N
Exclude a Layer – Set a Layer to be excluded from processing. – 0=No Layer Excluded, 1-32=Layer to be Excluded
Form Classification – Assign Form Classification or Type. – Input Form Classification Name
Form Display Width – Make E-Form Display Larger or Smaller – 8i
Go to Next Page – Control automatic Go To Next Page for multi-page forms. – Y/N
Internet Submit Server Type – Distributed E-Forms Internet server type. – NCSA or Website
Internet Submit URL – Distributed E-Forms URL for Submit Button – Type in URL
Set Form Position Layer – Set a Merge Form Position Layer. – 0=No Layer Merged, 1-32=Layer to Hold Data Positions
Show Field Border – Show Borders on all fields (display only). – Y/N
Show Splash Screen – Show Splash Screen on startup. – Y/N
Signature with Validation – Signatures use MD5 validation. (Default No) – Y/N
Tabable Check Boxes – Checkboxes are to be included in tab order. – Y/N
Update Catalog INI – Form controls Catalog INI file update. – Y/N
Update Form INI – Form controls INI file update.
Windows Display Bold Font – Use bold style font for Field Text. – Y/N
Windows Display Font – Font to be used for Field Text. – Choose from font list.
Windows Display Italic Font – Use Italic style font for Field Text. – Y/N

About Info Tab



EForm Preferences

Application For Employment

This form was created with the OneForm Software Package

OneForm - The easy way to convert paper forms to electronic forms

For more information contact Amgraf, Inc. at:

Phone: 1-800-304-4797 Email: info@amgraf.com URL: www.amgraf.com

Version

Major: 1 Minor: 00

Background Picture Path

C:\Program Files\Amgraf\OneForm\Examples\clouds.bmp

PDF Document Info

Author:

Title:

Subject:

Creation Date: Mod Date:

Must be: YYYYMMDD Note: Keywords are entered in MetaData Tab.

Figure 6-5: About Info Tab

The **About Info Tab** is used for two types of forms. The top half is VB E-Forms only, the bottom is for PDF E-Forms.

For PDF E-Forms:

PDF Document Info: This area of the tab when filled will be displayed in the PDF file’s Document Properties. "Keywords" is found under Meta Data Tab.

For VB E-Forms:

Form Information: There are five fields to put the Form Title and Company Information such as address, web site, and phone number. This information will appear on a Splash Screen when opening an E-Form. It will also appear when the user accesses the About under the Help menu.

Version: The Version Number can be used for managing With E-Forms, it is used to link ASCII files to their originating forms. When a FIL file is opened, it looks for the correct executable and version. For this reason, it is important to modify the version number whenever field changes have been made to the form. Also, when using the E-Forms Access Manager, updating only takes place when a new or higher version is encountered.

Background Picture Path: This allows you to have a background image of your choice on your Splash Screen. The default is an image of clouds. You may choose to keep the clouds background or to use one of your own. A company logo is a nice addition here. Add your own logo by clicking on the [...] box and searching for your own image.

Multi Page Tab

Working with Multi-Page Forms

When you are working on a multi-page form, it is recommended to work with each page as a stand-alone ELF first. Generate each ELF individually and test for look of design, tab order, any field custom code behavior, etc.

When all pages generate and have been tested, open the first page and go to this Multi Page tab and add all of the other pages to the first. Save your file and from that point forward, you will be opening just the first page. You will notice that the paging icons allow you to move through the ELFs. You can make changes to any of them. All of them are saved when you click on the Save tool.

Again, remember to now open the first page for database connections, getting server ready, and any time you generate either the HTML or Save As a PDF Form.

When you go through the steps of output database connections, it will see all fields in the project. Go through the server setup as you normally would, making sure you are on page 1.

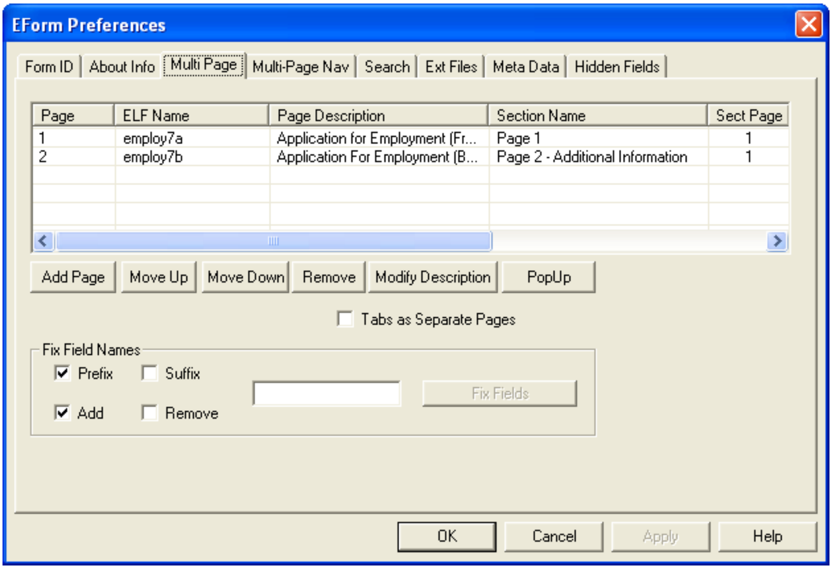
You can still open each one up individually and make any fielding changes, tab ordering, etc and save them. But you MUST open page 1 to edit your database connections, generate HTML, etc.

The Employee/employ07 folder in the Forms Library holds a multiple page example.

Using the Multi-Page Tab

The Multiple Page tab is looking at different ELF files in the same directory. It uses the order that each file is added to determine the page sequence. Use the **Move Up** or **Move Down** to arrange the order.

Add Page: This brings up an Open dialog. Search for the ELFs that will make up the subsequent pages of your form.



Page	ELF Name	Page Description	Section Name	Sect Page
1	employ7a	Application for Employment (Fr...	Page 1	1
2	employ7b	Application For Employment (B...	Page 2 - Additional Information	1

□ Tabs as Separate Pages

Fix Field Names

☑ Prefix □ Suffix

☑ Add □ Remove

Figure 6-6: Multi Page Tab

Remove Page: This removes a page from the Multi Page list. Click on the form to remove and then [Remove Page].

Modify Description: This modifies the Form Description and Page/Section Name (labels on the Tabs design). Highlight the page to modify, then click [Modify Description] to edit. [Okay] the changes.

PopUp: Identifying a page as a "PopUp" places a "Y" in the PopUp column. This page is accessible through a button action causing a popup window to show the page, not the navigation bar. Popups require javascript code, see the example in the Forms Library: Workorder/wrkord01.

Fix Field Names: It allows you to add or remove a prefix/suffix name to a group of fields. It will change the field name, tabbing name, and database reference name. This is nice if your pages are identical and yet you need them to have unique fields. You can apply a prefix of "pg1, pg2" etc.

Miscellaneous Notes

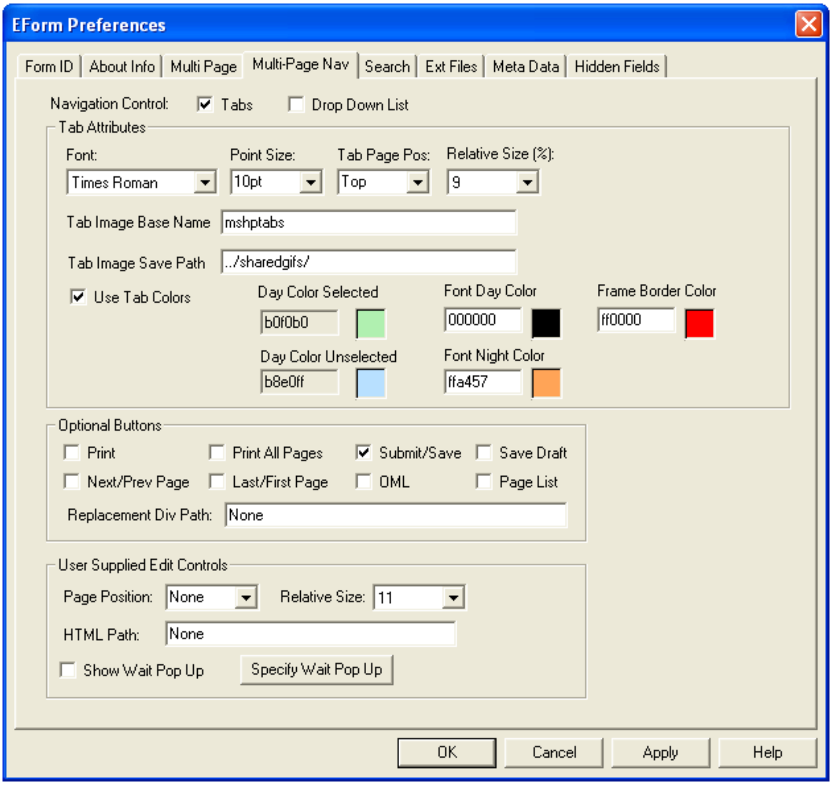
Forms can share data between the separate ELFs. There are two requirements: Field Names must match exactly and the fields must be the same Field Type.

All pages must have at least one field, you can make it hidden or read-only if the field is not really used.

If your multipage form has custom calculations within the EForm Preferences, they must be in Page 1.

Turn off the "Go to Next Page" setting under [Edit Form Global Info], to stop automatic tabbing through pages. This location also holds the option "Use Navigational Bar for Single Pages" if needed.

Multi-Page Navigational Tab



Navigation Control: ☑ Tabs ☐ Drop Down List

Tab Attributes

Font: Times Roman

Point Size: 10pt

Tab Page Pos: Top

Relative Size (%): 9

Tab Image Base Name: mshptabs

Tab Image Save Path: ../sharedgifs/

☑ Use Tab Colors

Day Color Selected: b0f0b0 🟩

Font Day Color: 000000 ⬛

Frame Border Color: ff0000 🟥

Day Color Unselected: b8e0ff 🟦

Font Night Color: ffa457 🟧

Optional Buttons

☐ Print ☐ Print All Pages ☑ Submit/Save ☐ Save Draft

☐ Next/Prev Page ☐ Last/First Page ☐ OML ☐ Page List

Replacement Div Path: None

User Supplied Edit Controls

Page Position: None

Relative Size: 11

HTML Path: None

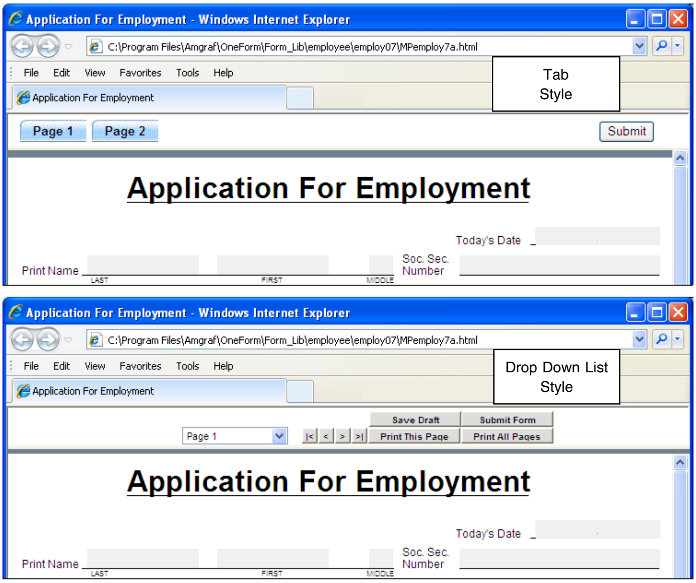
☐ Show Wait Pop Up

Figure 6-7: Multi-Page Navigational Tab

This tab allows you to control the look and functionality of the HTML multi-page control panel. You can set the font attributes, control look, and choose a variety of buttons for the control panel. You can even replace it with your own "Div" or "HTML" frame.

Multiple page HTML E-Forms can be generated with two different looking page control panels: Tabs or Drop Down List.

The default panel is the "Drop Down List" style. Changing your form to the "Tabs" style is easy. Check "Tabs", choose any of the optional buttons (such as Submit), and click the [Apply] button.


Application For Employment - Windows Internet Explorer
C:\Program Files\Amgraf\OneForm\Form_Lib\employee\employ07\MPemploy7a.html
File Edit View Favorites Tools Help
Application For Employment
Page 1
Page 2
Submit
Application For Employment
Today's Date
Print Name
LAST
FIRST
MIDDLE
Soc. Sec. Number
Save Draft
Submit Form
Page 1
|< < > >|
Print This Page
Print All Pages

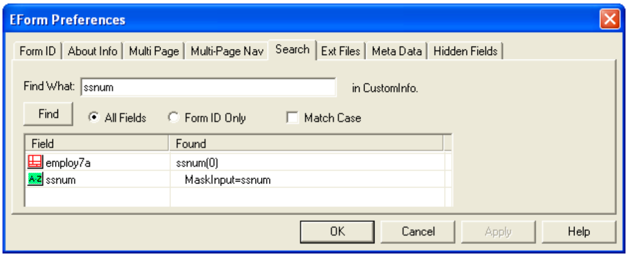
Tab Style

Drop Down List Style

Figure 6-8: Multi-Page Navigational Styles

Search Tab

This allows you to search through the contents of the programming section and custom information for strings of text. For example, you might use this to search for a field name.


EForm Preferences
Form ID | About Info | Multi Page | Multi-Page Nav | Search | Ext Files | Meta Data | Hidden Fields
Find
OK
Cancel
Apply
Help

Find What: ssnum in CustomInfo.

● All Fields ○ Form ID Only □ Match Case

Field	Found
employ7a	ssnum(0)
ssnum	MaskInput=ssnum

Figure 6-9: Multi-Page Navigational Styles

Meta Data Tab

The only area applicable here is **Keywords.** They get transferred into the PDF document. The rest of the tab is reserved for use with OML E-Forms.

Hidden Fields Tab

This tab displays a list of the hidden fields on your form.

CHAPTER 7

Adding the Finishing Touches for 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**
- **Adding Text Hyperlinks**
- **Adding Help Messages**
- **Modify Required Message**
- **Adding Input Masks**
- **Numeric Formats**
- **Custom Calculations**
- **Change Field Colors/Border**

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.

If the tab sequence has not been defined, an error log message will appear when the HTML is generated, and the user will not be able to move through the fields correctly. As a minimum, *Sort Tabs in Reading Order*.

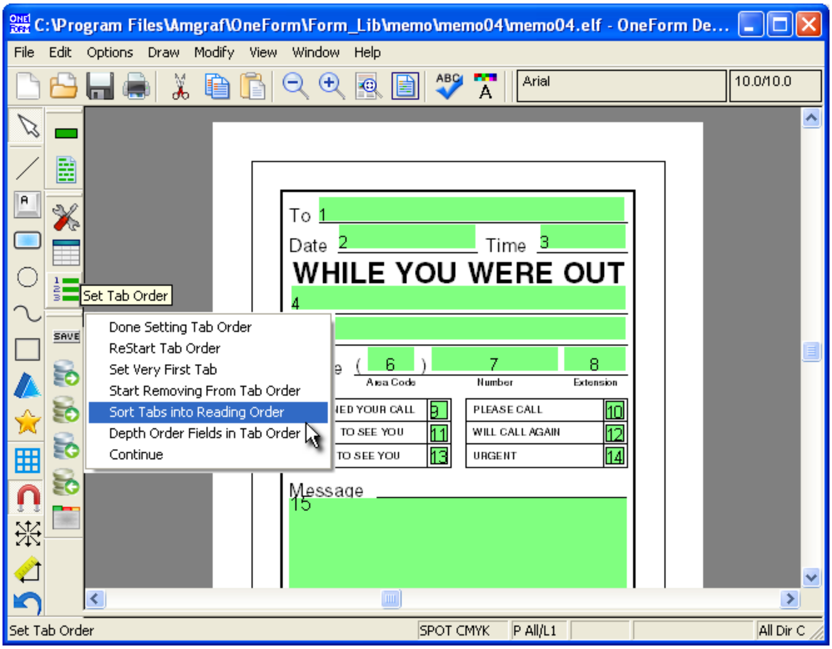


Figure 7-1: EForm Set Tab Order

Sort Tabs in Reading Order

Step 1: To set the tab order, go to the **Options** menu, **EForm Options** and **EForm Set Tab Order.** Notice that numbers will appear on the fields. These numbers represent the default tab order. They appear displaying the print properties of 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 Form Preferences.

Step 4: You can always see the tab order by going to the **View** menu and toggle the **EForm Fields, Show Tab Numbers.** *Read Only* fields cannot be tabbed into and will have a 0 number.

Resetting a Tab Sequence

Step 1: Go to the **Options** menu, **EForm Options** and **EForm Set Tab Order.** 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.

Tabable Check Boxes

CheckBoxes can be made tabable by going through the following steps:

Step 1: Go to the **Options, EForm Options, EForm Preferences Dialog.**

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

Step 3: Select **HTML 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.

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

Step 7: Now when you go to **EForm Set Tab Order,** the options will recognize checkboxes as being tabable.

AutoTab 1 Character Fields

Fields that hold only 1 character can be made auto tabable after typing the character, by going through the following steps:

Step 1: Go to the **Options, EForm Options, EForm Preferences Dialog.**

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

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

Step 4: Select **AutoTab 1 Character Fields** 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: Then go to **EForm Set Tab Order,** and set the tab order of the fields.

Step 8: The option "Depth Order Fields in Tab Order" must be chosen before saving the tab order.

Tab Order Pop-up Menu

The following options are found in the Tab Order menu:

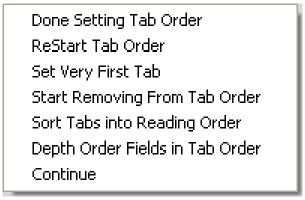

Done Setting Tab Order
ReStart Tab Order
Set Very First Tab
Start Removing From Tab Order
Sort Tabs into Reading Order
Depth Order Fields in Tab Order
Continue

Figure 7-2: Tab Order 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 it 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.

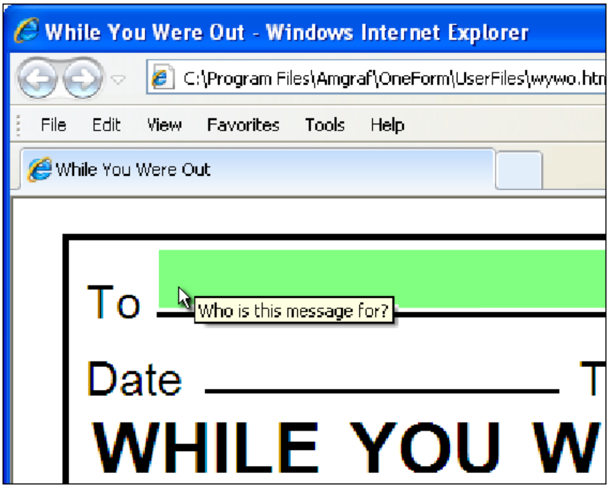
Adding Text HyperLinks

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^
```

Adding Help Messages to Fields

The field properties Field ID Tab can hold a textual Help message. When the cursor hovers over a field, the Help Text for that field will appear.


While You Were Out - Windows Internet Explorer

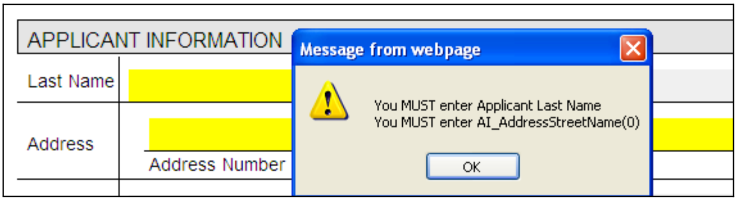
C:\Program Files\Amgraf\OneForm\UserFiles\wywo.htm

Who is this message for?

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

Modify Required Field’s Message Text

When a field is required and empty on Submit, by default the message will read: "You MUST enter (fieldname)". Message Text on the Field ID Tab allows you to customize the required message for the field. This text will replace the (fieldname) portion.



You MUST enter Applicant Last Name
You MUST enter AI_AddressStreetName(0)

Figure 7-4: Required Field Alerts With and Without Message Text

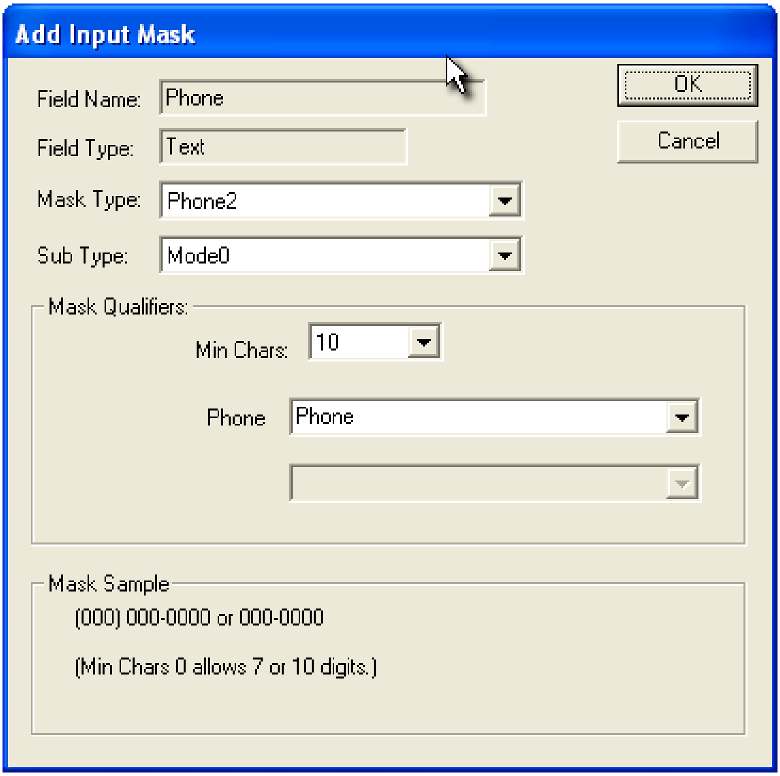
Message Text will also appear on the Validation Tab and requires "AutoFieldValidation=Y" in Form Preferences Genform Options to generate the validate.js files.

Adding Input Masks

A formatting mask can be added to 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

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. Normally the Input and Output Field is your current field.



Add Input Mask

Field Name: Phone

Field Type: Text

Mask Type: Phone2

Sub Type: Mode0

Mask Qualifiers:

Min Chars: 10

Phone: Phone

Mask Sample

(000) 000-0000 or 000-0000

(Min Chars 0 allows 7 or 10 digits.)

Figure 7-5: 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.

Common Text Masks

Bank (ABA Routing Number): This adds a check digit to an 8-digit routing number. (NNNNNNNNN)

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 hypen in the Federal ID structure (NN-NNNNNNN).

MECCA Units: Converts input into hundreths 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.

Numeric Formats

There are many Numeric Formats for Numeric Field Types found on the Input Tab. The patterns are used to describe with or without decimals, and with or without the $ character.

Max Digits: is how many numbers are allowed including the decimal positions. Use 0 for unlimited.

Decimal Position: controls how many digits allowed after the decimal.

Numeric Format: is the format shown regardless of Maximum Digits needed.

Allow Zeros: By default, if the value of a field is zero, the field is left blank. This will display a zero instead and save a zero in the data table.

Use Parentheses: If it is a negative value, replace the minus with parentheses.

Use Red Text: If it is a negative value, display in a Red color.

1 Decimal	Max Digits	Dec. Pos.	Numeric Format
12345.6	6	1	#.00
$12345.6	6	1	$#.00
1,2345.6	6	1	#,##0.00
$1,2345.6	6	1	$#,##0.00
2 Decimal			
123456.78	8	2	#.00
123,456.78	8	2	#,##0.00
$123,456.78	8	2	$#,##0.00
3 Decimal			
12345.678	8	3	#.00
12,345.678	8	3	#,##0.00
$12,345.678	8	3	$#,##0.00
4 Decimal			
1234.5678	8	4	#.00
1,234.5678	8	4	#,##0.00
$1,234.5678	8	4	$#,##0.00
No Decimal			
12345	5	0	#
$12345	5	0	$#
12,345	5	0	#,###
$12,345	5	0	$#,###
No Decimal / Decimal Prints			
1,234.	4	0	#,###.
$1,234.	4	0	$#,###.

Figure 7-6: Examples of Numeric Formats

Suppressing Zeros for Total Fields

If using calculations and you do not want zeros in the totals, they must be suppressed. This requires JavaScript coding to suppress zeros for total fields.

HTML Calculations

```
// suppress zeros
   _ThisForm["tot10"].value = FPNFormatZS
   (_ThisForm["tot10"],
   _fldsFmt["tot1"]);
```

PDF Calculations

```
// suppress zeros
   if (_ThisForm["tot1"].value == 0)
   {_ThisForm["tot1"].value =  "";}
```

For examples of JavaScript calculations and suppressing of zeros in the totals, see the *Forms Library.*

What if you Need Custom Calculations

Simple math calculations can be added through the Calculation Wizard. But, if you know JavaScript programming, you can add custom features and calculations to your form by including JavaScript. It is widely used with web browsers and there are many resources to learn JavaScript coding.

All custom programming is added to your form through one of two programming sections. Where you place the programming depends on its purpose.

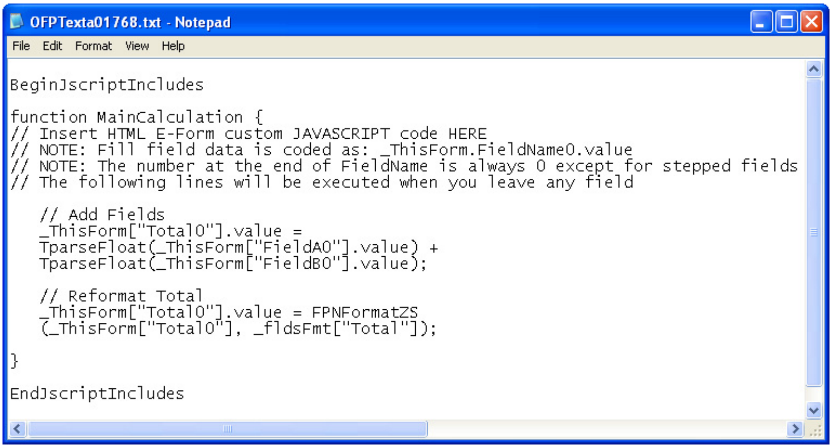
If the programming effects only one field and nothing else on the form, the custom programming can be placed within that Field's Properties. The Custom Info Tab is designed for this purpose. Clicking the [Edit] button will open Notepad allowing you to input your code. Save the work and close Notepad.

Following is an example of a function that can be used in the field properties CustomInfo Tab of individual fields.

```
BeginJscriptIncludes
function onBlur {
    $frm.secondfield0.value = $frm.firstfield0.value;
}
EndJscriptIncludes
```

Some of the functions available are onBlur, onKeyPress, onFocus, onLostF, onClick, onDblClick, and onChange.

If the programming effects many fields, add the JavaScript into the form's programming section in EForm Preferences. It is still within **JscriptIncludes,** but in a function called **MainCalculation** as shown below:



```
BeginJscriptIncludes

function MainCalculation {
// Insert HTML E-Form custom JAVASCRIPT code HERE
// NOTE: Fill field data is coded as: _ThisForm.FieldName0.value
// NOTE: The number at the end of FieldName is always 0 except for stepped fields
// The following lines will be executed when you leave any field

   // Add Fields
   _ThisForm["Total0"].value =
   TparseFloat(_ThisForm["FieldA0"].value) +
   TparseFloat(_ThisForm["FieldB0"].value);

   // Reformat Total
   _ThisForm["Total0"].value = FPNFormatZS
   (_ThisForm["Total0"], _fldsFmt["Total"]);

}

EndJscriptIncludes
```

Figure 7-07: Custom Calculations with JavaScript

This is an example of custom calculation where the **Total = FieldA + FieldB.**

The EForm Preferences will have a section for JavaScript similar to the one below:

BeginJscriptIncludes
function MainCalculation {
Insert Your Calculations Here.
}
EndJscriptIncludes

The calculation must begin with an open brace and end with an end brace. Each line of code must end with a semi-colon, as if putting a period at the end of a sentence to symbolize the end of a thought. The names of the fields have brackets to allow for variables.

A heading must be placed before each component within a calculation. There are four acceptable headings: **document.forms[0], $frm, the formname** or **_ThisForm.**

An additional statement must be added in the string representing the type of number being referenced. All field types are referenced as "value", but must be further defined as either being an integer or floating decimal. **ParsInt** refers to an integer. An integer is any whole number either positive or negative. Decimal values will be discarded when using ParsInt. The **TparsFloat** is a floating decimal and will accept any number, including a decimal.

Changing a Field’s Color and Border

In the Form Preferences, electronic fields have default colors and these can be globally changed there on the Form ID Tab. The User input is Black, the background Green identifies the active field cursor location. The background Yellow identifies required to be filled-in, and the background of White is read-only and cannot be edited.

Here we will discuss how you can individually change a field’s color or border using JavaScript code within the programming section of the Form Preferences. The code is different for HTML and PDF E-Forms, so examples of both are given below.

Changing the Color of the User Input

The default color for field user input is Black. Controlling this color value is different between HTML and PDF.

For HTML E-Forms: Add the following JavaScript code within the Form Preferences to allow the form to Initialize with a field input color change, before it reads the main calculations.

```
BeginJscriptIncludes
function Initialize {

// Changing Text Input Color with RGB Values
// Example: $frm.FieldName0.style.color = "FF0000";
```

```
$frm.AreaCode0.style.color = "FF0000";
$frm.Phone0.style.color = "FF0000";
$frm.Extension0.style.color = "FF0000";
}

function MainCalculation {
etc...
```

The above example changes three fields to have Red input through an RGB color value. You can easily get RGB values from Options, Drawing Properties, [Set Layer Colors] dialog.

For PDF E-Forms: If your form is using the Spot Color Model, you can use the Print Tab’s "Layer" attribute within Field Properties to control the color of the user input. Default is Layer 1 = Black. Layer color assignments are controlled through the Drawing Properties, [Set Layer Colors] dialog.

For PDF Process Color E-Forms you will have to place PDF JavaScript within the Form Preferences.

```
BeginPDFMainCalculation

// Changing Text Input Color with PDF Color Names
// Example: FieldName[0].textColor = color.blue;

   AreaCode[0].textColor = color.red;
   Phone[0].textColor = color.red;
   Extension[0].textColor = color.red;

EndPDFMainCalculation
```

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

The above example changes three fields to have red input.

Changing the Background Color of the Field

There are several default colors for a field background. The Green which identifies the active field cursor location and the Yellow identifying required to be filled-in, are standards that cannot be changed.

The White for read-only and optional fields can be changed to a color. Controlling this color value is different between HTML and PDF.

For HTML E-Forms: Add the following JavaScript code within the Form Preferences with the MainCalculations to make a field background color change.

```
BeginJscriptIncludes
function MainCalculations {

// Changing Optional Field Background Color with RGB Values
// Example: $frm.FieldName0.style.background = "FF0000";
$frm.AreaCode0.style.background = "00FFFF";
$frm.Phone0.style.background = "00FFFF";
$frm.Extension0.style.background = "00FFFF";
}
EndJscriptIncludes
```

The above example changes three fields to have Cyan as a background instead of White. You can easily get RGB values from Options, Drawing Properties, [Set Layer Colors] dialog.

For PDF E-Forms: Regardless of the Color Model, you will have to place PDF JavaScript within the Form Preferences.

```
BeginPDFMainCalculation

// Changing Background Color with PDF Color Names
// Example: FieldName[0].fillColor = color.blue;

   AreaCode[0].fillColor = color.cyan;
   Phone[0].fillColor = color.cyan;
   Extension[0].fillColor = color.cyan;

EndPDFMainCalculation
```

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

The above example changes three fields to have cyan backgrounds.

Adding Borders to the Fields

The default for generating an HTML E-Form is to "Show Field Borders" and Field Backgrounds' White. The Generate E-Forms dialog has checkbox options to toggle off these defaults. To effect just one field, use the following:

```
$frm.FieldName0.style.border = "none";
```

PDF E-Forms have a checkbox for a Transparent Field Background, but does not have a switch for the field border. This will need to be accomplished with PDF JavaScript.

```
BeginPDFMainCalculation

// Changing Stroke Color with PDF Color Names
// Example: FieldName[0].strokeColor = color.black;

   AreaCode[0].strokeColor = color.black;

EndPDFMainCalculation
```

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

The above example places a black border around the field.

CHAPTER 8

Associating Databases to the E-Form

Database tables can be associated to your form through OneForm's Input and Output Database options. Once selected, the database table names are referenced within the field's Database Tab and are ready for CGI script generation.

Note 1: It is necessary to assign the database table/field associations before going into the E-Form Server Preferences for CGI script preparation.

Note 2: When a form has both input and output tables associated, they must reside in the same database. There is a maximum of 24 tables per form that can be assigned.

Note 3: Our examples use a System DSN to locate a database. For more information on locating a database, see page 120.

The ***Input*** Database option *Reads* from an existing data table and its main purpose is to pull in information to pre-populate fields. This may be as simple as pulling in a list of States into a drop-down field for selection. Using Input data table associations can make your form easier to use for your clients. One selection may trigger other fields to populate according to the database record referenced. For example, a user may select a company name causing the address, phone number, and fax number to populate as a result.

The ***Output*** Database option is used to *Write* to a database table for data storage of the form, with the ability to recall these records. Through an assigned *Output Key Field* on the form, that holds a unique identifier for each submit, you can save and retrieve the data.

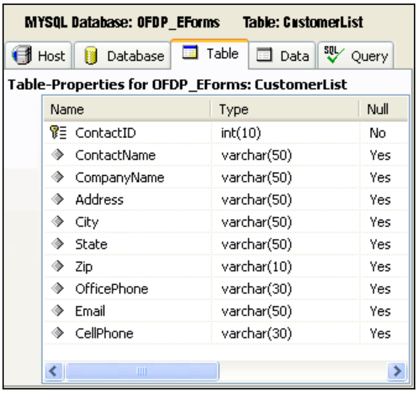
We will start by discussing the Input Database Table options for pre-populating the e-forms with data as they are being used. Then we will discuss the Output Database Table options that will be used on every e-form for the data storage of the filled-in form.

Associating Input Database Tables

Input database tables can never be written to; they are read only. Their only purpose is to populate associated fields and if needed, make them available for selection by the end user. At the time of selection, other fields can automatically be populated from the input table. Such as *CompanyName* populating *Address, City, State, Zip,* etc. In this context the CompanyName field is a trigger field.

To use an Input database table it must first be located. Our dialog can locate databases through their DSN Name or allow you to browse for an Access MDB file. Both options will then list the data tables available in the database. Once the table has been selected, its fields can then be assigned to the individual form fields needing populated.

In our example we will be using a MYSQL database data table *CustomerList*. It resides in the *OFDP_EForms* database. If you are going to follow along, you will need a database table similar to the one below:



MYSQL Database: OFDP_EForms Table: CustomerList

Table-Properties for OFDP_EForms: CustomerList

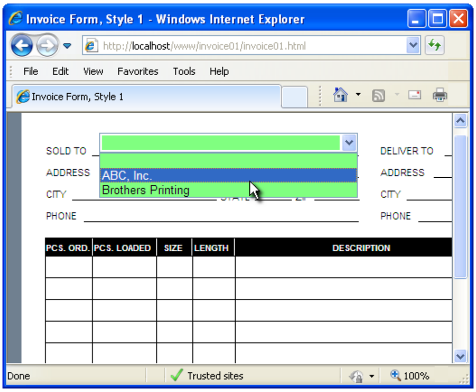
Name	Type	Null
ContactID	int(10)	No
ContactName	varchar(50)	Yes
CompanyName	varchar(50)	Yes
Address	varchar(50)	Yes
City	varchar(50)	Yes
State	varchar(50)	Yes
Zip	varchar(10)	Yes
OfficePhone	varchar(30)	Yes
Email	varchar(50)	Yes
CellPhone	varchar(30)	Yes

Figure 8-1: MYSQL Data Table "CustomerList"

Another requirement is that the HTML e-form must be served from an OFDP web server. The HTML e-form will have a reference to OneForm's "Query" CGI scripts for locating the input field data.

Using the Input Database Table Options

For our first example, we are going to simply populate a drop-down selection from a list of entries in the data table. Then we will make it trigger input of other fields, according to our selection from the list.



ABC, Inc.
Brothers Printing

Figure 8-2: List from Input Table on a Web Served HTML E-Form

Step 1: Start OneForm Designer Plus and open **invoice01/invoice01.elf** from the Forms Library. Save it into another folder. Our example has been saved into a folder on our local testing OFDP web server *C:\Inetpub\www\invoice01*.

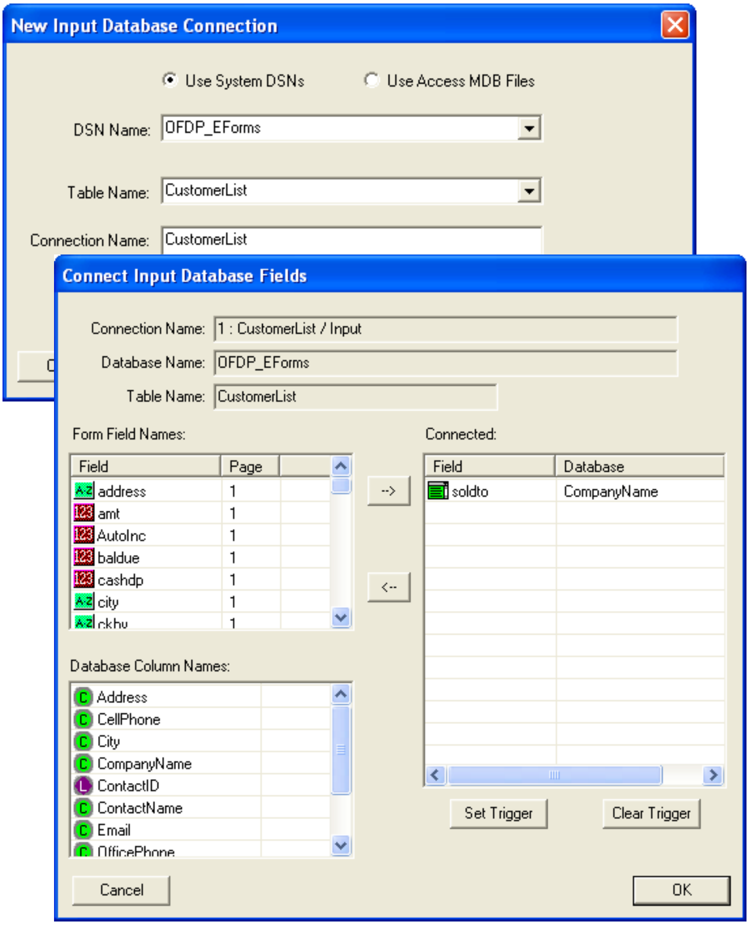
For our first example, we are going to simply populate a drop-down selection. In the field poperties of *SoldTo,* make it a **ComboBox Field Type,** so it can list the entries from the data table.

Note for PDF E-Forms: Edit the field properties CustomInfo and **delete the choices** that are there by default, leaving the Begin and End strings.

Step 2: Go to the **New Input DB Table** tool to open the dialog.

Step 3: Check the option for **Use System DSNs** and select a system database. We are selecting **OFDP_EForms** for ours. This will read that database and populate the table list.

Step 4: Select a table from the drop-down list and press **[Next].** We chose "CustomerList" as our input table.


New Input Database Connection

● Use System DSNs ○ Use Access MDB Files

DSN Name: OFDP_EForms

Table Name: CustomerList

Connection Name: CustomerList

Connect Input Database Fields

Connection Name: 1 : CustomerList / Input

Database Name: OFDP_EForms

Table Name: CustomerList

Form Field Names:

Field	Page
address	1
amt	1
AutoInc	1
baldue	1
cashdp	1
city	1
ckbv	1

Connected:

Field	Database
soldto	CompanyName

Database Column Names:

- Address
- CellPhone
- City
- CompanyName
- ContactID
- ContactName
- Email
- OfficePhone

Figure 8-3: New Input Database Connection Dialog

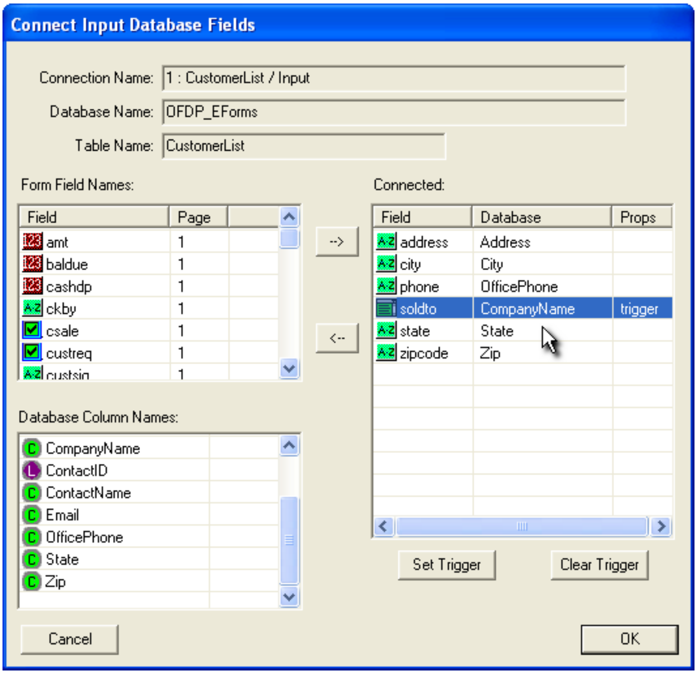
Step 5: This will display two windows on the left, *see Figure 8-3*. The top is the list of fields on the invoice01 form. The bottom window shows the field columns in the "CustomerList" table.

Highlight one field from each window on the left to represent the data from the table to be inserted into which field on the form. **Press the ==> button.** This will place them in the right-hand window showing they are associated. Notice that our *SoldTo* is assigned to "CustomerName" in the table. Click on **[Okay]** to save the association. Now you are ready to generate an HTML e-form and test it on the web server.

You will see that the ComboBox Field now has a list of choices from the input table. Ours is listing the CompanyName data. *See Figure 8-2*.

Using a Trigger Field

For our next example, we are going to make the selection from the list automatically populate (trigger) other fields. An input trigger field can be a ComboBox, Text, or Numeric field type. Our trigger field will be our *SoldTo* ComboBox field.



Connection Name: 1 : CustomerList / Input

Database Name: OFDP_EForms

Table Name: CustomerList

Form Field Names:

Field	Page
amt	1
baldue	1
cashdp	1
ckby	1
csale	1
custreq	1
custsig	1

Database Column Names:

- CompanyName
- ContactID
- ContactName
- Email
- OfficePhone
- State
- Zip

Connected:

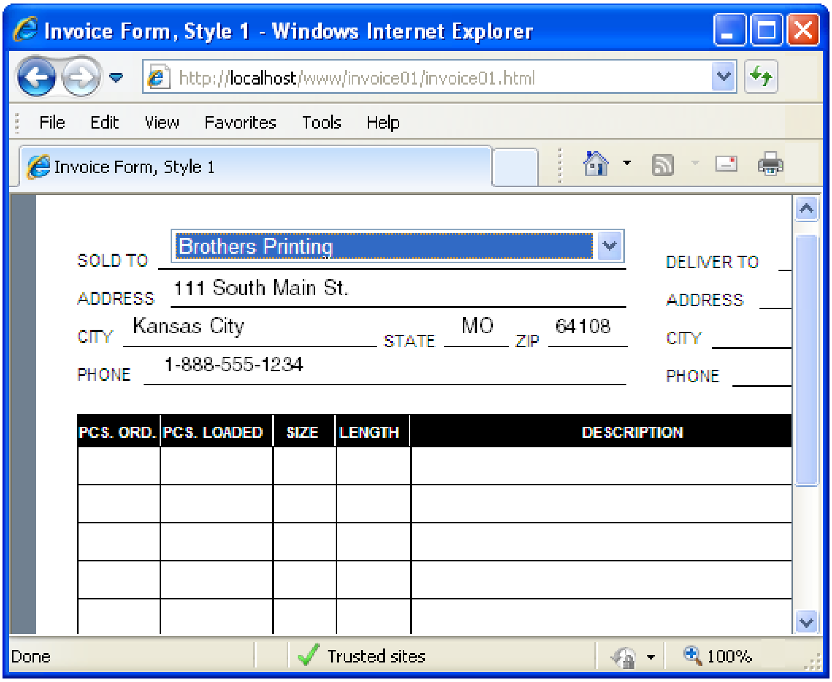
Field	Database	Props
address	Address	
city	City	
phone	OfficePhone	
soldto	CompanyName	trigger
state	State	
zipcode	Zip	

Figure 8-4: Edit Input Database Connection Dialog

Step 6: Go back into the input table dialog by selecting the **Edit Input DB Columns** tool. For the *Address, City, State, Zip,* and *Phone* fields, **highlight one field from each window on the left** to represent the data from the table to be inserted into the field on the form. **Click the ==> button.** This will place them in the right-hand window showing they are associated.

Step 7: We want the *SoldTo* field to trigger input of these fields. **Highlight SoldTo** in the right-hand window and click the **[Trigger]** button. Now you can **[OK]** to save. *See Figure 8-4.*

Step 8: On the form, the *Address, City, State, Zip,* and *Phone,* are TEXT fields. We need to make them **Read-Only,** so that they can only be populated through the input data table trigger selection. Now you are ready to generate an HTML e-form and test it again on the web server. *See Figure 8-5.*


Invoice Form, Style 1 - Windows Internet Explorer
http://localhost/www/invoice01/invoice01.html
File
Edit
View
Favorites
Tools
Help
Invoice Form, Style 1
Done
Trusted sites
100%

SOLD TO Brothers Printing

ADDRESS 111 South Main St.

CITY Kansas City STATE MO ZIP 64108

PHONE 1-888-555-1234

DELIVER TO

ADDRESS

CITY

PHONE

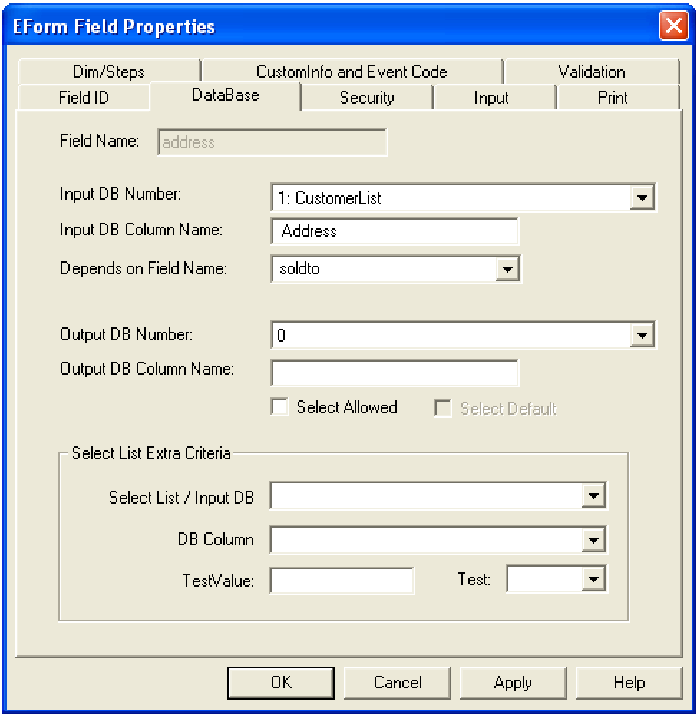
PCS. ORD.	PCS. LOADED	SIZE	LENGTH	DESCRIPTION

Figure 8-5: Selection Triggers Associated Fields

Your e-form can have more than one input table, just associate the table to the form and designate what fields should get tied together. Just remember all trigger fields must be ComboBoxes, Text, or Numeric field types. With the Text and Numeric triggers, you must know the exact value to type in. Comboboxes will display a list.

Field Properties Database Tab

When you use the database dialogs they are updating the Database Tab of those fields. If you open the field properties of *Address* and go to the Database Tab, you will see that it is associated to the input table.



Field Name: address

Input DB Number: 1: CustomerList

Input DB Column Name: Address

Depends on Field Name: soldto

Output DB Number: 0

Output DB Column Name:

□ Select Allowed □ Select Default

Select List Extra Criteria

Select List / Input DB

DB Column

TestValue: Test:

Figure 8-6: Database Tab within Field Properties

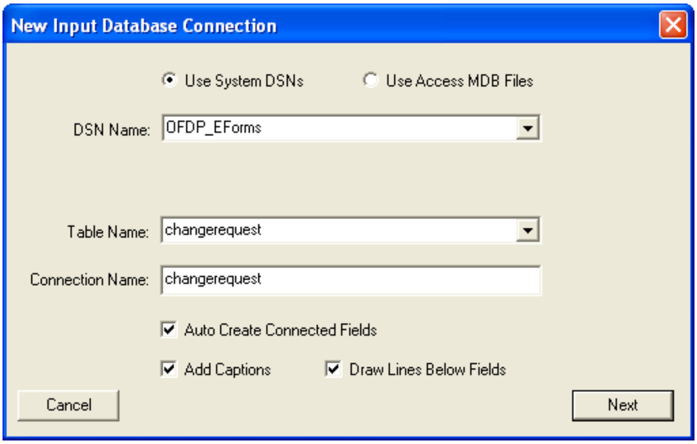
Your e-form can have up to 24 total database associations. Each time you associate an input table, they are assigned a consecutive number. This lets you know how many you have used.

The Database Tab also indicates what field triggers the current field.

Create New Fields from Input Table

There is an option within the Input's New Connection dialog which creates new fields on your form that are automatically associated to the input database table. This feature also has options to label and underline the fields. When you select [Next], you will notice on the right-hand side, that the database table columns are automatically assigned to fields on the form with identical names.

Don't forget to select the trigger as we did previously.



New Input Database Connection

● Use System DSNs ○ Use Access MDB Files

DSN Name: OFDP_EForms

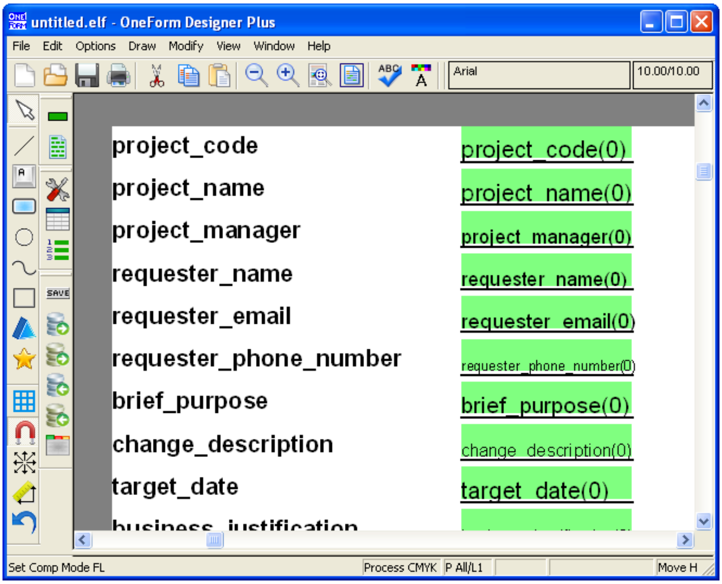
Table Name: changerequest

Connection Name: changerequest

☑ Auto Create Connected Fields

☑ Add Captions ☑ Draw Lines Below Fields

Figure 8-7: Check Auto Create Connected Fields



untitled.elf - OneForm Designer Plus

project_code	project_code(0)
project_name	project_name(0)
project_manager	project_manager(0)
requester_name	requester_name(0)
requester_email	requester_email(0)
requester_phone_number	requester_phone_number(0)
brief_purpose	brief_purpose(0)
change_description	change_description(0)
target_date	target_date(0)
business_justification	

Figure 8-8: New Fields in Upper-Left Corner of Page

After selecting [OK] on the input database dialog, you will see newly created fields on your form. They will always appear in the upper-left corner of the page and be TEXT field types.

These fields are ready to be moved into a more suitable location on your e-form.

Associating a New Output Database Table

An output database table is used to save the form's data. For example, when the form is filled in and the [Submit] button is clicked, the data in the fields will be *Written* to the output database table and stored as a record.

An e-form can have one primary output table and multiple output sub-tables. The program checks whether an output "Primary" table has already been established on the form. If so, it will take you directly to the sub-table dialogs.

A unique feature of OFDP is the *New Output Table* option. This function prepares the form fields for data collection by assigning them to a database table field of the same name. It also gathers the information necessary for the "Create Table" CGI script. When this script is run on the web server, it will create a brand new database table that includes every field on the form.

If the table name already exists, it will update the table with any new fields needed. It will never remove a DB field or shrink a field length, but it will add fields and lengthen field character counts. Using the "New Output Table" option is a very easy way to get started.

The DB Table Output Key Field

The primary output database table must have what is called an "Output Key Field" for data control. This is a field that holds unique entries specifically for record identification. OneForm provides two output key field choices for those who do not want to program their own: *AutoInc* and *FormID*. These fields are supplied through the *Add Button* tool or *Add EForm Buttons and Fields* menu.

The *AutoInc* field is for automatic incrementing each submitted record starting with 1. It is a numeric field with the properties of Integer Only with No Separators, 11 Digits. In the Output DB Table dialog, when selecting "AutoInc" as the Output Key Field, also select "AutoIncrement" as the Output Key Format.

If you are using a MYSQL database, it will have an auto-increment property that controls the sequential numbering. Each subsequent submit will query the existing table and make a new record incrementing the AutoInc value by 1.

With SQL, this action is identical, except it queries a specific "sequence number table" that holds a list of all OFDP tables and their last submitted sequence number.

OFDP's *FormID* field generates a unique number to identify each record (for example: 58a7c92e7b1541fc9222146575e51ebc), no number being repeated. In the Output DB Table dialog, when selecting "FormID" as the Output Key Field, also select "MD5-UserID-TOD" as the Output Key Format.

On the following pages, we will go through the steps to associate one of the OFDP library forms to an output database table.

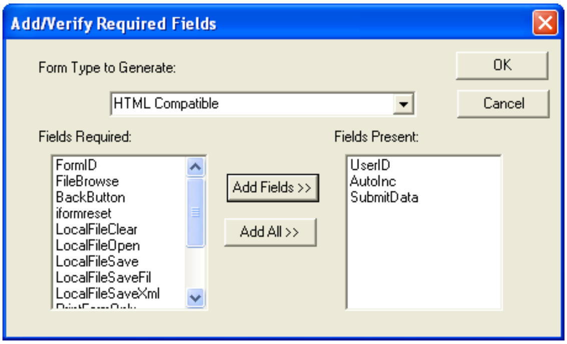
Open ELF and Add Required Fields

Step 1: Start OneForm Designer Plus and open **invoice01/invoice01.elf** from the Forms Library. Save it into another folder. Our example has been saved into a folder on our local testing OFDP web server *C:\Inetpub\www\invoice01*.

In this example, we will associate the Invoice form to a brand new table, create the table, and then run through a test of submitting and retrieving new data.

Add the AutoInc Primary Key and Submit Data Button

Step 2: We will be using OFDP's *AutoInc* field for sequentially numbering the data records starting with the number 1. Let's apply it to our form by clicking the **Add Button** *(and fields)* tool.



Form Type to Generate: HTML Compatible

Fields Required:
FormID
FileBrowse
BackButton
iformreset
LocalFileClear
LocalFileOpen
LocalFileSave
LocalFileSaveFil
LocalFileSaveXml

Fields Present:
UserID
AutoInc
SubmitData

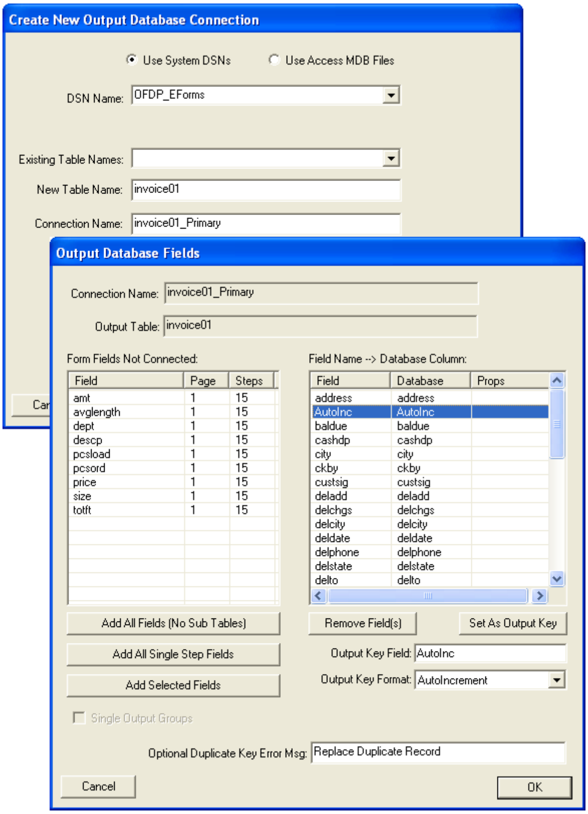
Figure 8-9: Required Fields are Easily Added

Within its dialog select **HTML Compatible** and choose to add the **AutoInc** field. The AutoInc field is a read-only incremental numbering field specifically designed for use as an Output Key Field for HTML and PDF E-Forms.

We also need a Submit data button, so go ahead and add that to your form. These fields will be placed in the upper right-hand corner of your form. They can be moved if needed.

The Output Database Dialogs

Step 3: Go to the tool **New Output DB Table.**


Create New Output Database Connection
Car
Output Database Fields
Add All Fields (No Sub Tables)
Add All Single Step Fields
Add Selected Fields
Remove Field(s)
Set As Output Key
Cancel
OK

● Use System DSNs ○ Use Access MDB Files

DSN Name: OFDP_EForms

Existing Table Names:

New Table Name: invoice01

Connection Name: invoice01_Primary

Connection Name: invoice01_Primary

Output Table: invoice01

Form Fields Not Connected:

Field	Page	Steps
amt	1	15
avglength	1	15
dept	1	15
descp	1	15
pcsload	1	15
pcsord	1	15
price	1	15
size	1	15
totft	1	15

Field Name --> Database Column:

Field	Database	Props
address	address	
AutoInc	AutoInc	
baldue	baldue	
cashdp	cashdp	
city	city	
ckby	ckby	
custsig	custsig	
deladd	deladd	
delchgs	delchgs	
delcity	delcity	
deldate	deldate	
delphone	delphone	
delstate	delstate	
delto	delto	

Output Key Field: AutoInc

Output Key Format: AutoIncrement

□ Single Output Groups

Optional Duplicate Key Error Msg: Replace Duplicate Record

Figure 8-10: Create a New Output Database Connection

Step 4: Check the option **Use System DSN** and locate an existing database from the DSN Name list. We are selecting **OFDP_EForms** for ours. This will read that database and populate the table names that already exist. The New Table Name will automatically populate using the ELF basename.

You can type in a different name, but check to make sure it is not already used, then click **[Next].**

Step 5: This will take you to the Output Fields Dialog for choosing the fields that are to capture data. The left-hand window lists the fields on the form that are not associated to a table. The right-hand window lists the fields that are associated and the DB column name that will receive the data.

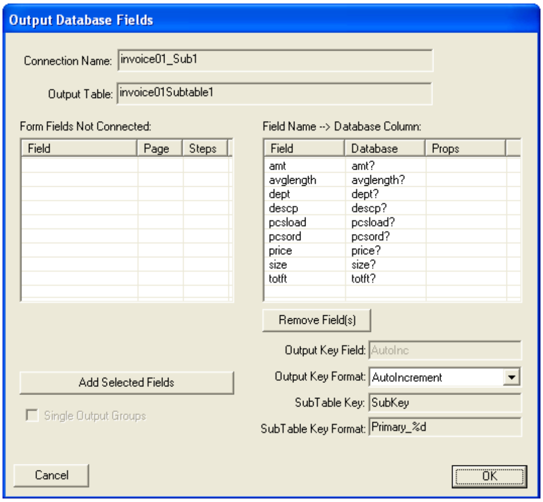
Selecting the *[Add All Fields (No Sub-Tables)]* will make each field (including stepped fields) have its own field column within the Primary table.

Selecting **[Add All Single Stepped Fields]** will collect only the single fields into the Primary table *(as shown in Figure 8-10),* leaving the stepped fields to be attached later into a sub-table. This is the option we will be using in our example. Notice that the "New Output DB Table" option, automatically names the data table column the same name as our form field.

The fields in the right-hand window identify which fields are submitting to the table. **Highlight "AutoInc"** (the primary key field) listed in this window, and click **[Set as Output Key].** Select the **Output Key Format** as **AutoIncrement. Click [OK]** to complete the table association to the non-stepped fields and close the dialog.

Adding a Sub-Table Association

Step 6: To associate the stepped fields with a sub-table, click on the **New Output DB Table** tool again. The table name will have a reference to a sub-table, click **[Next].**



Output Database Fields

Connection Name: invoice01_Sub1

Output Table: invoice01Subtable1

Form Fields Not Connected:

Field	Page	Steps

Field Name --> Database Column:

Field	Database	Props
amt	amt?	
avglength	avglength?	
dept	dept?	
descp	descp?	
pcsload	pcsload?	
pcsord	pcsord?	
price	price?	
size	size?	
totft	totft?	

Output Key Field: AutoInc

Output Key Format: AutoIncrement

SubTable Key: SubKey

SubTable Key Format: Primary_%d

☐ Single Output Groups

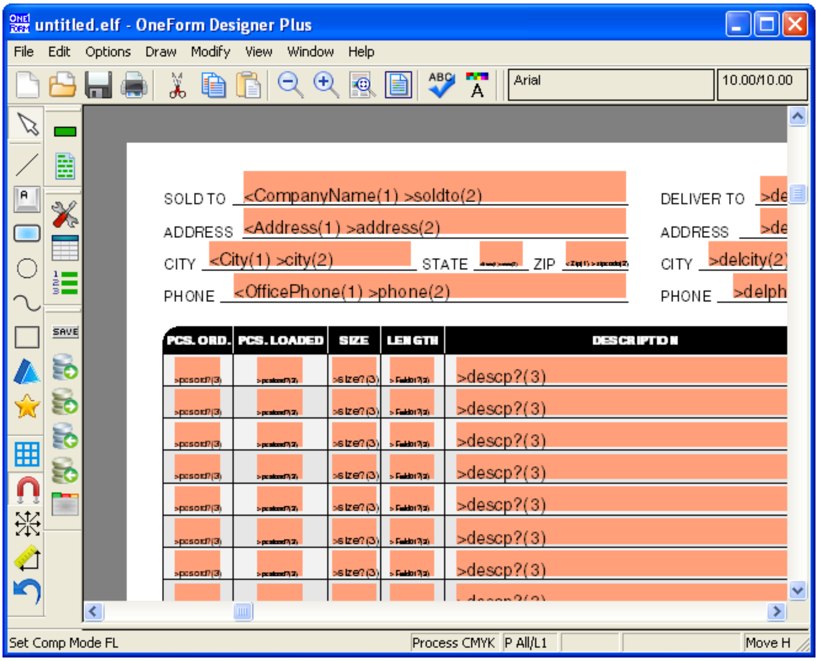
Figure 8-11: Output Database Sub-Table Dialog

The remaining fields will be listed in the left-hand window. **Highlight all stepped fields,** and click **[Add Selected Fields].** This will move them to the right-hand window showing their associated DB column name. The question mark represents the stepped-number variable. Click **[OK]** to complete the table association to the fields and close the dialog.

Viewing Database Field Names

Step 7: Going to the **View** menu and selecting the **EForms Fields, Show DB Table Names,** will display the field name within the DB table and the (#) behind each name is a reference to which table.

Look at the field "CompanyName(1)" next to SOLD TO. The (1) represents the first table connected, which was the input table. The "soldto(2)" represents the primary output table. The "<CompanyName(1) >soldto(2)" means the data will come in from the input table and be submitted into the primary output table. The reference to (3) in the "descp(3)" is the sub-table.



SOLD TO <CompanyName(1) >soldto(2)
ADDRESS <Address(1) >address(2)
CITY <City(1) >city(2) STATE <State(1) >state(2) ZIP <Zip(1) >zipcode(2)
PHONE <OfficePhone(1) >phone(2)

DELIVER TO >de
ADDRESS >de
CITY >delcity(2)
PHONE >delph

PCS. ORD.	PCS. LOADED	SIZE	LENGTH	DESCRIPTION
>pcsord?(3)	>pcsloaded?(3)	>size?(3)	>Field1?(3)	>descp?(3)
>pcsord?(3)	>pcsloaded?(3)	>size?(3)	>Field1?(3)	>descp?(3)
>pcsord?(3)	>pcsloaded?(3)	>size?(3)	>Field1?(3)	>descp?(3)
>pcsord?(3)	>pcsloaded?(3)	>size?(3)	>Field1?(3)	>descp?(3)
>pcsord?(3)	>pcsloaded?(3)	>size?(3)	>Field1?(3)	>descp?(3)
>pcsord?(3)	>pcsloaded?(3)	>size?(3)	>Field1?(3)	>descp?(3)
>pcsord?(3)	>pcsloaded?(3)	>size?(3)	>Field1?(3)	>descp?(3)

Figure 8-12: View Database Names

Testing the Output Database Connection

Steps 8 through 11 are the standard procedures for getting an HTML E-Form server ready. Since you cannot really test your data table associations without going through the entire server ready process, it is quickly described. If you need more information, these steps are described in detail in *Chapter 14.*

Generating the CGI Scripts

The E-Form Server Preferences allow you to specify the actions of the CGI Scripts that run on the web server to save, search, list, and retrieve the e-form. The scripts also manage the creation of the server data table described in the previous Output Database dialogs. The objective is to generate all necessary server scripts to host the HTML form and connect the fill fields to a server output database.

Step 8: Start by clicking the **Server Preferences** tool. Go directly to the **Gen Scripts Tab** and click on the **Generate Scripts** button. This will use the defaults for creating scripts and administrative pages for our HTML e-form.

Generate the Final E-Form HTML

Step 9: At this point, the final HTML e-form is ready to be generated. Go to the **Generate HTML** tool. Choose **"Generate Code for" Electronic Form - HTML** and click **[Okay].**

Copy Files to the Server

Step 10: Copy the project folder and its contents to the document root location on the web server and test through a web browser.

Note: Copy the files to the appropriate location as directed by your Webmaster. If you are using FTP, all *.html* and *.cgi* files generated in this directory MUST be sent as ASCII. They will not execute otherwise. Any images specific to this project (*.gif* and *.jpg* files) MUST be sent as BINARY. Also, all CGI script files must have executable permissions on the server.

Create the Data Table for the E-Form

Step 11: To start testing, you need to locate the administration HTML file for your project (admin_project.html), and launch it from your browser.

The URL will be similar to the following:

http://servername/invoice01/admin_invoice01.html

To create the database tables, select the **Create Database Table** link to run that script. A message should appear stating that the tables have been created successfully.

Run and Test the E-Form

Test the e-form through the **admin_invoice01.html** links. With it you can open the new e-form and submit data, and examine the contents of the database.

Start testing the e-form by clicking the link **"Fill Out New HTML Form"** and then inputting data into the e-form and clicking the **[Submit]** button. Also, click on **"Find an Existing Form"** and use the Search to locate the saved record for retrieval. AutoInc is a sequential number for each submit, that starts with 1.

If corrections need to be made, make the changes within OneForm Designer Plus and then copy the updated files back to the server and test again.

Working with Existing Data Tables

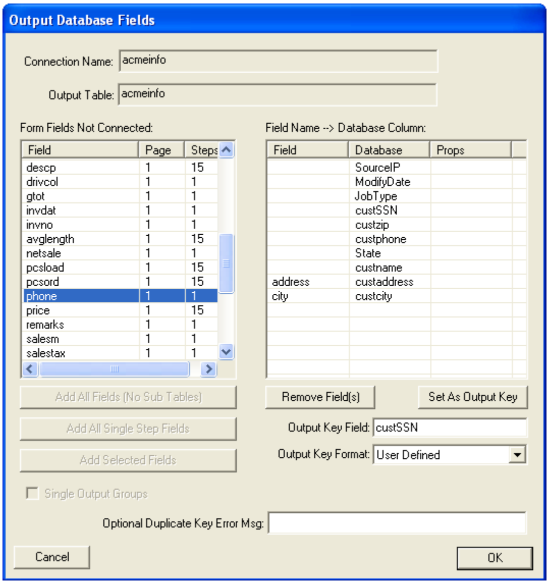
Edit Fields/Columns of a Table Association

There are both Input and Output tools and menus for *Editing the Field Database Associations* of the form. This allows you to add or remove field database associations. As you do so the field properties of those fields will be modified. Then by generating new scripts (if you have an existing table you are testing), you can either drop it or modify it through the admin page links. Remember, a table is not created until you execute the scripts on the server.

If the table name already exists, running create will update the table with any new fields needed. It will never remove a DB field or shrink a field length, or change a field type, but it will add fields and lengthen field character counts.

Associate Fields Using an Existing Output DB Table

If the output database table exists, and you want to associate your fields using it, use the "Connect to an Existing Table" option. It is located at *Options, EForm Options, EForm Output Database, Connect to Existing Table.*



Output Database Fields

Connection Name: acmeinfo

Output Table: acmeinfo

Form Fields Not Connected:

Field	Page	Steps
descp	1	15
drivcol	1	1
gtot	1	1
invdat	1	1
invno	1	1
avglength	1	15
netsale	1	1
pcsload	1	15
pcsord	1	15
phone	1	1
price	1	15
remarks	1	1
salesm	1	1
salestax	1	1

Field Name --> Database Column:

Field	Database	Props
	SourceIP	
	ModifyDate	
	JobType	
	custSSN	
	custzip	
	custphone	
	State	
	custname	
address	custaddress	
city	custcity	

Output Key Field: custSSN

Output Key Format: User Defined

□ Single Output Groups

Optional Duplicate Key Error Msg:

Figure 8-13: Connect to Existing Database Table

It will allow you to choose a DSN source for the database and list its existing tables. After choosing a table, selecting [Next], will display two windows. The left is the list of fields on the form. The right window shows the field columns in the existing table.

Highlight one field from the left window to represent the data from the field, and then **pick the DB column field name.** This will place them in the right-hand window showing they are associated. Do this for all fields that need data submitted to the database table. Then **[OK]** to save.

This updated the Database Tab of those fields and prepared the form for generating database CGI scripts.

Creating Fields from an Existing Database Table

There is an option within the Input's New Connection dialog which creates new fields on your form that are automatically linked to an input database table. After selecting "Auto Create New Fields" and proceeding to the next dialog, you will notice that the database table columns are automatically linked to fields on the form with identical names. You would still need to select an input trigger and move the fields to their proper position.

If you are wanting these fields for output only, just remove the input table association and establish a new output table association.

Remove Database Associations

Both options for Input and Output Database have a *Remove Connection*. This will list the database number and table names. Identify the table to remove and select the [Remove Button]. You must remove all sub-table associations, before it will allow you to remove the primary table association.

View Database Tree

To view a list of fields linked to data tables, go to *Options, EForm Options, Database Options, EForm Database Tree.*

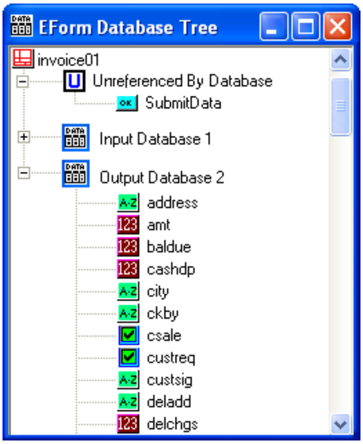


Figure 8-14: View List of Fields Connected to Database

At a quick glance, this enables you to see the hierarchy of database tables, and the form fields they reference.

Locating a Database for the OFDP Dialogs

When associating database tables to fields, our dialogs want the database name. You can use either the ODBC System Data Source Name (DSN), browse for a local Access MDB database, or just type in a dummy Access MDB database name.

Creating a "System DSN"

OFDP can use a Data Source Name (DSN) to locate databases. If you are not familiar with this Windows feature, the following may help.

An ODBC System Data Source stores information about how to connect to the indicated data provider. The option for creating data sources is found under Control Panel, Administrative Tools, Data Sources (ODBC), System DSN Tab. Click [Add] to select a driver for your data source. Dialogs to assist in setup will appear, basically it needs an English Name and Description. The English Name is used in our dialogs.

Browse for a Local Access MDB Database

When selecting the option "Use Access MDB File" a Browse button will appear letting you browse for a local .mdb file. This will allow the table names list to show that database’s current table names.

Use a Dummy MDB Database Name

If you want to use the "Create New Table" option and cannot locate a local database, you can just type in a dummy database name. Use the option "Use Access MDB File" and type in a dummy .mdb file name and select [Next]. This will go ahead and assign your new table name to that "dummy" database name. You will be able to generate CGI scripts without any problems. The create CGI script will still create the table within the web server’s linked database, when it is run on the server. OFDP doesn’t really need the database name for the "Create New Table" option. It just keeps a count of how many tables are referenced within a database.

CHAPTER 9

E-Form Server Preferences Dialogs

The E-Form Server Preferences allow you to specify the actions of the CGI Scripts that run on the web site server to save, search, list, and retrieve the E-Form. These scripts also manage the creation of the server data table described in the Output Database dialogs. The objective of server preferences is to generate all of the CGI server scripts to host the HTML form, and connect the fill fields to a server output database.

Note: OneForm Designer Plus creates Perl5 CGI Scripts for a "Generic UNIX Apache Platform" using UNIX pathing. These scripts will also work with Microsoft Windows IIS Servers. UNIX file naming is case dependent, and because of this, lowercase is required for the scripts of your E-Form project. The base name of the ELF file as lowercase, is used to name the CGI scripts and is also used throughout the scripts for database connectivity. Using lowercase letters with no spaces in the file names avoids any file naming problems associated with CGI scripts not locating files because they do not match exactly.

Requirements

Before generating scripts, there are three requirements:

- ELF filename must be lowercase with no spaces.
- Required fields must be added.
- The output database associations must be assigned.

WEBMASTERS:

Refer to the Chapter titled ***E-Forms Server Setup*** for information about setting up an internet server, installing E-Forms, and database connectivity.

List of Files Required for Each E-Form

DRAW FORM & GENERATE HTML E-FORM	
USAGE	**FILE NAMES**
OneForm Designer Plus Form	*eformname*.elf
Fillable and Printable E-Form	*eformname*.html

CREATED BY E-FORM SERVER PREFERENCES	
USAGE	**FILE NAMES**
E-Form Database Scripts & Files Create a New Table Drop an Existing Table List of All Records in Database Show Database Record Details List Database Records for Deletion Delete Record from Database Table	create_*eformname*Table-pl.cgi drop_*eformname*Table-pl.cgi list_*eformname*-pl.cgi details_*eformname*-pl.cgi dlist_*eformname*-pl.cgi delete_*eformname*-pl.cgi
Transaction Scripts & Files Save Submitted Text to Database Display Database Search Form Search Database and List Results Open Selected E-Form Open New Blank E-Form	save_*eformname*-pl.cgi search_*eformname*-pl.cgi and search_*eformname*.html results_*eformname*-pl.cgi open_*eformname*-pl.cgi blank_*eformname*-pl.cgi
Help with Finding New E-Forms Show List of Forms and Options Select New Form or Find Form	admin_*eformname*.html index_*eformname*.html

When generating server scripts, all of the files shown above will be created from the base ELF file. As you can see, each file created will include the basename of the ELF file in its name. So, the ELF file name is important.

All of these files will need to be copied over to your Internet server. It is best to start by creating a project directory specifically for each E-Form and copy the entire folder to the server's document root.

E-Form Server Preferences Options

The **E-Form Server Preferences** option is found as tool on the E-Form Server toolbar. It is also under **Options, E-Form Options.**

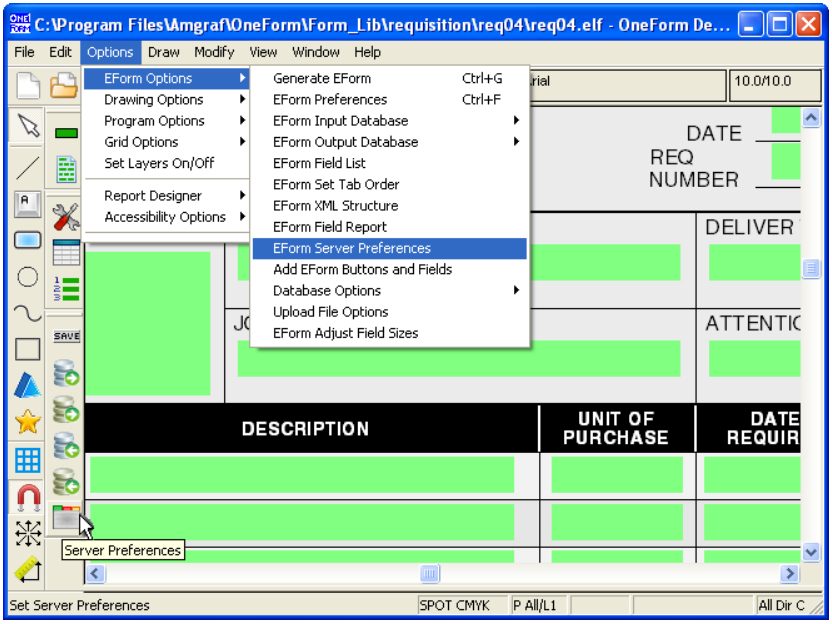
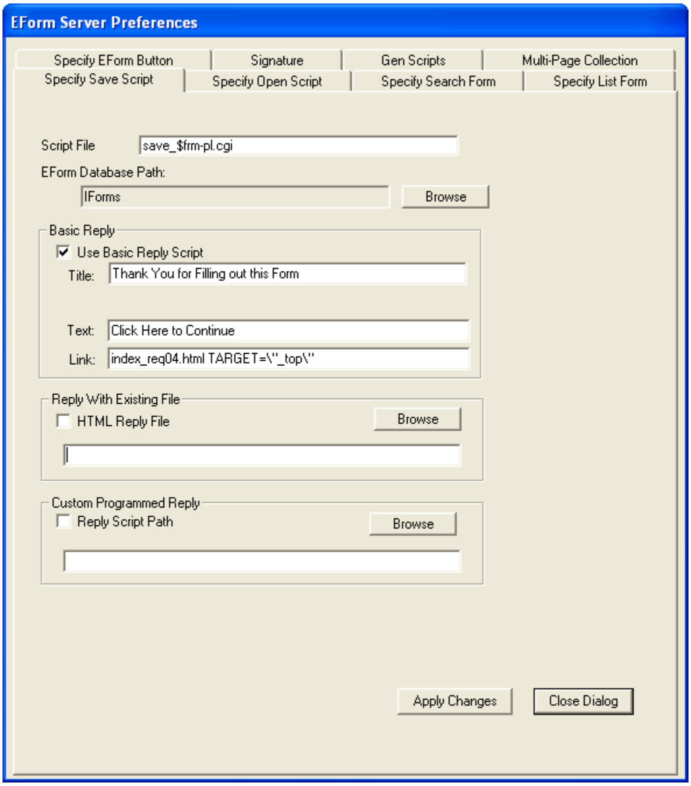


Figure 9-1: E-Form Server Preferences Tool and Menu

The E-Form Preferences dialog consists of eight tabs. Everything is already set for normal use. In the following pages we will discuss each dialog and the options that are available for customizing your CGI scripts. When making a change to any of the tabs, clicking [Apply] will save the changes.

Note: The Signature Tab is discussed in the *Forms with Signature Fields chapter*. The Multi-Page Collection Tab is used with the optional PDF Generator Module for printing continuation pages.

Specify Save Script



Script File: save_$frm-pl.cgi

EForm Database Path: IForms

Basic Reply

☑ Use Basic Reply Script

Title: Thank You for Filling out this Form

Text: Click Here to Continue

Link: index_req04.html TARGET=\"_top\"

Reply With Existing File

□ HTML Reply File

Custom Programmed Reply

□ Reply Script Path

Figure 9-2: Specifying the Save Script

The *Specify Save Script* dialog is automatically filled out for you.

The name of the Save script being created is *save_eformname-pl.cgi*. (The *$frm* tells it to pick up the basename of the ELF.)

The "E-Form Database Path" is the location that the Save script will place the data based upon the currently selected system DSN configuration.

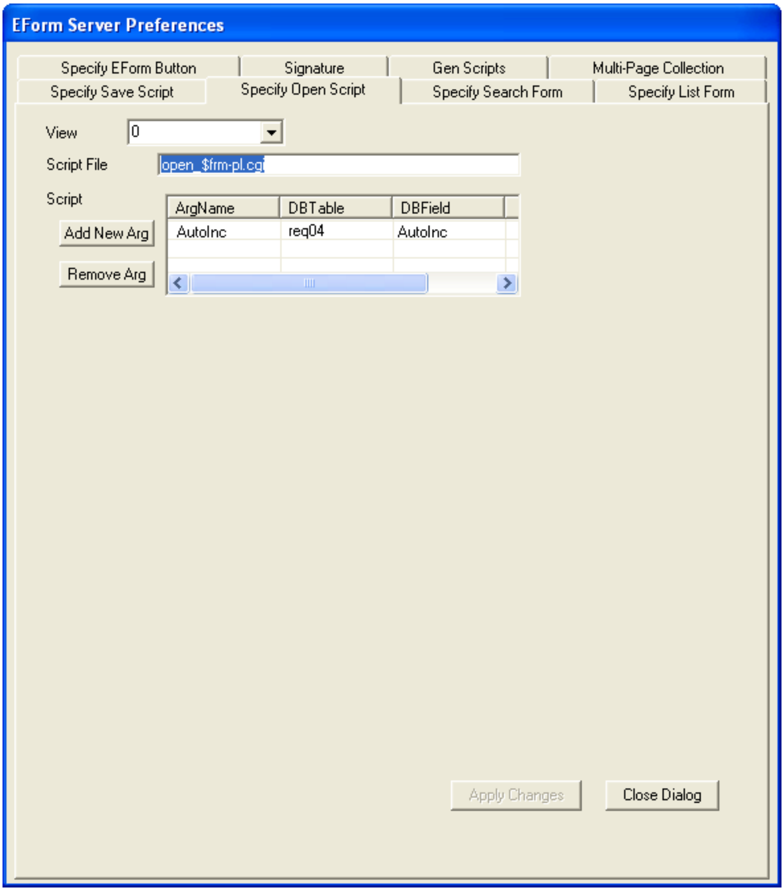
The Basic Reply Script is a message sent back once the user submits their information. This message lets the user know that their data has been received. It contains a link to another page called the *index_eformname.html*.

If you do not want the Basic Reply Script, click on one of the other two choices: HTML Reply File or Reply Script Path. Then, supply the complete path of your custom HTML file or CGI script

Specify Open Script

The *Specify Open Script* tab is already set for normal use. The Open script being created is *open_eformname-pl.cgi.*

This tab references the previously set Output Database Table Primary Key.



View: 0

Script File: open_$frm-pl.cgi

Script

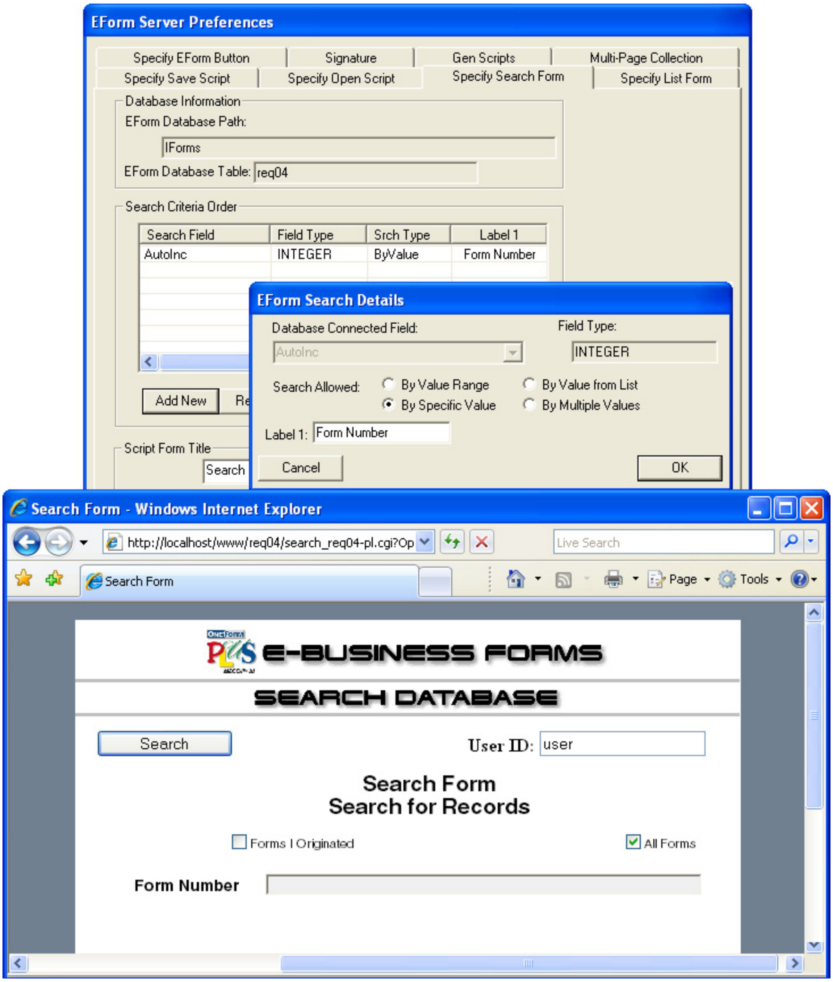
ArgName	DBTable	DBField
AutoInc	req04	AutoInc

Figure 9-3: The Open Script Dialog

Specify Search Form and Specify List Form

The Search and List Tabs are used to build the search and list forms. Each time a user fills out an E-Form and saves, a record is written in the data table. To retrieve this record as a filled-in E-Form, a search form is used to locate the data. The **Specify Search Form** tab builds this search HTML form using the E-Form field names specified.

The **Specify List Form** tab indicates how the search results are displayed. Both tabs work in the same manner and will always have at least the Output Database Table Primary Key.



EForm Server Preferences

Database Information

EForm Database Path: \Forms

EForm Database Table: req04

Search Criteria Order

Search Field	Field Type	Srch Type	Label 1
AutoInc	INTEGER	ByValue	Form Number

Script Form Title: Search

EForm Search Details

Database Connected Field: AutoInc

Field Type: INTEGER

Search Allowed: ○ By Value Range ● By Specific Value ○ By Value from List ○ By Multiple Values

Label 1: Form Number

Search Form - Windows Internet Explorer

http://localhost/www/req04/search_req04-pl.cgi?Op

User ID: user

Search Form
Search for Records

□ Forms I Originated ☑ All Forms

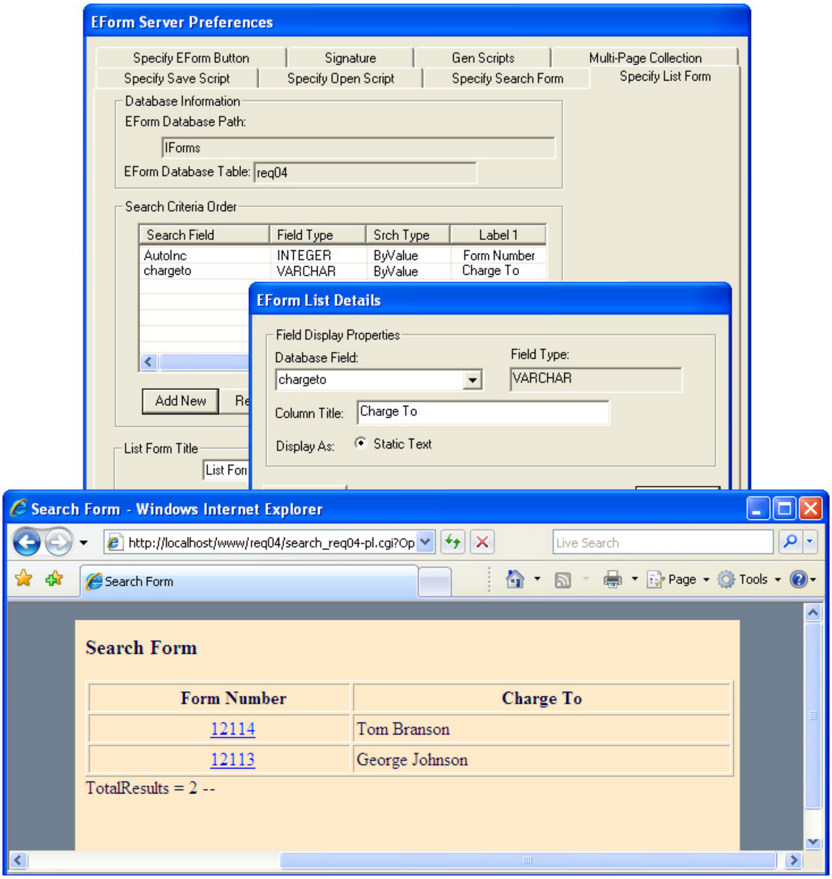
Form Number

Figure 9-4: The Search Dialog and Search Form it Creates

[Add New] allows you to locate a database connected field on the form and define its search: by value range, specific or multiple values, or from a list.

[Remove] will allow you to remove fields from the search. Just highlight the search field name and click [Remove]. [Modify] lets you go back and modify the search details previously specified. Through the [Move Up] and [Move Down] options you can rearrange the display of the resulting list.

After setting up the Search Form, click on the [Copy to List] button. This will copy all settings to the Specify List Form dialog, saving you from having to set it up manually.



Database Information

EForm Database Path: IForms

EForm Database Table: req04

Search Criteria Order

Search Field	Field Type	Srch Type	Label 1
AutoInc	INTEGER	ByValue	Form Number
chargeto	VARCHAR	ByValue	Charge To

List Form Title: List For

Field Display Properties

Database Field: chargeto

Field Type: VARCHAR

Column Title: Charge To

Display As: ● Static Text

http://localhost/www/req04/search_req04-pl.cgi?Op

Search Form

Form Number	Charge To
12114	Tom Branson
12113	George Johnson

TotalResults = 2 --

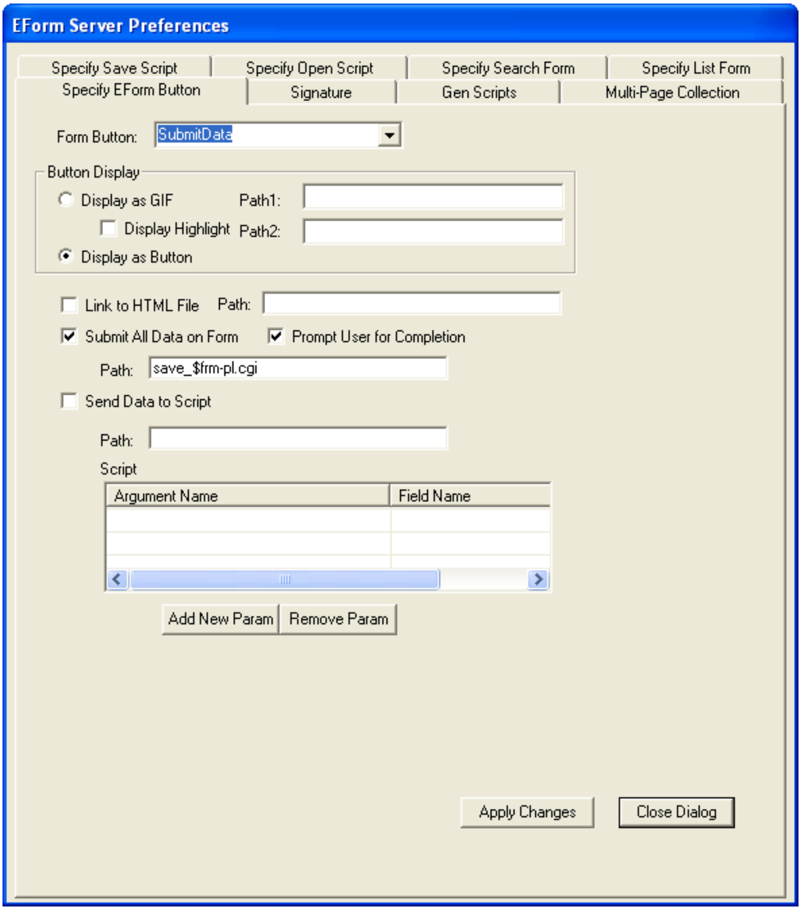
Figure 9-5: Specifying the Results List Script

Rename the Title of the List **"Results of Search".** Clicking **[Apply]** will save your changes to this tab.

The Specify E-Form Button Tab

Be sure to select the **SubmitData** button from the Form Button drop-down list. Then indicate it to **Submit All Data on Form** by clicking the checkbox as shown in the following illustration.

When **Prompt User for Completion** is checked, a message will appear (to the user) before submitting any data. Click **[Apply]** to save your settings.



Form Button: SubmitData

Button Display

○ Display as GIF Path1:

□ Display Highlight Path2:

● Display as Button

□ Link to HTML File Path:

☑ Submit All Data on Form ☑ Prompt User for Completion

Path: save_$frm-pl.cgi

□ Send Data to Script

Path:

Script

Figure 9-6: Submit Data Button Settings

The **Button Display** can be an image instead of the standard button icon. Check "Display as GIF" and type in the path of the GIF file into "Path1". If you want a different image when the mouse rolls over the button, then check "Display Highlight" and type in the path of the secondary GIF file into "Path2".

Link to HTML File – To have a button become a link to another HTML file, identify the button from the Form Button drop-down list. Then click on the checkbox for "Link to HTML File" and type in a URL path to the HTML file the button should open.

Send Data to Script –If you have your own custom save script and want to use it, check "Send Data to Script" and type in a URL path to the CGI script the button should use. The arguments and fields being referenced should also be set.

When clicking [Apply] on this dialog, the appropriate code is placed into the button field, Custom Info.

The Generate Scripts Tab

The **Gen Scripts** tab allows you to create the server scripts and administrative HTML pages needed. If you have met the requirements for server preferences all of the necessary options will be checked and you just click on [Generate Scripts].

Database Scripts

Database Create and Drop (remove) Scripts allow you to create a database table for the E-Form and remove the table. The Database Save Script is what controls the save function of the E-Form. The Database Open Script allows retrieval of data once saved.

Search for Record Script

This option creates the search script allowing the database to be searched through by specified fields and search criteria.

Diagnostics Scripts

If you check "Delete Record Script", then a CGI script and HTML Form will be created that allows you to delete a record from the database table.

Helper Index Pages

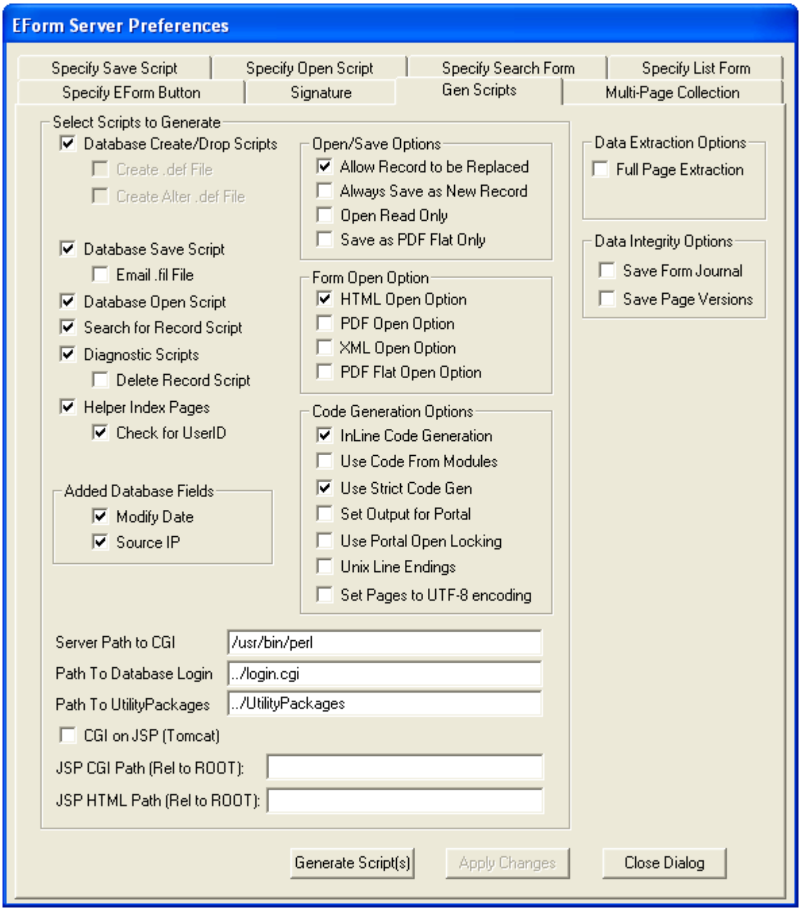
The administrative HTML and index HTML pages are created with links to all of the CGI scripts for creating a database, browsing the database, opening the E-Form and Searching through the database.

Added Database Fields

If the Modify Date or Source IP is checked, the database create script will automatically include these fields when it creates the database table.

Open/Save Options

The default action for E-Forms is to allow modified forms to be saved as replacements for prior versions. If you wish, you may select the option labeled "Always Save as New Record" in order to force every version of a filled and saved E-Form to be preserved in the database. If your E-Form is to be used for data presentation only, click the option "Open Read Only".



Select Scripts to Generate

- ☑ Database Create/Drop Scripts
  - □ Create .def File
  - □ Create Alter .def File
- ☑ Database Save Script
  - □ Email .fil File
- ☑ Database Open Script
- ☑ Search for Record Script
- ☑ Diagnostic Scripts
  - □ Delete Record Script
- ☑ Helper Index Pages
  - ☑ Check for UserID

Added Database Fields

- ☑ Modify Date
- ☑ Source IP

Open/Save Options

- ☑ Allow Record to be Replaced
- □ Always Save as New Record
- □ Open Read Only
- □ Save as PDF Flat Only

Form Open Option

- ☑ HTML Open Option
- □ PDF Open Option
- □ XML Open Option
- □ PDF Flat Open Option

Code Generation Options

- ☑ InLine Code Generation
- □ Use Code From Modules
- ☑ Use Strict Code Gen
- □ Set Output for Portal
- □ Use Portal Open Locking
- □ Unix Line Endings
- □ Set Pages to UTF-8 encoding

Data Extraction Options

- □ Full Page Extraction

Data Integrity Options

- □ Save Form Journal
- □ Save Page Versions

Server Path to CGI: /usr/bin/perl

Path To Database Login: ../login.cgi

Path To UtilityPackages: ../UtilityPackages

□ CGI on JSP (Tomcat)

JSP CGI Path (Rel to ROOT):

JSP HTML Path (Rel to ROOT):

Figure 9-7: Standard Generate Scripts Dialog

Note: You may wish to have two working versions of your E-Form, the first for filling and the second for displaying data. Use these Open/Save options to generate the scripts for each of these versions from a single design file.

Form Open Options

Select how many different ways you need to open the form. You will get a link on the helper HTML pages for each selection type.

Code Generation Options

The default is "InLine Code Generation".

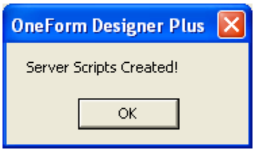
Server Pathing

The default paths for the CGI Interpreter and Database Login file are shown at the bottom of the Generate Scripts panel. Change these paths as desired for your particular web site. If you are not sure of these paths, ask for assistance from your webmaster.

[Generate Scripts]

Click **[Generate Scripts]** to create CGI scripts for the checked functions.

A pop-up message will confirm that your scripts have been created. You can **[OK]** this and **[Close]** the eform Preferences dialog. Clicking **[Apply]** will save your changes to this tab.

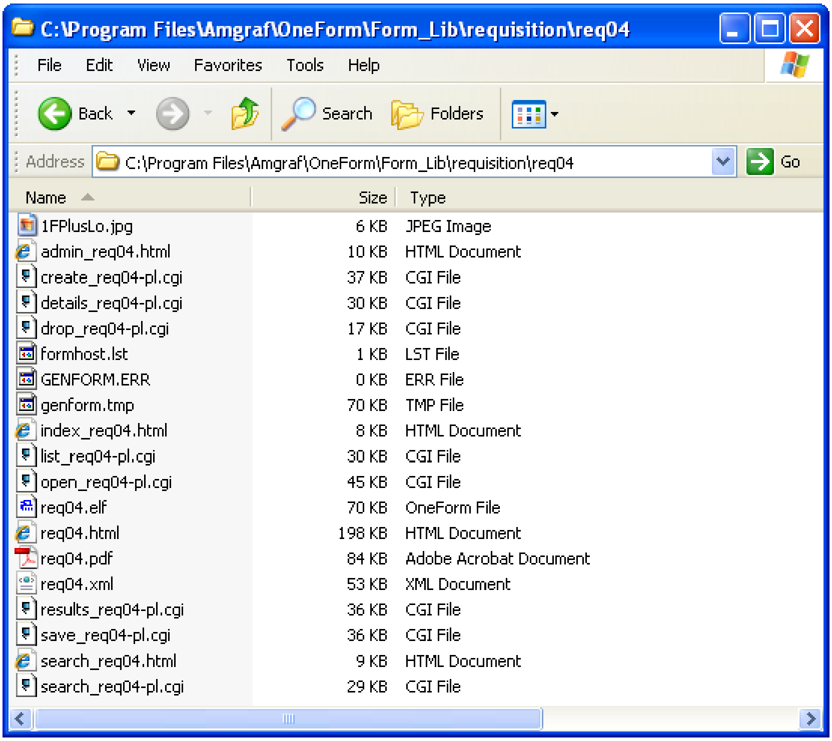


OneForm Designer Plus

Server Scripts Created!

Figure 9-8: Generate the Server Scripts

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



C:\Program Files\Amgraf\OneForm\Form_Lib\requisition\req04

Address: C:\Program Files\Amgraf\OneForm\Form_Lib\requisition\req04

Name	Size	Type
1FPlusLo.jpg	6 KB	JPEG Image
admin_req04.html	10 KB	HTML Document
create_req04-pl.cgi	37 KB	CGI File
details_req04-pl.cgi	30 KB	CGI File
drop_req04-pl.cgi	17 KB	CGI File
formhost.lst	1 KB	LST File
GENFORM.ERR	0 KB	ERR File
genform.tmp	70 KB	TMP File
index_req04.html	8 KB	HTML Document
list_req04-pl.cgi	30 KB	CGI File
open_req04-pl.cgi	45 KB	CGI File
req04.elf	70 KB	OneForm File
req04.html	198 KB	HTML Document
req04.pdf	84 KB	Adobe Acrobat Document
req04.xml	53 KB	XML Document
results_req04-pl.cgi	36 KB	CGI File
save_req04-pl.cgi	36 KB	CGI File
search_req04.html	9 KB	HTML Document
search_req04-pl.cgi	29 KB	CGI File

Figure 9-9: Files are Automatically Created by [Generate Scripts]

Copy Files to the Server

After creating the scripts and helper HTML pages, the project folder and its contents will need to be copied to the document root on the web server and test from there. Copy to the appropriate location as directed by your Webmaster.

Transferring Files with FTP

If you are using FTP, all *.html* and *.cgi* files generated in this directory MUST be sent as ASCII. They will not execute otherwise. Any images specific to this project (*.gif* and *.jpg* files) MUST be sent as BINARY.

Make CGI Scripts Executable

All CGI script files must have executable permissions on a UNIX server. You may need assistance from your Webmaster to do this.

CHAPTER 10

Creating HTML E-Forms

Before Getting Started

OneForm Designer Plus (OFDP) creates all of the files needed to deploy Electronic Forms on Microsoft Windows IIS and UNIX web servers, with databases such as SQL Server 2000, Oracle 8i, and MySQL. Although OFDP creates all of the files needed for E-Forms, you will need the assistance of a webmaster to help setup your web server.

For example, you may need help to create a Document Root location for your E-Forms, or give the CGI scripts executable permissions. Also when connecting to databases through a server, database drivers are needed. Having the assistance of a webmaster to help setup the first working E-Form and getting your web server OFDP compatible is usually necessary.

When starting a new E-Form, it is best to create a project directory specifically for that particular E-Form. If possible for management purposes, name the project folder the same as the ELF. Many files are generated and will need to be copied over to your web server. It is easier just to copy the entire folder to the server's document root.

Note: OneForm Designer Plus creates Perl5 CGI Scripts for a "Generic UNIX Apache Platform" using UNIX pathing. These scripts will also work with Microsoft Windows IIS Servers. UNIX file naming is case dependent, and because of this, lowercase is required for the scripts of your E-Form project. The base name of the ELF file as lowercase, is used to name the CGI scripts and is also used throughout the scripts for database connectivity. Using lowercase letters with no spaces in the file names avoids any file naming problems associated with CGI scripts not locating files because they do not match exactly.

Testing the look of the E-Form and local functions, like calculations and tab ordering, can be tested without moving the E-Form to the server. To make sure any images (GIF's or JPEG's) can be found, they should be stored within the project folder.

WEBMASTERS:

Refer to the Chapter titled ***E-Forms Server Setup*** for information about setting up an internet server, installing E-Forms, and database connectivity.

When generating a new E-Form, all of the files shown above will be created from the base ELF file. As you can see, each file created will include the basename of the ELF file in its name. So, the ELF file name is important.

List of Files Required for Each E-Form

DRAW FORM & GENERATE HTML E-FORM	
USAGE	**FILE NAMES**
OneForm Designer Plus Form	*eformname*.elf
Fillable and Printable E-Form	*eformname*.html

CREATED BY E-FORM SERVER PREFERENCES	
USAGE	**FILE NAMES**
E-Form Database Scripts & Files	
Create a New Table	create_*eformname*Table-pl.cgi
Drop an Existing Table	drop_*eformname*Table-pl.cgi
List of All Records in Database	list_*eformname*-pl.cgi
Show Database Record Details	details_*eformname*-pl.cgi
List Database Records for Deletion	dlist_*eformname*-pl.cgi
Delete Record from Database Table	delete_*eformname*-pl.cgi
Transaction Scripts & Files	
Save Submitted Text to Database	save_*eformname*-pl.cgi
Display Database Search Form	search_*eformname*-pl.cgi and search_*eformname*.html
Search Database and List Results	results_*eformname*-pl.cgi
Open Selected E-Form	open_*eformname*-pl.cgi
Open New Blank E-Form	blank_*eformname*-pl.cgi
Help with Finding New E-Forms	
Show List of Forms and Options	admin_*eformname*.html
Select New Form or Find Form	index_*eformname*.html

If Starting from Stratch

If you are starting from scratch there are some hints at the end of this chapter. Also, the *Business Forms Composition* reference manual has a step-by-step tutorial on drawing artwork used in our fielding example, along with details on all of the drawing tools.

Importing a File

There are many times that the form you want to make into an electronic form is already composed in another format. OneForm Designer Plus can import EPS and PDF files. If you have either of these, there is more information in the *Business Forms Composition* reference manual, *Chapter 9: Importing EPS and PDF into OFDP*.

Basic Steps for Creating an HTML E-Form

Within this document, we will step you through the process of taking an already fielded form and make it server-ready. We will start with a sample OneForm file named **req04.elf.** It is included with the OneForm Designer Plus Forms Library under the *requistion* folder.

Outlined below are the basic steps for creating a working E-Form, complete with a submit button action, database table associations, a reply message, and form retrieval scripts.

Step 1: Open Sample ELF (A Form with Fillable Form Fields)

Step 2: Add UserID, Submit Button, and Primary Output Key

Step 3: Reset the Tabbing Order

Step 4: Generate HTML Electronic Form and Test

Step 5: Setup Database Table Associations

Step 6: Generate CGI Scripts for E-Form

Step 7: Generate Final HTML E-Form

Step 8: Copy Files to Web Server

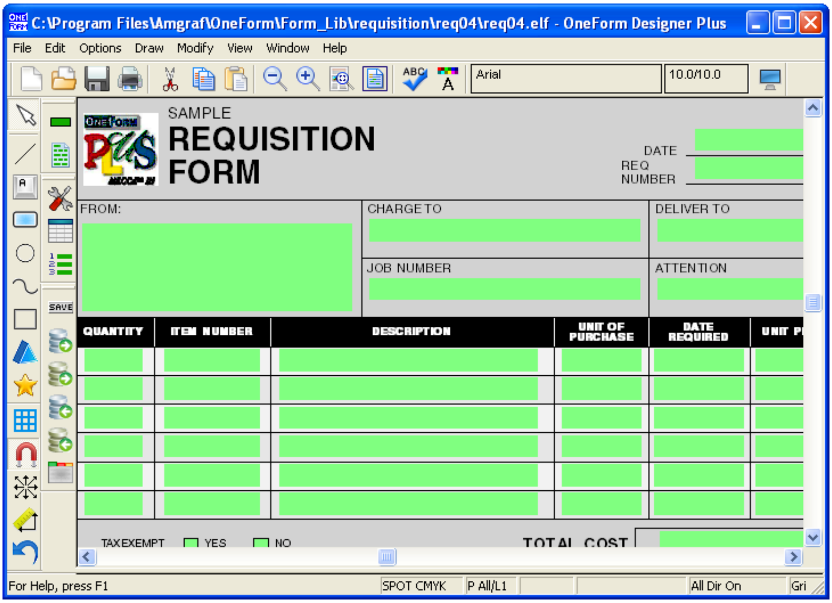
Step 9: Test Database Connectivity

Step 1: 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\requisition\req04

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



SAMPLE
REQUISITION
FORM

DATE
REQ NUMBER

FROM:
CHARGE TO
DELIVER TO
JOB NUMBER
ATTENTION

QUANTITY	ITEM NUMBER	DESCRIPTION	UNIT OF PURCHASE	DATE REQUIRED	UNIT P

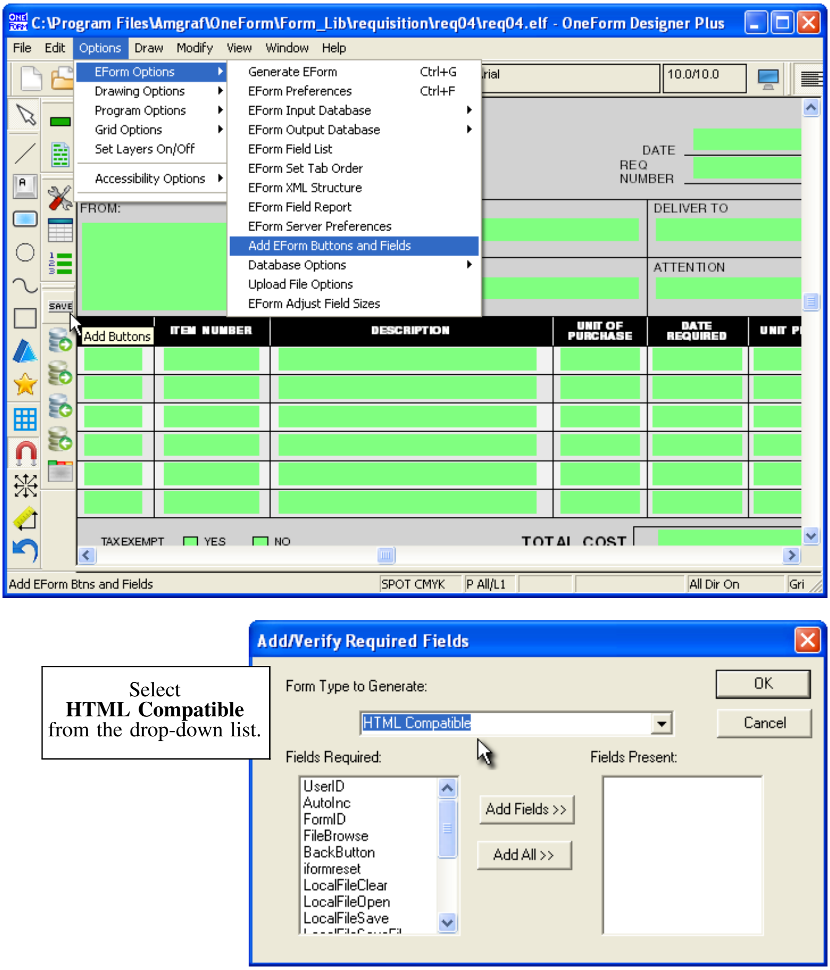
TAX EXEMPT □ YES □ NO
TOTAL COST

Figure 10-1: Our Sample Form

This form is ready to be turned into an HTML Fillable E-Form. We are going to work with it

Step 2: Add Submit Button and Primary Output Key

We begin the process of creating an HTML E-Form by adding the necessary fields. This would be the submit button, the primary output key field, and the form status field. If you want, there is also a User ID field available. Click on the **Add Buttons** tool. Select **HTML Compatible** as the Form Type to Generate.



Form Type to Generate: HTML Compatible

Fields Required:

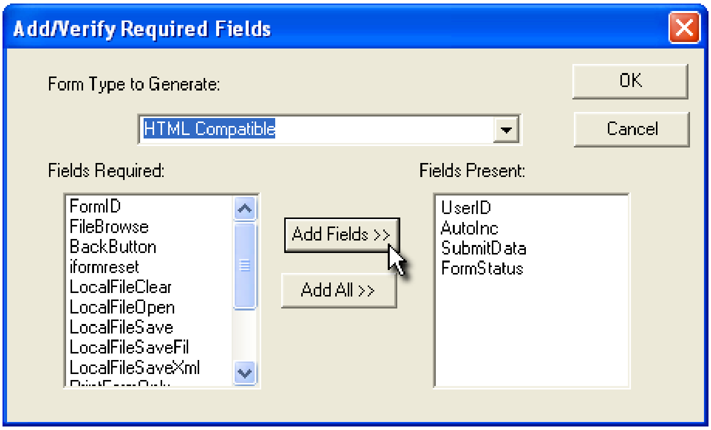
UserID
AutoInc
FormID
FileBrowse
BackButton
iformreset
LocalFileClear
LocalFileOpen
LocalFileSave

Fields Present:

Select
HTML Compatible
from the drop-down list.

Figure 10-2: Add AutoInc, SubmitData, FormStatus, and UserID Fields

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 four fields: **User ID, AutoInc, FormStatus** and **SubmitData.**



Form Type to Generate: HTML Compatible

Fields Required:

FormID
FileBrowse
BackButton
iformreset
LocalFileClear
LocalFileOpen
LocalFileSave
LocalFileSaveFil
LocalFileSaveXml

Fields Present:

UserID
AutoInc
SubmitData
FormStatus

Figure 10-3: Selecting Fields

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

All of the fields found under the HTML Compatible section are described at the end of this chapter. The ones we have chosen are described below:

The UserID field keeps track of who is submitting the data.

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

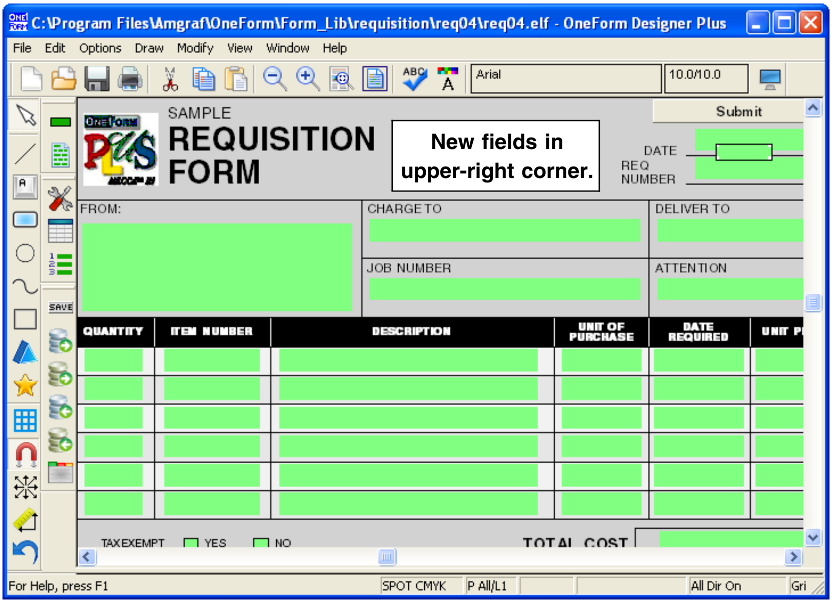
The AutoInc field will be used as our primary output key during the database table connectivity. It will apply a sequential number to each new form record that is submitted for tracking purposes.

The FormStatus field keeps track of the type of Submit. By default it is either a Save or SaveDraft.

All of the fields found under the HTML Compatible section are described at the end of this chapter.

Move Fields to Suitable Location

The software added four fields to the form, with one of them being a button for Submit.



New fields in upper-right corner.

Figure 10-4: The Fields are Automatically Added to the Upper-Right Corner

The Submit Button will be in the top right-hand corner, and the AutoInc will always fall one inch down below it.

The UserID field will be a tiny hidden field at the bottom left of the form. In the View menu, turn on EForm Fields, Hidden Fields if you want to see it.

Since the new visible fields we have added always fall in the right-hand corner, they will need to be moved to a more suitable location. We have moved ours to the center of the top margin.

Step 3: Reset the Tabbing Order

Since you have added a button to the form, it is necessary to establish its tabbing order. Go to the **Set Tab Order** tool. You will notice the new fields have a tabbing 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 the [Submit] button field. This will assign it the next tab order number.

The AutoInc field will keep the tabbing order of 0, since it is a read-only field.

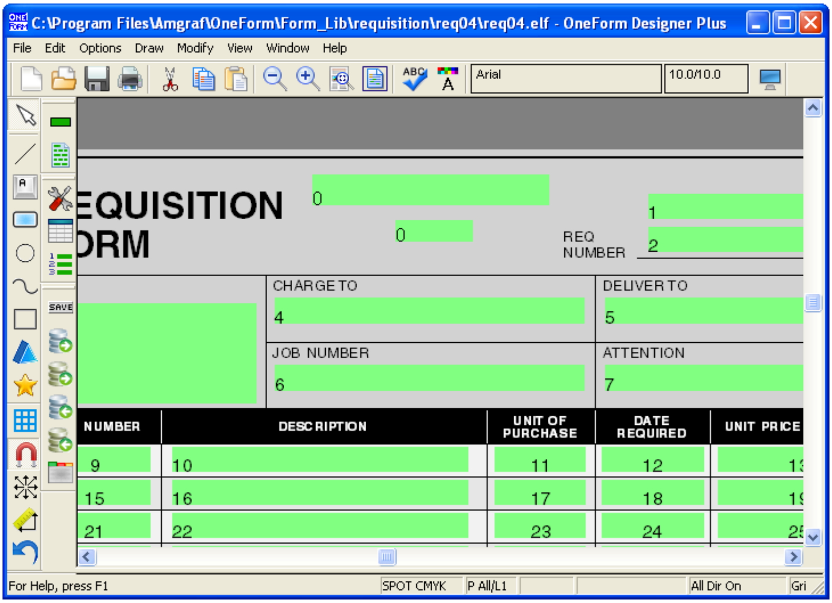


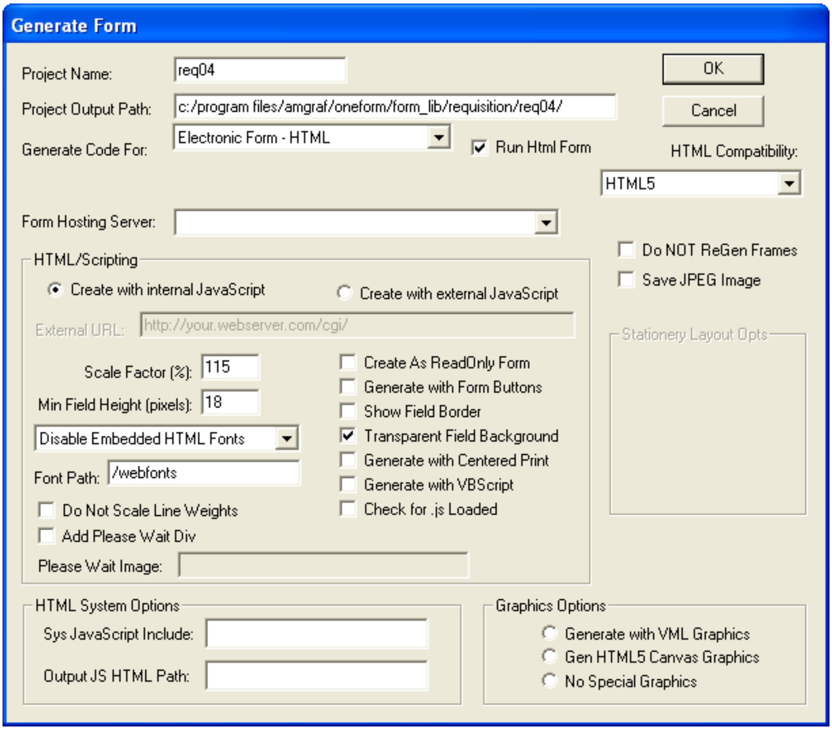
Figure 10-5: Setting Tab Order of the Submit Button

Use the pop-up menu's **Done Setting Tab Order** to save the tab settings.

This would be a good time to also **Save** these changes made to the file.

Step 4: Generate HTML Electronic Form and Test

At this point, the HTML file is ready to be generated. Click on the blue **Generate EForm** tool on the top tool bar. For the Generate Code, select within the drop-down list **Electronic Form - HTML.** Also click on the checkbox for **Run HTML Form** and select **[Okay].**



Generate Form

Project Name: req04

Project Output Path: c:/program files/amgraf/oneform/form_lib/requisition/req04/

Generate Code For: Electronic Form - HTML

☑ Run Html Form

HTML Compatibility: HTML5

Form Hosting Server:

☐ Do NOT ReGen Frames

☐ Save JPEG Image

HTML/Scripting

● Create with internal JavaScript ○ Create with external JavaScript

External URL: http://your.webserver.com/cgi/

Scale Factor (%): 115

Min Field Height (pixels): 18

Disable Embedded HTML Fonts

Font Path: /webfonts

☐ Do Not Scale Line Weights

☐ Add Please Wait Div

Please Wait Image:

☐ Create As ReadOnly Form

☐ Generate with Form Buttons

☐ Show Field Border

☑ Transparent Field Background

☐ Generate with Centered Print

☐ Generate with VBScript

☐ Check for .js Loaded

Stationery Layout Opts

HTML System Options

Sys JavaScript Include:

Output JS HTML Path:

Graphics Options

○ Generate with VML Graphics

○ Gen HTML5 Canvas Graphics

○ No Special Graphics

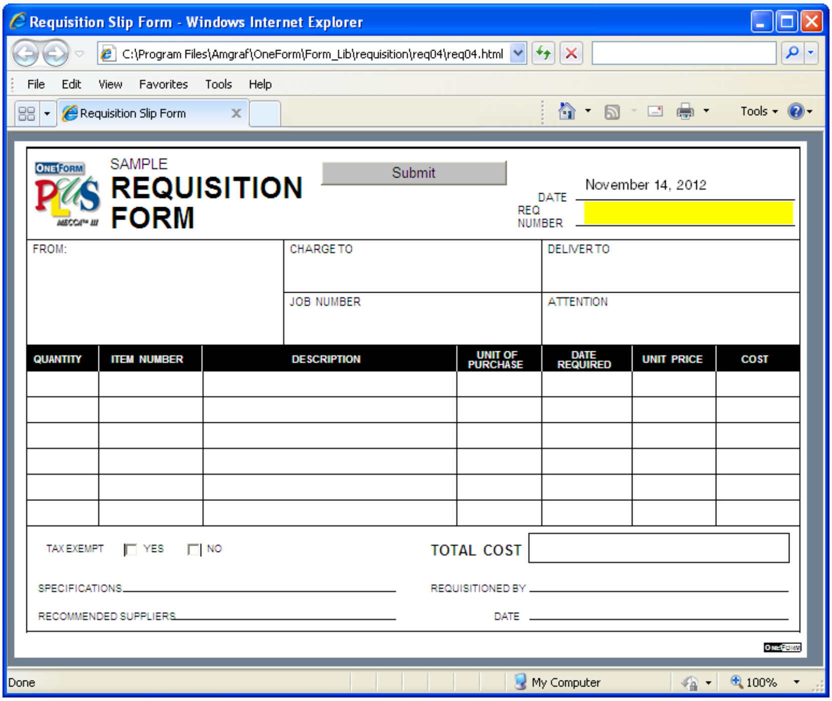
Figure 10-6: Generate E-Form

This option will generate the form into HTML/JavaScript code and open the form as an .html page in your default browser. See *Figure 10-8* on the next page. The HTML file created will pick up the basename of the ELF and be placed into your project folder.

We have toggled off the "Show Field Border" option and toggled on the "Transparent Field Background" option.

Note: The Generate EForm option is also found under the **Options** menu, **EForm Options** and **Generate EForm** menu and has a shortcut key **[Ctrl]-[G].** See page 161, at the end of this chapter, for complete details on the Generate Form dialog options.

Test the HTML E-Form


Requisition Slip Form - Windows Internet Explorer
C:\Program Files\Amgraf\OneForm\Form_Lib\requisition\req04\req04.html
File Edit View Favorites Tools Help
Requisition Slip Form
Tools
Done
My Computer
100%

SAMPLE
REQUISITION FORM

Submit

DATE November 14, 2012
REQ NUMBER

FROM:	CHARGE TO	DELIVER TO
	JOB NUMBER	ATTENTION

QUANTITY	ITEM NUMBER	DESCRIPTION	UNIT OF PURCHASE	DATE REQUIRED	UNIT PRICE	COST

TAX EXEMPT □ YES □ NO

TOTAL COST

SPECIFICATIONS__________ REQUISITIONED BY __________

RECOMMENDED SUPPLIERS__________ DATE __________

Figure 10-7: Testing an E-Form

Test the *req04.html* and make any necessary changes to the ELF file in OFDP. Items to test for are:

- tabbing order
- legibility of data input
- custom calculations
- exclusive checkboxes
- the height, width, and position of the fields
- image display

Closing the browser will take you back to OFDP. Repeat the generate process until you are satisfied with the look and usage of the HTML E-Form.

You will NOT be able to test the database connectivity or [Submit] button until you complete Step 8 and get the project on a web server.

Step 5: Setup Database Table Associations

A unique feature of OFDP is the *New Output DB Table* option. This function prepares the form fields for data collection by assigning them to a database table field of the same name. It also gathers the information necessary for the "Create Table" CGI script. Our example will use a DSN connection to a MySQL database.

Here we will only cover the steps for associating output database tables. These tables will be used to store and retrieve the data submitted from the E-Form. The stored data can be retrieved back into its original E-Form using the "Primary Key" field.

Our Primary Output Key field is the newly added **AutoInc.**

Associating the Output Database Primary Table

Start by clicking the tool for **New Output DB Table.**

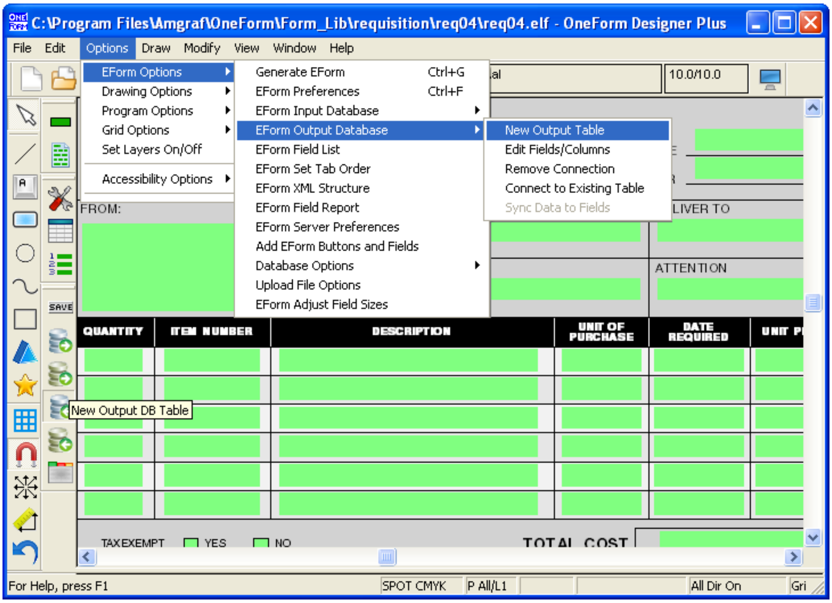
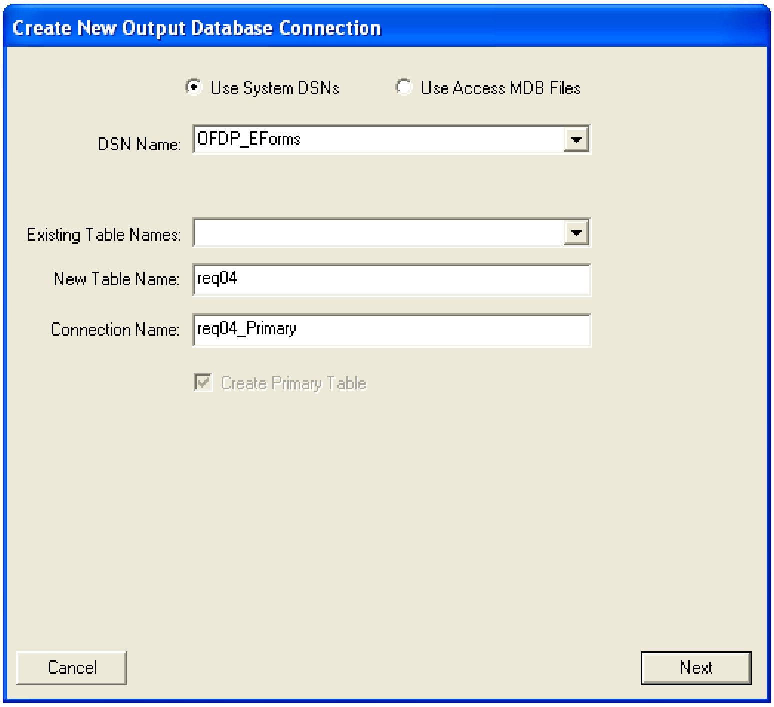


Figure 10-8: New Output DB Table Tool and Menu

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



Create New Output Database Connection

● Use System DSNs ○ Use Access MDB Files

DSN Name: OFDP_EForms

Existing Table Names:

New Table Name: req04

Connection Name: req04_Primary

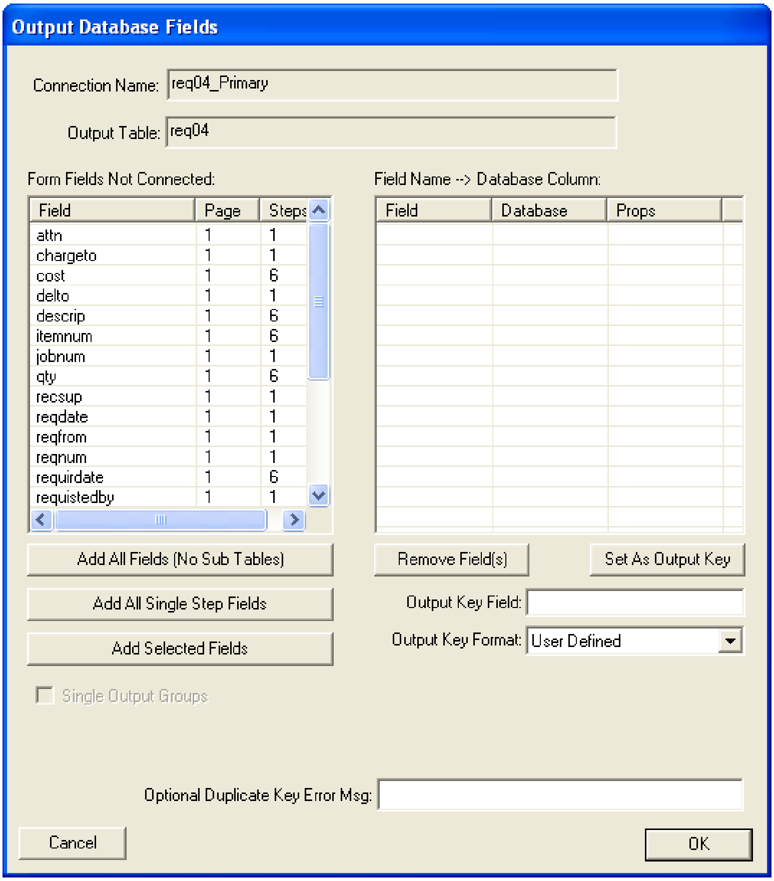
☑ Create Primary Table

Figure 10-09: Locate Your Database

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

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

The Output Database Fields Dialog lists the fields on the E-Form in the left-hand window. The right-hand window will list fields as they become associated to the table and the DB column name that will receive the data.



Output Database Fields

Connection Name: req04_Primary

Output Table: req04

Form Fields Not Connected:

Field	Page	Steps
attn	1	1
chargeto	1	1
cost	1	6
delto	1	1
descrip	1	6
itemnum	1	6
jobnum	1	1
qty	1	6
recsup	1	1
reqdate	1	1
reqfrom	1	1
reqnum	1	1
requirdate	1	6
requistedby	1	1

Field Name --> Database Column:

Field	Database	Props

Output Key Field:

Output Key Format: User Defined

☐ Single Output Groups

Optional Duplicate Key Error Msg:

Figure 10-10: Output Database Fields Dialog

The dialog has three main options:

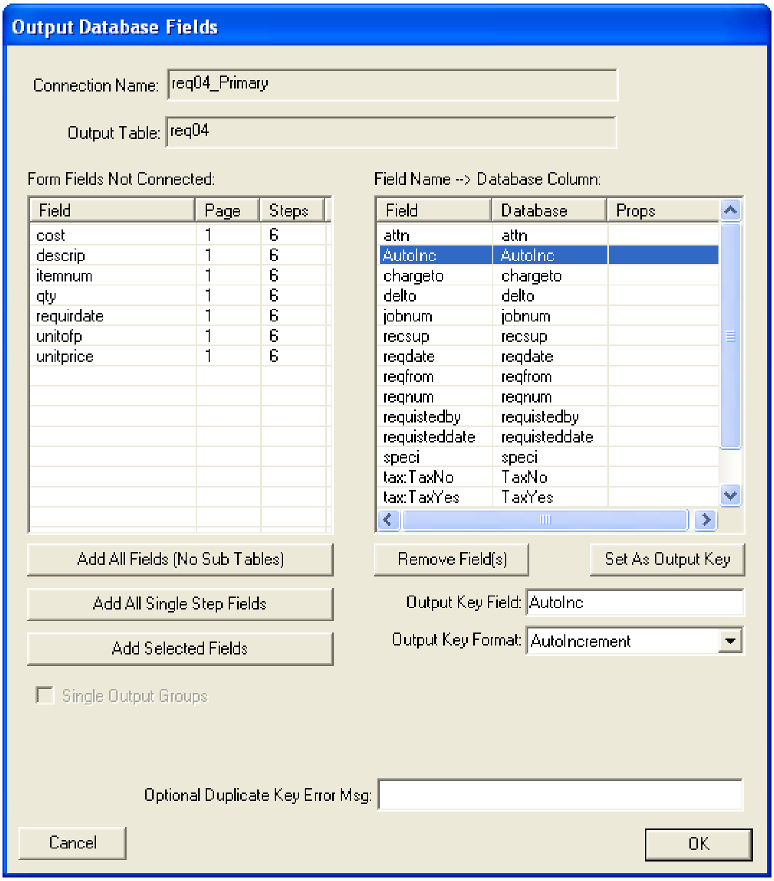
[Add All Fields (No sub-tables)] – This button takes all fields and associates them to the primary output database table. Even stepped fields will be included.

[Add All Single Stepped Fields] – This will collect only the single fields into the Primary table, leaving the stepped fields to be associated later.

[Add Selected Fields] – Use this option to identify one or more fields to add individually. Select them using the standard Windows highlighting techniques. This option will be used to attach additional output database tables which we call "sub-tables".

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 associated later with a sub-table.

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



Output Database Fields

Connection Name: req04_Primary

Output Table: req04

Form Fields Not Connected:

Field	Page	Steps
cost	1	6
descrip	1	6
itemnum	1	6
qty	1	6
requirdate	1	6
unitofp	1	6
unitprice	1	6

Field Name --> Database Column:

Field	Database	Props
attn	attn	
AutoInc	AutoInc	
chargeto	chargeto	
delto	delto	
jobnum	jobnum	
recsup	recsup	
reqdate	reqdate	
reqfrom	reqfrom	
reqnum	reqnum	
requistedby	requistedby	
requisteddate	requisteddate	
speci	speci	
tax:TaxNo	TaxNo	
tax:TaxYes	TaxYes	

Output Key Field: AutoInc

Output Key Format: AutoIncrement

□ Single Output Groups

Optional Duplicate Key Error Msg:

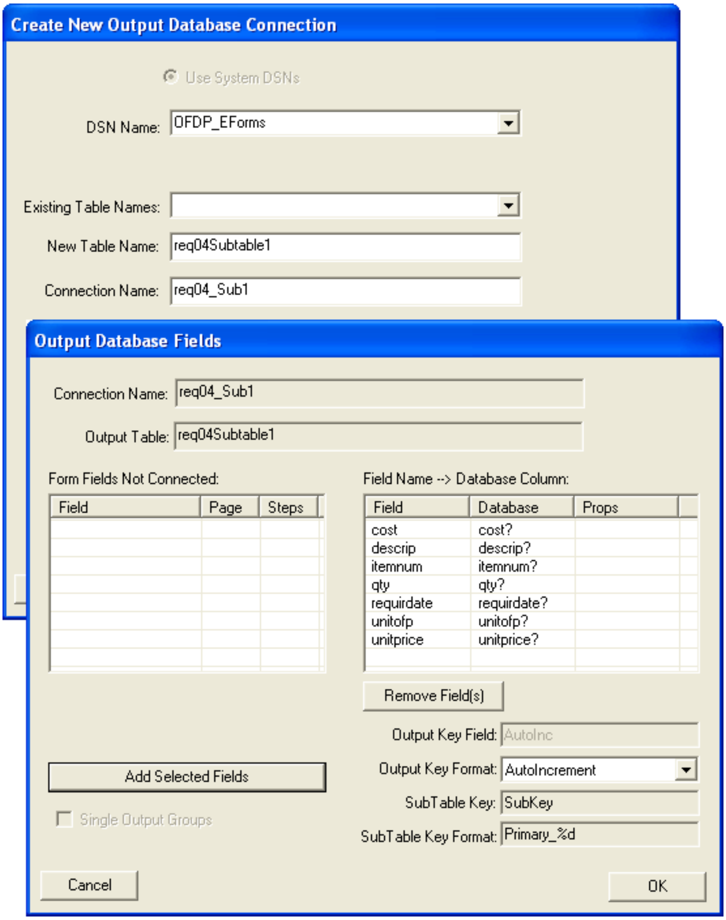
Figure 10-11: The Primary Key is the field named "AutoInc"

Click **[OK]** to close the dialog.

Associating Stepped Fields to Sub-Table

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

Click the tool for **New Output DB Table** again.



Create New Output Database Connection

● Use System DSNs

DSN Name: OFDP_EForms

Existing Table Names:

New Table Name: req04Subtable1

Connection Name: req04_Sub1

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?	
requirdate	requirdate?	
unitofp	unitofp?	
unitprice	unitprice?	

Output Key Field: AutoInc

Output Key Format: AutoIncrement

SubTable Key: SubKey

SubTable Key Format: Primary_%d

☐ Single Output Groups

Figure 10-12: [Add Selected Fields] will Associate Highlighted Fields

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

This time **highlight the stepped fields** within the left-hand window and click **[Add Selected Fields].** This will move them into the right-hand window showing they are associated. Click **[OK]** to save the sub-table.

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 attached; 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 associated; the output sub-table.

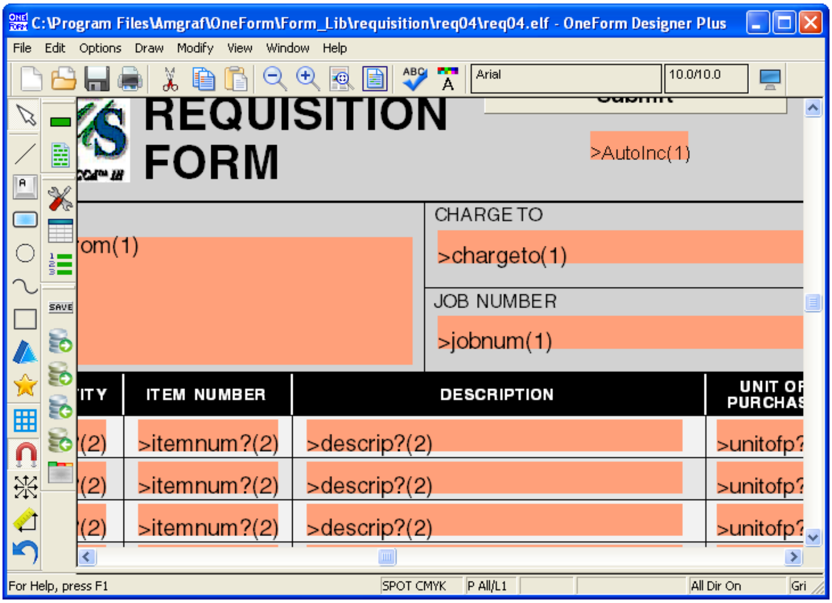


Figure 10-13: Viewing the Database Links

Step 6: Generating the CGI Scripts

The E-Form Server Preferences allow you to specify the actions of the CGI Scripts that run on the web site server to save, search, list, and retrieve the E-Form. The scripts also manage the creation of the server data table described in the previous Output Database dialogs. The objective is to generate all the server scripts to host the HTML form, and to connect the fill fields to a server output database.

E-Form Server Preferences

Start by clicking the **E-Form Server Preferences** tool on the E-Form Server toolbar.

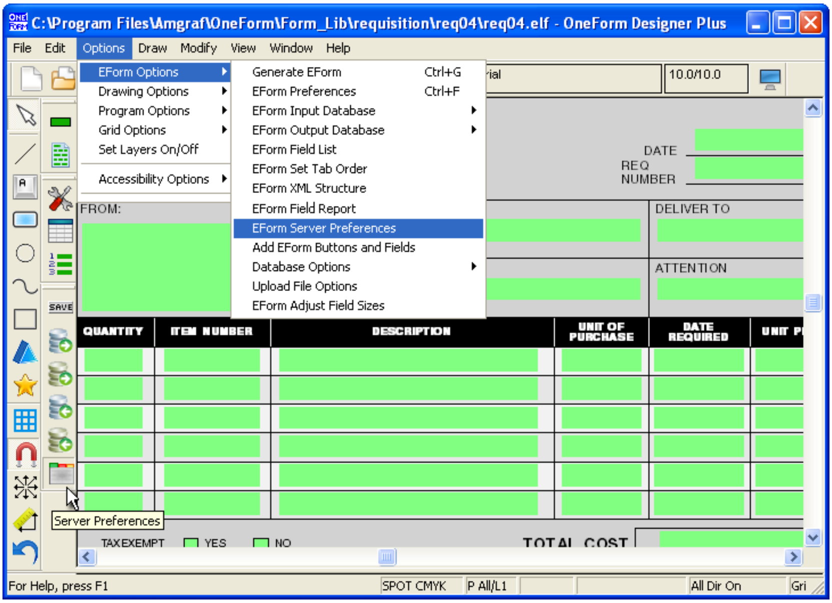
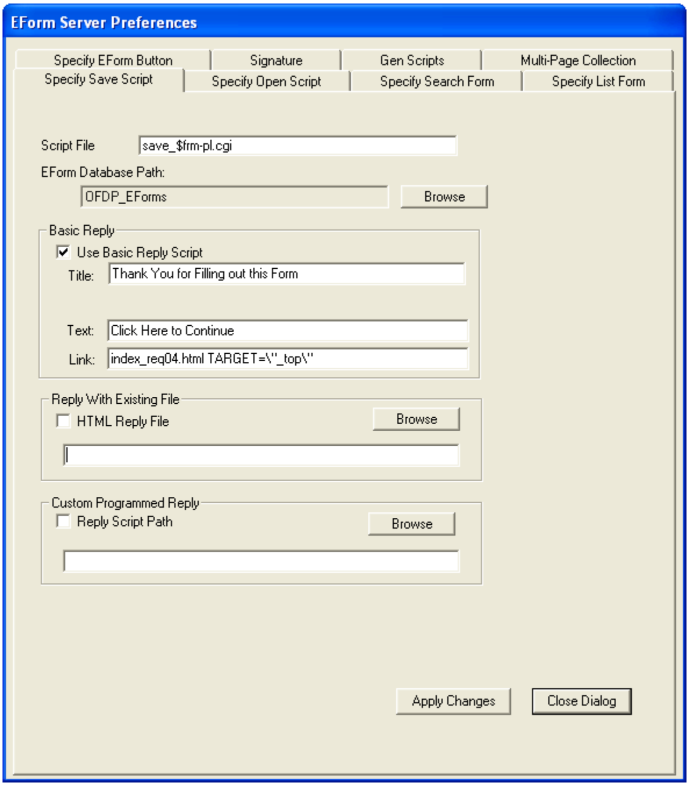


Figure 10-14: Go to EForm Server Preferences

The EForm Preferences dialog consists of eight tabs. Everything is already set for normal use. The one item that must be set is the Submit button. To give you an idea of what the tabs do, we will step you through making a few changes.



EForm Server Preferences

Script File: save_$frm-pl.cgi

EForm Database Path: OFDP_EForms

Basic Reply

☑ Use Basic Reply Script

Title: Thank You for Filling out this Form

Text: Click Here to Continue

Link: index_req04.html TARGET=\"_top\"

Reply With Existing File

□ HTML Reply File

Custom Programmed Reply

□ Reply Script Path

Figure 10-15: The Save Script Tab

The *Specify Save Script* and *Specify Open Script* dialogs are automatically filled out for you.

The "E-Form Database Path" is the location that the Save script will place the data based upon the currently selected system DSN configuration.

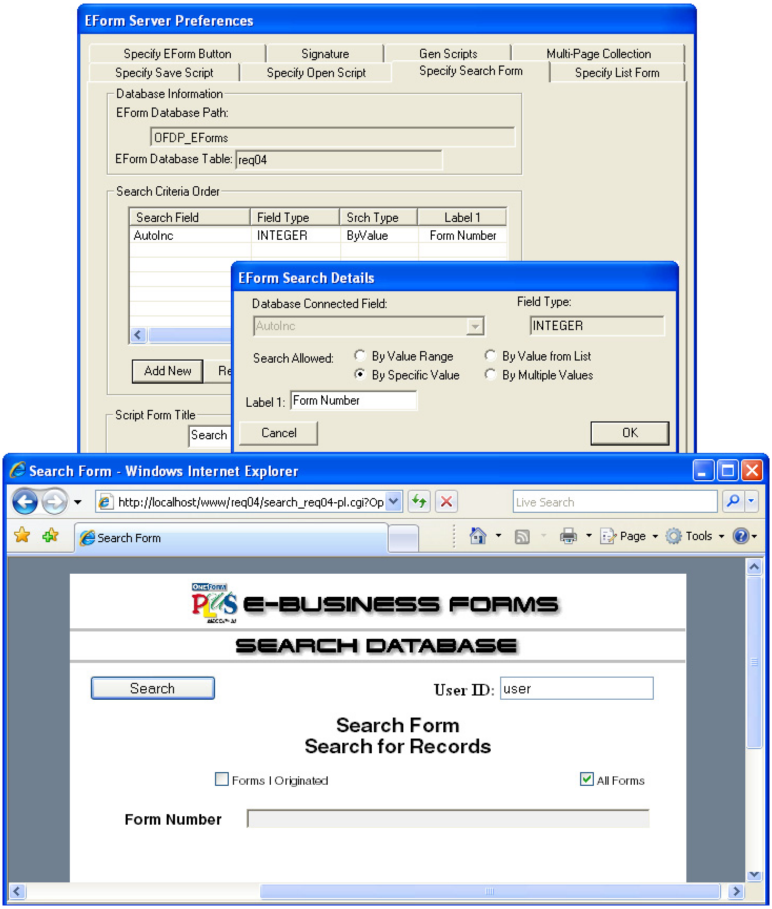
The reply message is the dialog sent back once the user submits their information. The message lets the user know that their data has been received and often contains a link to another page. It automatically responds with a Basic Reply Script.

The *Specify Open Script* tab references our previously set Output Primary Key which is AutoInc.

Specify the Search and List Criteria

Each time a user fills out an E-Form and saves it on a web server, a record is written in a data table. To retrieve a filled-in E-Form, it is necessary to use a search form to locate the data. The **Specify Search Form** tab builds this search form using E-Form field names specified. The **Specify List Form** tab indicates how the search results are displayed. Both tabs work in the same manner.

Let’s make a change to the search form. Click on the **Specify Search Form** tab. Here we want to change the label for the *AutoInc* field. Highlight **AutoInc** and click **[Modify].**



EForm Database Path: OFDP_EForms

EForm Database Table: req04

Search Field	Field Type	Srch Type	Label 1
AutoInc	INTEGER	ByValue	Form Number

Search

Database Connected Field: AutoInc

Field Type: INTEGER

Search Allowed: ○ By Value Range ● By Specific Value ○ By Value from List ○ By Multiple Values

Label 1: Form Number

http://localhost/www/req04/search_req04-pl.cgi?Op

User ID: user

Search Form
Search for Records

□ Forms I Originated ☑ All Forms

Form Number

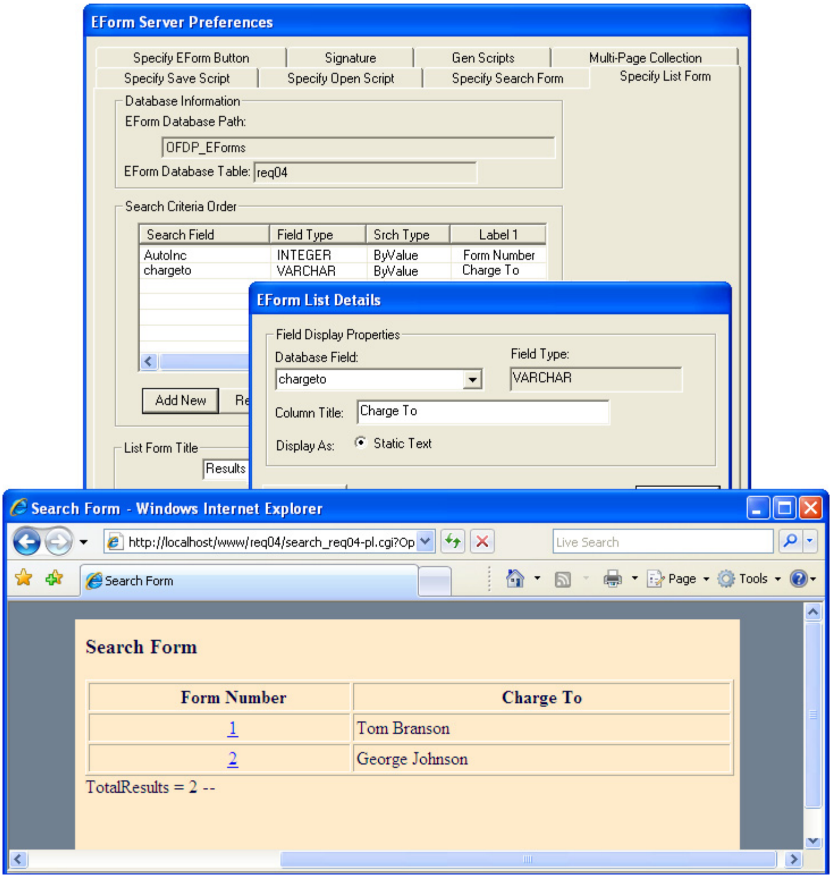
Figure 10-16: Modify "AutoInc" in the Search Script

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

The results list will also use AutoInc, so click on **[Copy to List].** Clicking **[Apply]** will save your changes to this tab.

The results list will have the AutoInc field. The default is to have the primary key field as a link so that you can retrieve a saved E-Form.

Click **[Add New]** and locate the field *chargeto* from the database list. Title it **"Charge To".** Click **[OK]** to save your change.



Database Information

EForm Database Path: OFDP_EForms

EForm Database Table: req04

Search Criteria Order

Search Field	Field Type	Srch Type	Label 1
AutoInc	INTEGER	ByValue	Form Number
chargeto	VARCHAR	ByValue	Charge To

List Form Title: Results

Field Display Properties

Database Field: chargeto

Field Type: VARCHAR

Column Title: Charge To

Display As: ● Static Text

http://localhost/www/req04/search_req04-pl.cgi?Op

Search Form

Form Number	Charge To
1	Tom Branson
2	George Johnson

TotalResults = 2 --

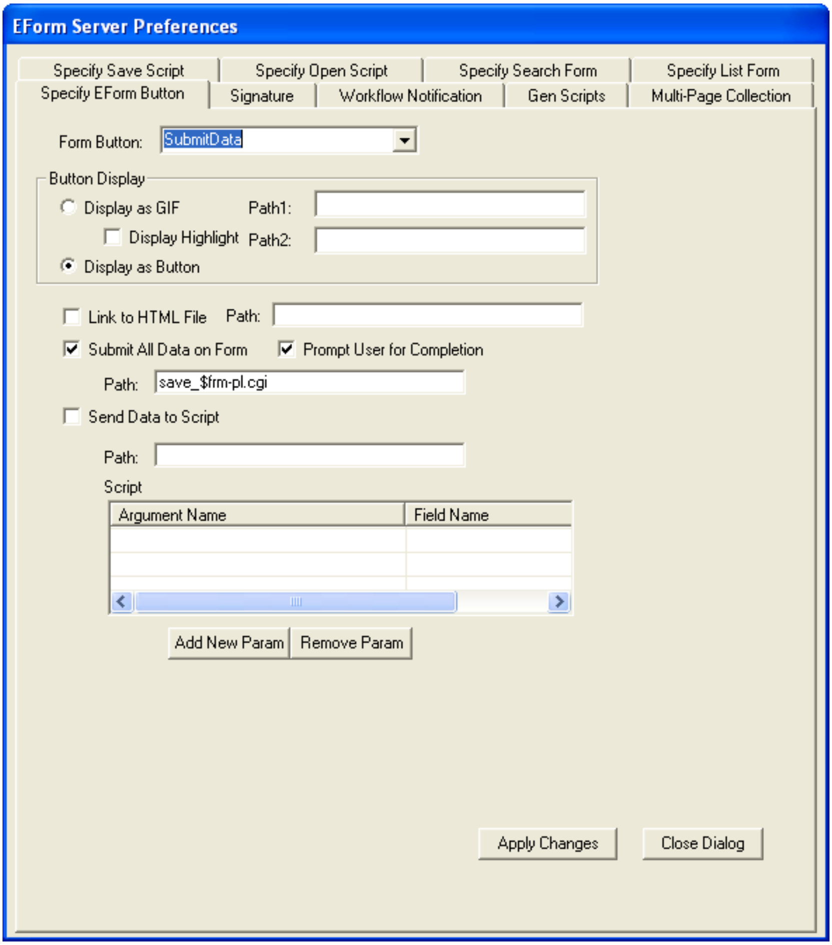
Figure 10-17: Specifying the Results List Script

Rename the Title of the List **"Results of Search for Requisition Form".** Clicking **[Apply]** will save your changes to this tab.

Set the Submit Data Button

Go to the **Specify EForm Button** tab. Be sure to select the **SubmitData** button from the Form Button drop-down list. Then indicate it to **Submit All Data on Form** by clicking the checkbox as shown in the following illustration.

When **Prompt User for Completion** is checked, a message will appear (to the user) before submitting any data. Click **[Apply]** to save your settings.



Form Button: SubmitData

Button Display

○ Display as GIF Path1:

□ Display Highlight Path2:

● Display as Button

□ Link to HTML File Path:

☑ Submit All Data on Form ☑ Prompt User for Completion

Path: save_$frm-pl.cgi

□ Send Data to Script

Path:

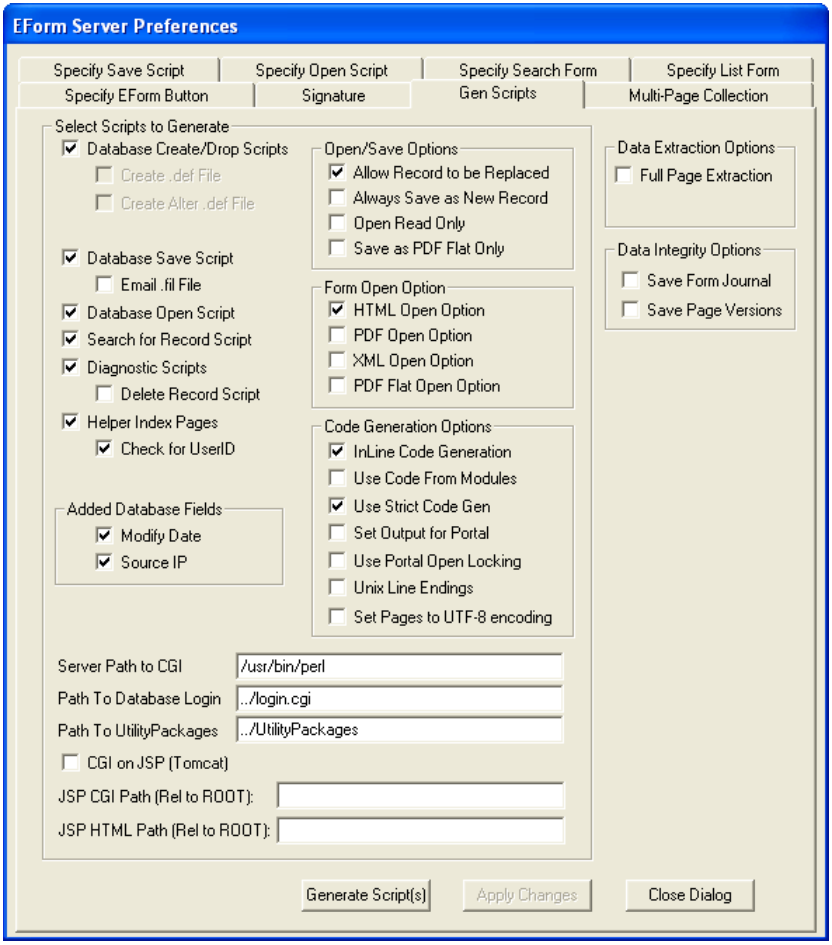
Script

Argument Name	Field Name

Figure 10-18: Submit Data Button Settings

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



Select Scripts to Generate
- ☑ Database Create/Drop Scripts
  - ☐ Create .def File
  - ☐ Create Alter .def File
- ☑ Database Save Script
  - ☐ Email .fil File
- ☑ Database Open Script
- ☑ Search for Record Script
- ☑ Diagnostic Scripts
  - ☐ Delete Record Script
- ☑ Helper Index Pages
  - ☑ Check for UserID

Added Database Fields
- ☑ Modify Date
- ☑ Source IP

Open/Save Options
- ☑ Allow Record to be Replaced
- ☐ Always Save as New Record
- ☐ Open Read Only
- ☐ Save as PDF Flat Only

Form Open Option
- ☑ HTML Open Option
- ☐ PDF Open Option
- ☐ XML Open Option
- ☐ PDF Flat Open Option

Code Generation Options
- ☑ InLine Code Generation
- ☐ Use Code From Modules
- ☑ Use Strict Code Gen
- ☐ Set Output for Portal
- ☐ Use Portal Open Locking
- ☐ Unix Line Endings
- ☐ Set Pages to UTF-8 encoding

Data Extraction Options
- ☐ Full Page Extraction

Data Integrity Options
- ☐ Save Form Journal
- ☐ Save Page Versions

Server Path to CGI: /usr/bin/perl

Path To Database Login: ../login.cgi

Path To UtilityPackages: ../UtilityPackages

☐ CGI on JSP (Tomcat)

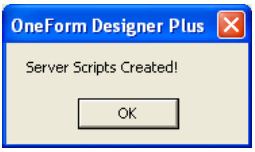
JSP CGI Path (Rel to ROOT):

JSP HTML Path (Rel to ROOT):

Figure 10-19: The Generate Scripts Tab

Click the **[Generate Scripts]** button to generate the server scripts as shown in the following illustration.

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

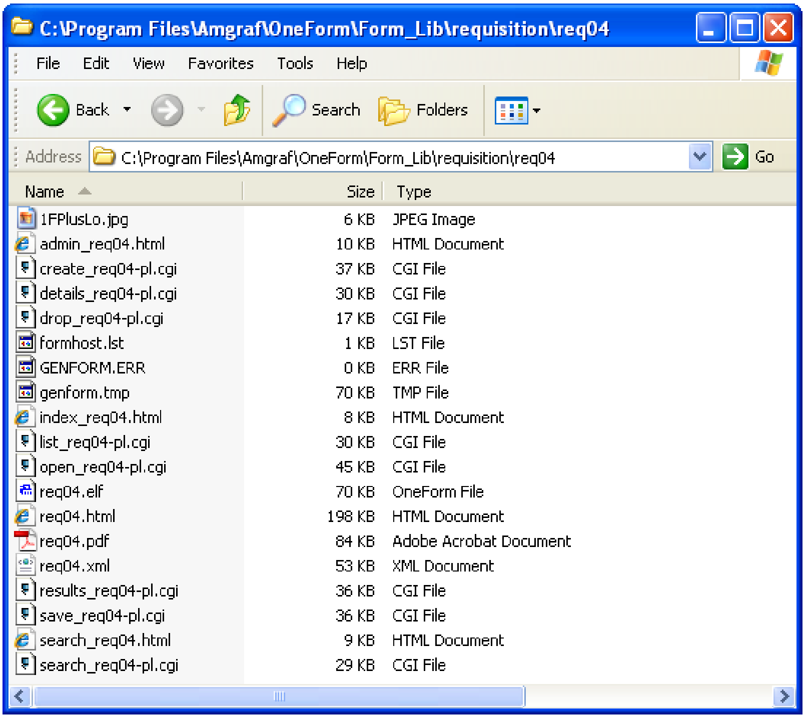


OneForm Designer Plus

Server Scripts Created!

Figure 10-20: Server Scripts Created

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



C:\Program Files\Amgraf\OneForm\Form_Lib\requisition\req04

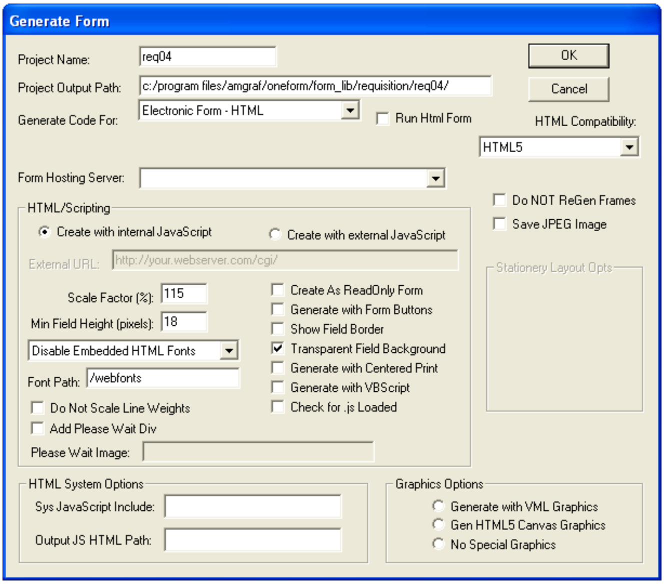
C:\Program Files\Amgraf\OneForm\Form_Lib\requisition\req04

Name	Size	Type
1FPlusLo.jpg	6 KB	JPEG Image
admin_req04.html	10 KB	HTML Document
create_req04-pl.cgi	37 KB	CGI File
details_req04-pl.cgi	30 KB	CGI File
drop_req04-pl.cgi	17 KB	CGI File
formhost.lst	1 KB	LST File
GENFORM.ERR	0 KB	ERR File
genform.tmp	70 KB	TMP File
index_req04.html	8 KB	HTML Document
list_req04-pl.cgi	30 KB	CGI File
open_req04-pl.cgi	45 KB	CGI File
req04.elf	70 KB	OneForm File
req04.html	198 KB	HTML Document
req04.pdf	84 KB	Adobe Acrobat Document
req04.xml	53 KB	XML Document
results_req04-pl.cgi	36 KB	CGI File
save_req04-pl.cgi	36 KB	CGI File
search_req04.html	9 KB	HTML Document
search_req04-pl.cgi	29 KB	CGI File

Figure 10-21: Files are Automatically Created by [Generate Scripts]

Step 7: Generate the Final E-Form HTML

At this point, the actual HTML E-Form is ready to be generated again. Go to the **Options** menu, **EForm Options** and **Generate EForm**. Choose **"Generate Code for" Electronic Form - HTML** and click **[Okay].**



Generate Form

Project Name: req04

Project Output Path: c:/program files/amgraf/oneform/form_lib/requisition/req04/

Generate Code For: Electronic Form - HTML

□ Run Html Form

HTML Compatibility: HTML5

Form Hosting Server:

□ Do NOT ReGen Frames

□ Save JPEG Image

Stationery Layout Opts

HTML/Scripting

● Create with internal JavaScript ○ Create with external JavaScript

External URL: http://your.webserver.com/cgi/

Scale Factor (%): 115

Min Field Height (pixels): 18

Disable Embedded HTML Fonts

Font Path: /webfonts

□ Do Not Scale Line Weights

□ Add Please Wait Div

Please Wait Image:

□ Create As ReadOnly Form

□ Generate with Form Buttons

□ Show Field Border

☑ Transparent Field Background

□ Generate with Centered Print

□ Generate with VBScript

□ Check for .js Loaded

HTML System Options

Sys JavaScript Include:

Output JS HTML Path:

Graphics Options

○ Generate with VML Graphics

○ Gen HTML5 Canvas Graphics

○ No Special Graphics

Figure 10-22: Generate Final HTML E-Form

This will overwrite the existing HTML file with one that has CGI links for the [Submit] button. You are now ready to copy the files to the server and test the database connectivity and [Submit] button.

Step 8: 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 to the appropriate location as directed by your Webmaster.

Transferring Files with FTP

If you are using FTP, all *.html* and *.cgi* files generated in this directory MUST be sent as ASCII. They will not execute otherwise. Any images specific to this project (*.gif* and *.jpg* files) MUST be sent as BINARY.

Make CGI Scripts Executable

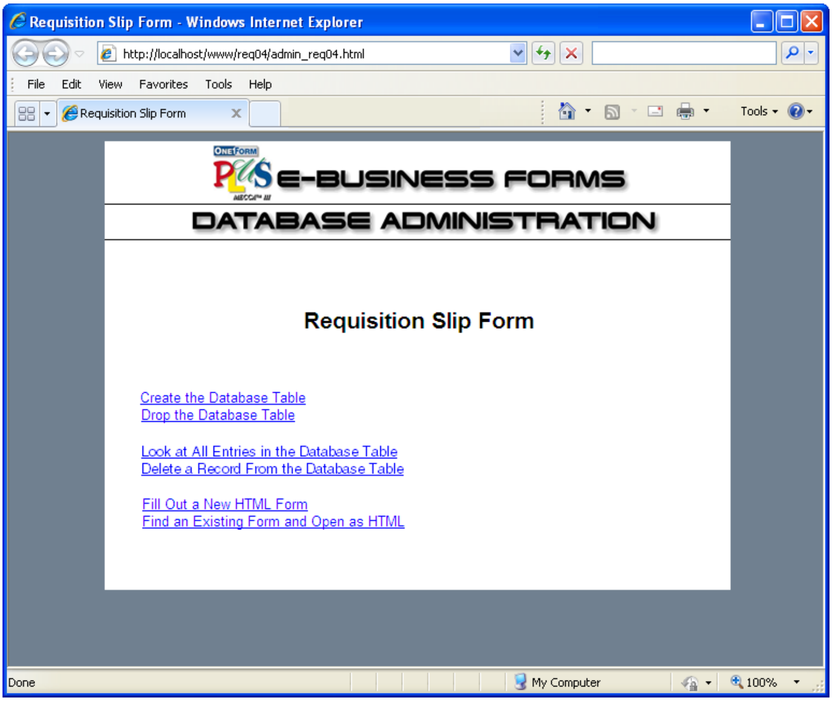
All CGI script files must have executable permissions on the server. You may need assistance from your Webmaster to do this.

Step 9: Test Database Connectivity and E-Form

To start testing you need to locate the administration HTML file (see *Figure 10-23),* for your project (admin_project.html) and launch it from your browser.

The URL will be similar to the following:

http://servername/req04/admin_req04.html



Requisition Slip Form - Windows Internet Explorer
http://localhost/www/req04/admin_req04.html
File Edit View Favorites Tools Help
Requisition Slip Form
Tools
Done
My Computer
100%

E-BUSINESS FORMS

DATABASE ADMINISTRATION

Requisition Slip Form

Create the Database Table
Drop the Database Table

Look at All Entries in the Database Table
Delete a Record From the Database Table

Fill Out a New HTML Form
Find an Existing Form and Open as HTML

Figure 10-23: Click the "Create Database Table" Link

If the UserID field is required, type in the ID necessary to modify database tables. By default "Administrator" is our standard UserID.

To create the database tables, select the **Create Database Table** link. A message will appear stating that the tables have been created successfully. (Your Webmaster can assist you in having database resources.)

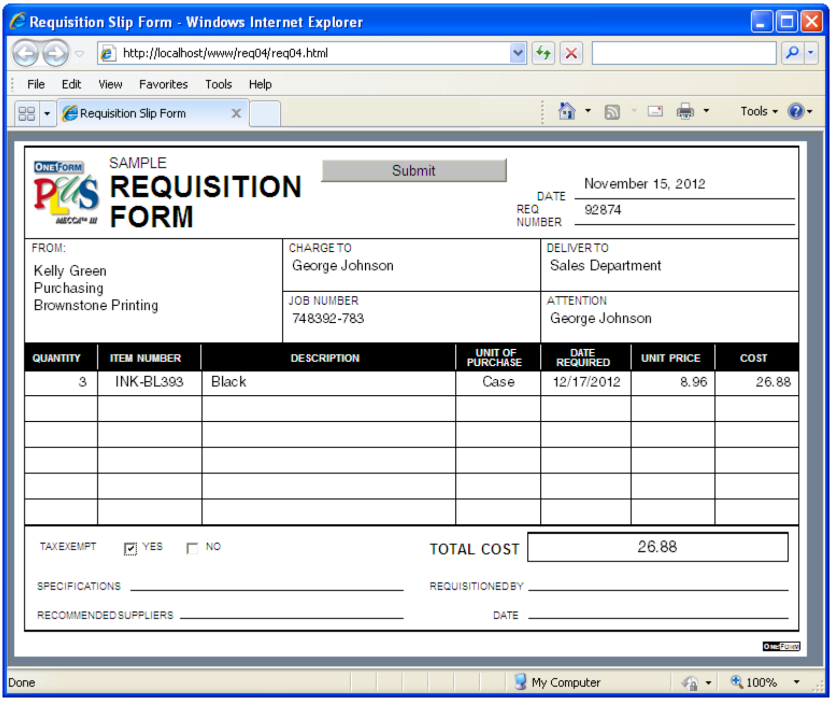
You can also remove a database table and start over if you find it needs to be replaced during testing, through the "Drop Database" link. Be sure to only use this during testing, and not on a table being used.

If you are experiencing problems at this stage refer to the *E-Forms Server Setup Reference Manual.*

Run and Test the E-Form

Test the E-Form through the **admin_req04.html** file. With it you can open the new E-Form and submit data, and then examine the contents of the database table.

Start testing the E-Form by clicking the link **"Fill Out New HTML Form"** and then inputting data into the E-Form and clicking the [Submit] button. If corrections need to be made, make the changes within OneForm Designer Plus and then put the updated files back on the server and test again.


Requisition Slip Form - Windows Internet Explorer
http://localhost/www/req04/req04.html
File Edit View Favorites Tools Help
Requisition Slip Form
Tools
Submit
Done
My Computer
100%

SAMPLE
REQUISITION FORM

DATE November 15, 2012
REQ NUMBER 92874

FROM:
Kelly Green
Purchasing
Brownstone Printing

CHARGE TO George Johnson

JOB NUMBER 748392-783

DELIVER TO Sales Department

ATTENTION George Johnson

QUANTITY	ITEM NUMBER	DESCRIPTION	UNIT OF PURCHASE	DATE REQUIRED	UNIT PRICE	COST
3	INK-BL393	Black	Case	12/17/2012	8.96	26.88

TAX EXEMPT ☑ YES □ NO

TOTAL COST 26.88

SPECIFICATIONS ______

REQUISITIONED BY ______

RECOMMENDED SUPPLIERS ______

DATE ______

Figure 10-24: Input Data and Click [Submit]

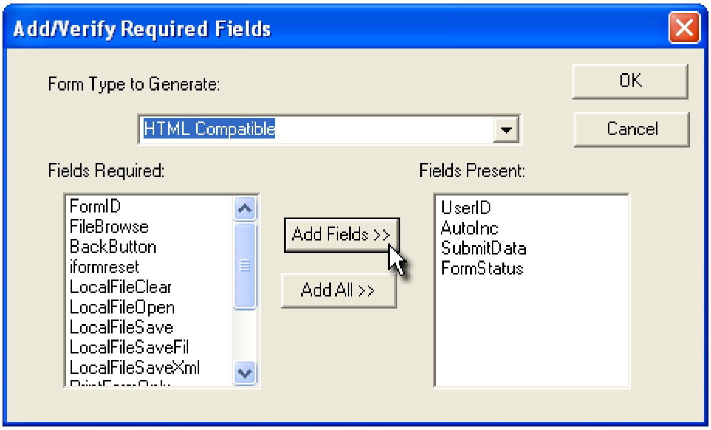
Also, click on **"Find an Existing Form"** and use the Search Form and Results of Search. This was unavailable when testing locally, so it needs to be tested at this time.

If corrections need to be made, make the changes within OneForm Designer Plus. Whenever changes are made, new files will need to be generated and copied back to the server for testing.

Add E-Form Buttons and Fields

Under **Options, EForm Options, Add EForm Buttons and Fields** are not only the required fields for E-Forms (Primary Key, SubmitData, and FormStatus), but some other useful fields. Most deal with submitting to a database, but there are a few that come in handy for general use.

Select a field then click **[Add Fields] [OK].** This will place the field on your form in the top right-hand corner.



Form Type to Generate: HTML Compatible

Fields Required:
FormID
FileBrowse
BackButton
iformreset
LocalFileClear
LocalFileOpen
LocalFileSave
LocalFileSaveFil
LocalFileSaveXml

Fields Present:
UserID
AutoInc
SubmitData
FormStatus

Figure 10-25: HTML Compatible Required Fields

AutoInc: This field is used for a Primary Key field in a data table and is for automatic incrementing each submitted record starting with 1. It is a numeric field with the properties of Integer Only with No Separators, 11 Digits. In the Output DB Table dialog, when selecting "AutoInc" as the Output Key Field, also select "AutoIncrement" as the Output Key Format.

If you are using a MYSQL database, it will have an auto-increment property that controls the sequential numbering. Each subsequent submit will query the existing table and make a new record incrementing the AutoInc value by 1.

With SQL, this action is identical, except it queries a specific "sequence number table" that holds a list of all OFDP tables and their last submitted sequence number.

BackButton: Will back up to the previous HTML page.

FileBrowse: This invokes an action of browsing for a file and keeping track of the file name selected.

FormID: This field is used for a Primary Key field in a data table. It generates a unique number (for example: 58a7c92e7b1541fc9222146575e51ebc), to identify each record, no number being repeated. In the Output DB Table dialog, when selecting "FormID" as the Output Key Field, also select "MD5-UserID-TOD" as the Output Key Format.

FormStatus: This is a status number that can be used to keep track of the form as it goes through a workflow. It is required for forms submitting to data tables.

IFormReset: This button will clear data out of the form.

LocalFile Buttons: (For Local E-Forms Only) In *Chapter 11* we describe a way to use an HTML to capture data and save it to your local harddrive as XML. These buttons are used on those HTML E-Forms to Open and Save the data.

PortalActiveDocument: This is for E-Forms being pushed up to an Amgraf E-Forms Portal.

PrintFormOnly: Will print the form without data.

SaveDraft: Saving with this button will change the FormStatus to a draft.

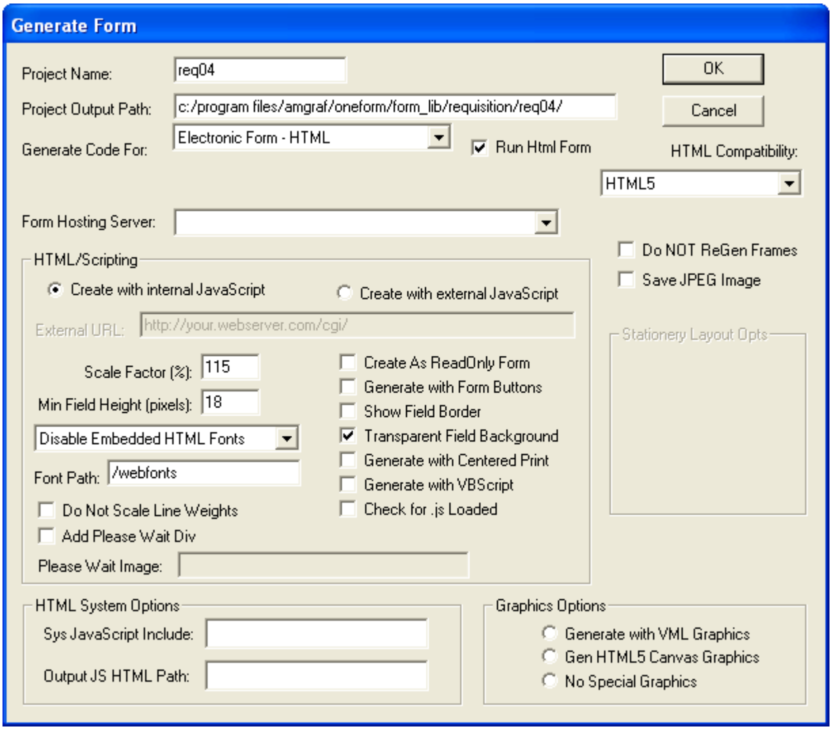
StatesUS: Will provide you with a ComboBox holding a drop-down list of U.S. States.

SubmitData: This is a required field on all forms and is used to invoke the save CGI script for submitting the data to the data table.

UserID: This field keeps track of the User filling out the form.

The Generate E-Form Dialog

The Generate Form dialog is your control over which language type to create from the ELF. Here we will only be discussing generating code for **Electronic Form - HTML.**



Project Name: req04

Project Output Path: c:/program files/amgraf/oneform/form_lib/requisition/req04/

Generate Code For: Electronic Form - HTML

☑ Run Html Form

HTML Compatibility: HTML5

Form Hosting Server:

□ Do NOT ReGen Frames

□ Save JPEG Image

HTML/Scripting

● Create with internal JavaScript ○ Create with external JavaScript

External URL: http://your.webserver.com/cgi/

Scale Factor (%): 115

Min Field Height (pixels): 18

Disable Embedded HTML Fonts

Font Path: /webfonts

□ Do Not Scale Line Weights

□ Add Please Wait Div

Please Wait Image:

□ Create As ReadOnly Form

□ Generate with Form Buttons

□ Show Field Border

☑ Transparent Field Background

□ Generate with Centered Print

□ Generate with VBScript

□ Check for .js Loaded

Stationery Layout Opts

HTML System Options

Sys JavaScript Include:

Output JS HTML Path:

Graphics Options

○ Generate with VML Graphics

○ Gen HTML5 Canvas Graphics

○ No Special Graphics

Figure 10-26: Generate Form Dialog

Project Name is the name all created files will follow. By default, it picks up the base file name of your project file. Project names must start with an alpha character, contain no punctuation or spaces, and be at least three characters.

Project Output Path is the folder location you would like the generated files to be placed. The default is the same folder of your source ELF.

Generate Code For allows you to select the appropriate language needed for your form project. For web-server based HTML forms displayed in a web browser, choose "Electronic Form - HTML".

Run E-Form allows you to immediately preview your HTML file. After selecting [OK], the HTML code will be generated and the web browser will open displaying the file. At this time it is ready for limited testing. Full testing of HTML can only be accomplished by deploying your project to a web server.

HTML Compatibility: This option by default chooses "Old Rendering Compatible" which is to generate OFDP HTML the same as prior to Version 4.7.0.9. This version now has options "W3C 4.01 Loose and Strict" to generate HTML code following the W3C guidelines. The "HTML5" option is necessary for e-forms used on mobile devices.

Server Xfer (Transfer) allows for the direct transfer of files generated to a predetermined location on an existing web server. Use of this option requires previous setup via the Options menu, Internet Settings, and FTP Server Settings. This will eliminate the need at a later time to manually use a FTP program to transfer files to your web server.

HTML/Scripting holds options which control the look of the HTML file. Scale Factor, Field Border, and Transparent Field Background are probably the most commonly used options which can make a big difference in the look of the form. Below are descriptions of each option.

Internal/External JavaScript – The default "Create with Internal JavaScript" is to include all JavaScript in each HTML E-Form when generated. If you want smaller files, you can opt to externalize the basic JavaScript coding into a file named "JSBasic.js". When you choose, "Create with External JavaScript", then set the URL Path for the external JavaScript file. The JSBasic.js file will need to be at that location on the web server.

Scale Factor allows you to proportionally enlarge or reduce the display size of your HTML form within the web browser without physically changing your drawing dimensions. A value between 50% and 250% is allowed.

Min. Field Height (pixels) will allow you to maintain a minimum field size within HTML form displays. A value between 10 and 50 pixels is allowed.

Disable/Enable Fonts allows users to control font displays within the end users web browser. By default "Disable Embedded HTML Fonts" is selected. This restricts fonts used within the browser to Arial, Helvetica, and Times.

Enable Fonts and Designated Font Path – In order to use additional fonts within an HTML form and ensure the end user will be able to view them, you must use the options of either "Enable Bitstream PFR Fonts" or "Enable Microsoft WEFT Fonts". Prior to enabling these font technologies you must create downloadable font files with the respective manufacturers. For more information about Bitstream PFR fonts go to www.bitstream.com. For Microsoft WEFT fonts visit http://www.microsoft.com/typography/web/embedding/weft/ Both of these options require that you designate where the fonts reside on the web server. The Font Path is from the document root.

Create as Read Only Form generates the current HTML form with all fields set with read-only properties.

Generate with Form Buttons – It allows buttons to be available on Read-Only E-Forms that would otherwise be generated without the buttons. Only applies to Read-Only E-Forms.

Show Field Border creates the HTML form with 3D style border frames around each form field drawn.

Transparent Field Background allows all form fields to be created without any interior fill. Objects located behind or inside fields will be visible if selected.

Generate with Centered Print – Toggle this on if you want the HTML E-Form to print centered in the page. The default is an upper-left corner orientation.

Generate with VBScript allows HTML projects to utilize the same custom code the E-Forms use, Vsual Basic Scripting. If you already have it working as an E-Form, then this option will use that custom code within the HTML.

Optimize Scripting Code reduces the amount of scripting included within the HTML project. Items such as standard Field Mask options, which are not used within the project, will not be embedded within the projects HTML code.

Check for .js Loaded If you have JavaScript files being referenced in the Form Preferences, you can have an alert if they are not found.

Add Please Wait Div or Image: If your project is many pages and takes a while to initially load, you can display a "Please wait" message during loading. Just toggle on the Div for the default, or create your own message as an image.

HTML System Options allow custom code to be included into the HTML file.

Sys JavaScript Include – If you want an externalized JavaScript file to be included into the HTML file, input the path and file name for the external JavaScript file to include.

Output JS HTML Path – Set the path for HTML file. If not set it will use the current directory.

HTML Graphic Options allow other graphic elements to be included into the file.

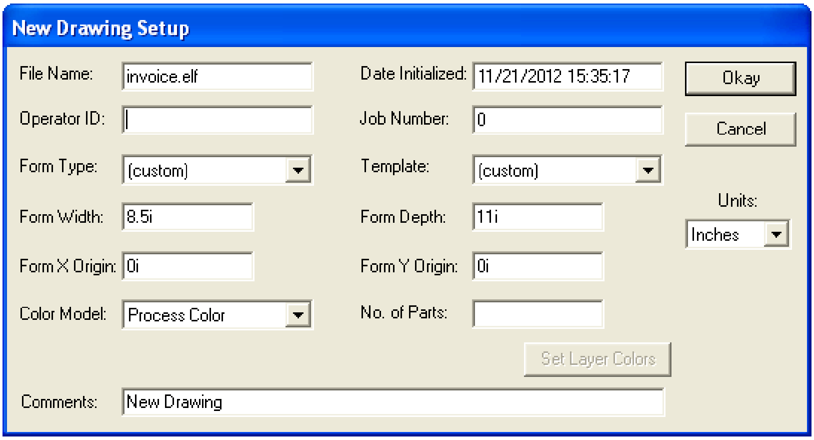
Generate with VML Graphics allows a larger variety of elements to be displayed within the web browser. Items such as Area Fills, Arcs, Diagonal Lines, and Rotated Text can be displayed using VML graphics. Note that the use of VML increases the load and display time within the end users web browser and is only compatible with Windows IE browers.

When Starting from Stratch

If you are starting from scratch and drawing the form’s artwork, following are some of the basic items to consider. Here you will find advice on page sizes and what components work best with HTML E-Forms.

Considerations When Drawing Base Art

When you click the **New File** tool you are creating a brand new file. For E-Forms we need it to have an extension of **.elf**. Before selecting **[Okay]**, you have an opportunity to type in a new form name and change the form size. The file name must begin with an alpha character and use only lowercase letters. Numbers are permitted but spaces are not. Also, the file name should be unique, since it will eventually be copied to a server.



New Drawing Setup

File Name: invoice.elf

Date Initialized: 11/21/2012 15:35:17

Operator ID:

Job Number: 0

Form Type: (custom)

Template: (custom)

Form Width: 8.5i

Form Depth: 11i

Form X Origin: 0i

Form Y Origin: 0i

Color Model: Process Color

No. of Parts:

Units: Inches

Comments: New Drawing

Figure 10-27: Drawing Properties Holds the Form Size

For electronic forms, systems with resolutions of 1024 x 768 or higher will display the full width of 8.5 inches on a normal portrait form. You can always use the "Scale Factor" value when you generate the HTML to enlarge or reduce the displayed size, as shown in Figure 10-7. Be aware, other factors such as the default Windows display for Large Fonts vs. Small Fonts will also affect the view size within your browser.

These graphical elements are supported by HTML JavaScript forms:

Vectors – Both horizontal and vertical lines.

Boxes – With or without fills. Rounded corners are not supported and will display as squared.

Text – Times and Arial are supported. Other fonts will be converted to Times. All whole point sizes are supported; the vertical and horizontal point size should be the same because "cross-scaling" is not supported. Leading should be set to normal standards, generally one point more than the vertical point size.

Note: Embedded fonts are supported for other typefaces if you follow the Microsoft WEFT guidelines and use an IE browser. There is detailed information at the following link: *http://www.microsoft.com /typography/web/embedding.*

Images – For photos use JPEGs, all other cases use GIFs.

Arcs, Splines, and Areas are only supported via the Vector Markup Language (VML) with the HTML Compatibility Type of "Old Rendering Option" during Generate E-Form. Substituting HTML unsupported elements as GIFs is recommended.

When working with a brand new file, you must save it before trying to generate to an HTML E-Form.

Importing a File

There are many times that the form you want to make into an E-Form is already composed in another format. OneForm Designer Plus can import EPS and PDF files. If you have either of these, we have a few tips on how to improve the look of the document when imported text is generated into HTML. For more information on imported files also see *Chapter 9: Importing EPS and PDF into OneForm Designer Plus* within the *Business Forms Composition Reference Manual.*

After you import a file, save the results then re-open the converted file before proceeding to edit, add fill fields, or generate.

Text Display of Imported Files

When importing PDF or EPS files into OneForm Designer Plus specifically for E-Forms, there are certain steps that should be taken in order to improve the quality of the display of the form.

Assemble broken text into single components. Many times when forms are imported, words will come in as individual letters, or sentences as individual words, etc. This is especially true if kerning, abnormal word space, or letter space has been used in the original creation software.

To re-assemble text, use **Combine Selected Text**. Begin by Grouping all text that you wish to combine, then choose the combine option from the right-mouse pop-up menu. The text will change to one text component and automatically ungroup itself.

When importing files, the Composition Mode always comes in as Flush Left. It will be necessary to check the Composition Mode of text items, especially those that appear to be Flush Right, Centered or Justified if you are going to re-compose them. Since .html substitutes fonts, the anchor position and Composition Mode of text items becomes very important. The Anchor position is where the text is fixed on the page and the Composition Mode is where the text aligns within the text box. For Flush Left, the anchor position is lower-left corner of the text item and the composition mode aligns to the left. To modify a Flush Left text item to be Flush Right; change the composition mode to be Flush Right within Text Properties. Next, position the lower-right corner of the field in its correct anchor position.

Placing Fields and Setting Properties

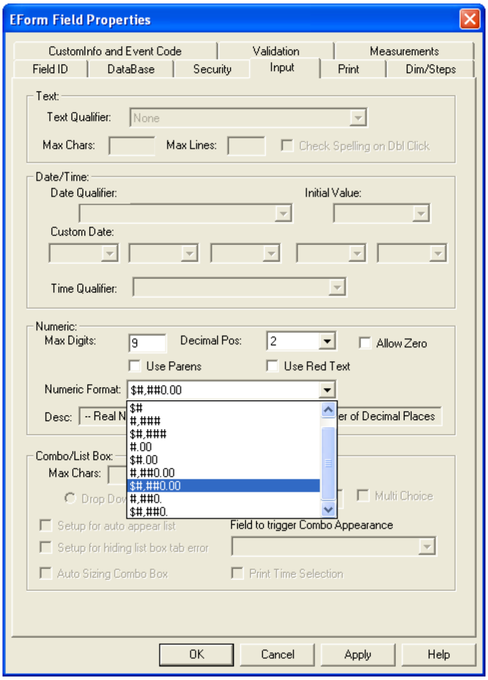
Once the base artwork is complete, begin placing fields on your form. Use the **Add Field** tool and **Auto Field** tool to place fields on your form. Set any standard Form Field Component Properties as you go.

Be sure that the field height and text print size are set at a proper ratio. For example, if the field's vertical size is 12 points, then the text size should not be larger than 10-point. If the text size is too large for the field, the text will appear to be cut off when viewing and printing. On the Generate E-Form Dialog, you have an option to turn on field borders. This will draw a border within the boundary of the field, stealing 2-points from the field text display. In this instance, you will need to make the height 14-points, to accommodate a text size of 10-points.

Buttons need to have a height of at least 14-points to accommodate the caption.

Currency Formatting within a Numeric Field

Numeric fields can hold a currency format within the field's Input tab. There is a list of specific formats: no decimal and with a decimal, with and without a dollar sign. For all of these formats set the Maximum Digits to include the decimal count and then how many of those digits should be after the decimal.



Max Digits: 9

Decimal Pos: 2

Numeric Format: $#,##0.00

Desc: -- Real N … er of Decimal Places

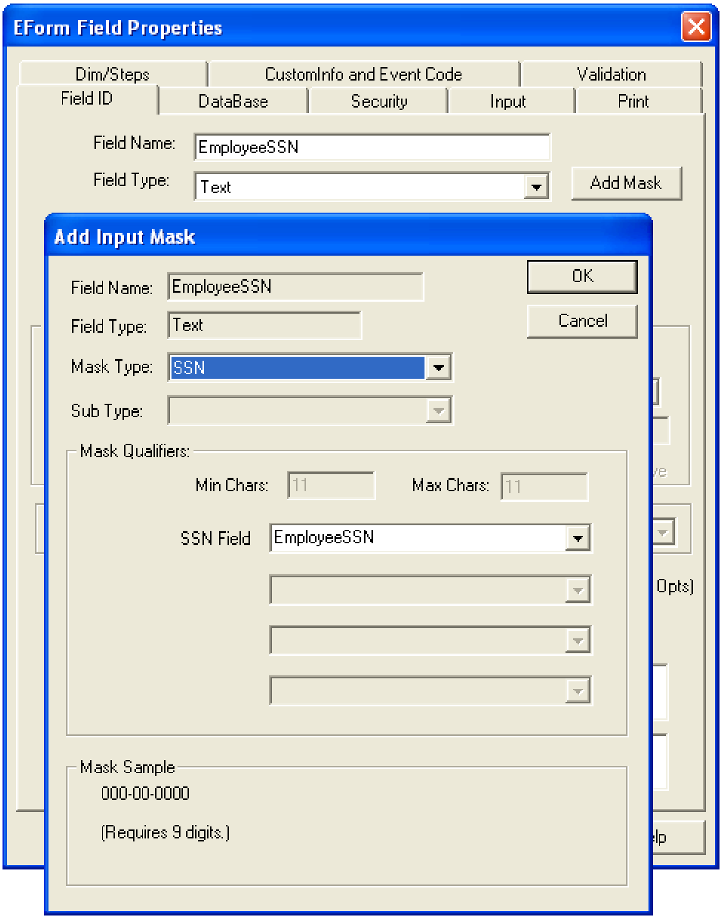
$#
#,###
$#,###
#.00
$#.00
#,##0.00
$#,##0.00
#,##0.
$#,##0.

Figure 10-29: Numeric Formats in Field Properties

Adding a Mask to a Text Field with [Add Mask]

A formatting mask can be added to TEXT fields. This controls the look of the data after the user tabs out of the field. Common masks are phone numbers and Social Security Number. OFDP has a wide variety of input masks: Bank Number, City State Zipcode, Federal ID, Pad Zero, Percent, Social Security Number, TimeString, and many others.

Our example puts hyphens into a Social Security Number. Click the [Add Mask] button, followed by selecting the Mask Type: SSN, as shown.



EForm Field Properties

Field Name: EmployeeSSN

Field Type: Text

Add Input Mask

Field Name: EmployeeSSN

Field Type: Text

Mask Type: SSN

Sub Type:

Mask Qualifiers:

Min Chars: 11 Max Chars: 11

SSN Field: EmployeeSSN

Mask Sample

000-00-0000

(Requires 9 digits.)

Figure 10-30: Click the [Add Mask] Button and Select SSN

CHAPTER 11

Electronic Forms Local File Save

The electronic forms created in the previous chapter are designed to submit the form data to a database located on the web server. If it is necessary for the form to save data on the local computer's harddrive, then adding the following buttons to your E-Form will allow this capability.

1. Add the following button(s) to your form. They can be any size and have a caption of your choice. The Field Name is controlling the action, so they must match exactly:

   - **LocalFileSave** – This button will save the data to the local computer with a choice of either a .FIL or .XML as the data format.
   - **LocalFileOpen** – This button will bring up an Open File dialog, allowing you to select either a .FIL or .XML file from the local computer.
   - **LocalFileClear** – This button will clear data out of the form.
   - **LocalFileSaveFil** – This button will always save the data to the local computer as a .FIL data file.
   - **LocalFileSaveXml** – This button will always save the data to the local computer as a .XML data file.

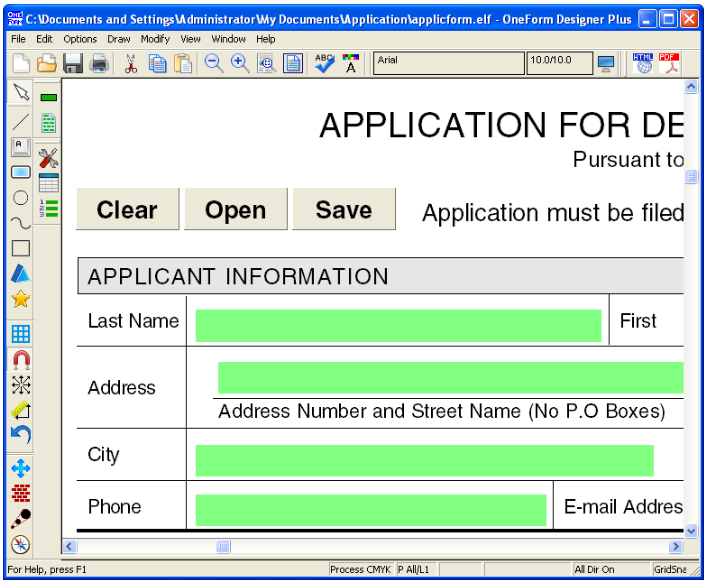
2. The Open and Save buttons are looking for the "C:\My Documents" folder. If this folder does not exist, the default will be the defined Windows Temp folder. To specify a particular folder instead, add the following variable to E-Form Preference's *BeginGenformOptions.*

   ```
   BeginGenformOptions
   ...
   ...
   ...
   LocalFileSavePath=any valid path on local system
   EndGenformOptions
   ```

3. When Generating the HTML, the checkbox for **Generate with VBScript** must be checked. This is found on the Generate Form Dialog Box.

4. You must allow ActiveX objects to run with your browser in order to locally save and open E-Forms. This may require a change in your browser security settings as described in this chapter. This will not place you at risk because you will always be "prompted" first to see if you wish to proceed before ActiveX objects start running.

Following is an example of the LocalFileSave, LocalFileOpen, and LocalFileClear buttons. You will want to place them at the top left-hand corner.



C:\Documents and Settings\Administrator\My Documents\Application\applicform.elf - OneForm Designer Plus

APPLICATION FOR DE

Pursuant to

Application must be filed

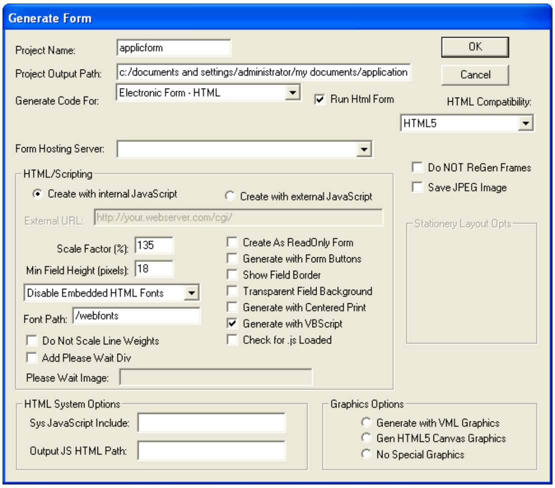
APPLICANT INFORMATION

Last Name	🟩	First	
Address	🟩 Address Number and Street Name (No P.O Boxes)		
City	🟩		
Phone	🟩	E-mail Addres	

For Help, press F1 | Process CMYK | P All/L1 | All Dir On | GridSna

Figure 11-1: Local Open/Save/Clear Buttons Added to Form

When you generate the HTML file, be sure to check the option *Generate with VBScript.*



Generate Form

Project Name: applicform

Project Output Path: c:/documents and settings/administrator/my documents/application

Generate Code For: Electronic Form - HTML

☑ Run Html Form

HTML Compatibility: HTML5

Form Hosting Server:

☐ Do NOT ReGen Frames

☐ Save JPEG Image

HTML/Scripting

● Create with internal JavaScript ○ Create with external JavaScript

External URL: http://your.webserver.com/cgi/

Scale Factor (%): 135

Min Field Height (pixels): 18

Disable Embedded HTML Fonts

Font Path: /webfonts

☐ Do Not Scale Line Weights

☐ Add Please Wait Div

Please Wait Image:

☐ Create As ReadOnly Form

☐ Generate with Form Buttons

☐ Show Field Border

☐ Transparent Field Background

☐ Generate with Centered Print

☑ Generate with VBScript

☐ Check for .js Loaded

Stationery Layout Opts

HTML System Options

Sys JavaScript Include:

Output JS HTML Path:

Graphics Options

○ Generate with VML Graphics

○ Gen HTML5 Canvas Graphics

○ No Special Graphics

Figure 11-2: Generated HTML E-Form

Allowing ActiveX Objects to Run

Since browsers can be setup to stop harddrive access, you will need to check the browser settings and possibly enable access. E-Forms are opening and saving as an ActiveX object. Following are the steps to check your settings:

Step 1: Click the Tools Menu, then Internet Options.

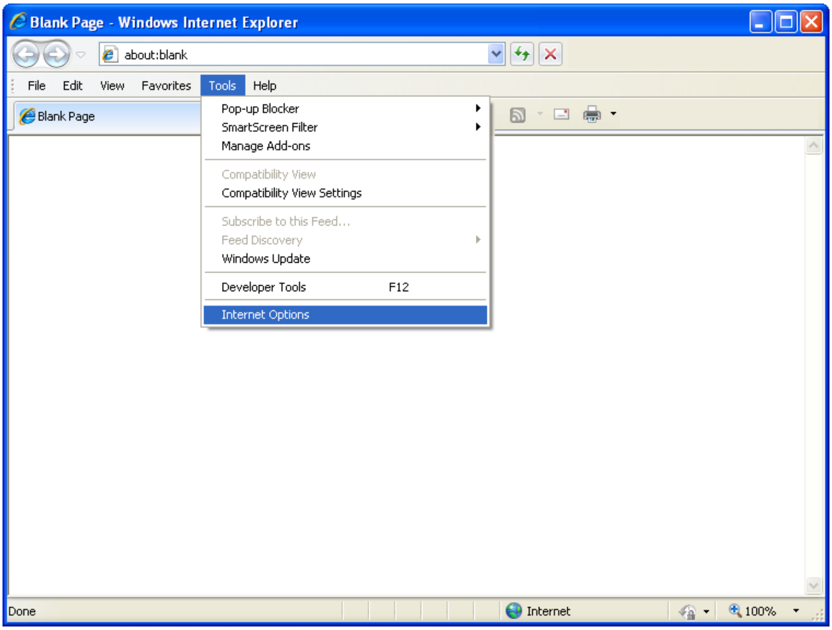
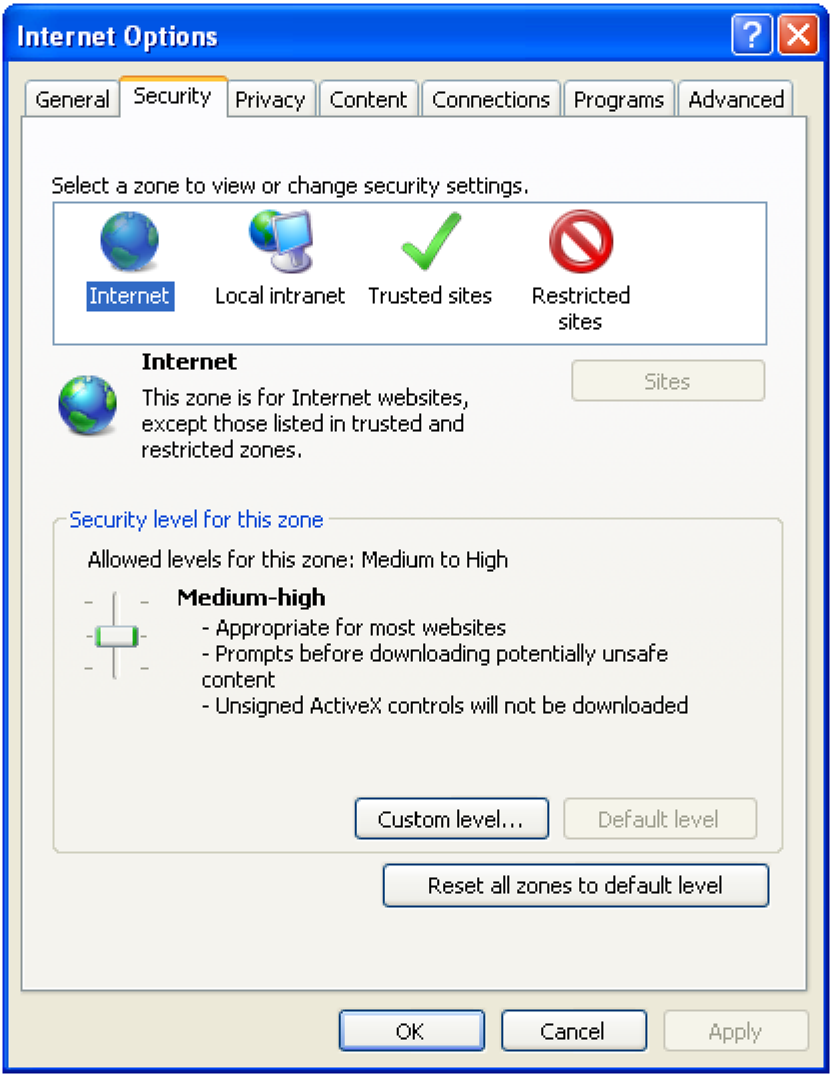


Figure 11-3: The Browser's Tool Menu

The following Pop-Up Window will appear with a Security Tab.

Step 2: Click the Security Tab and the following Pop-Up window will appear with a [Custom Level] button.



Select a zone to view or change security settings.

Internet

This zone is for Internet websites, except those listed in trusted and restricted zones.

Security level for this zone

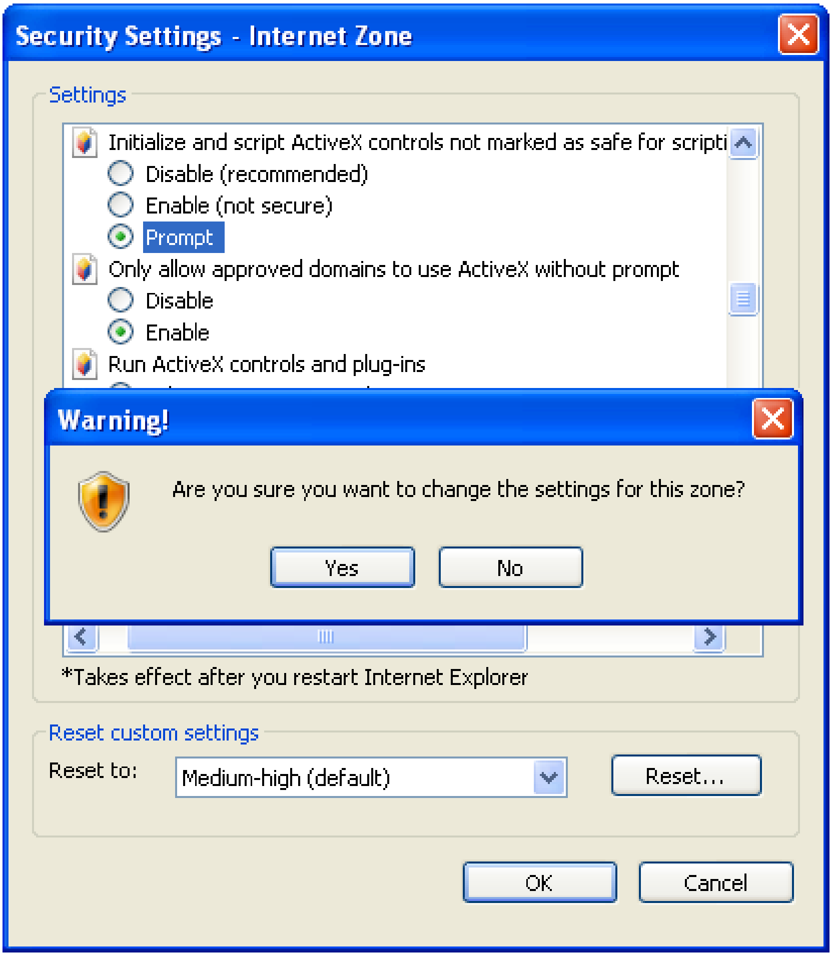
Allowed levels for this zone: Medium to High

Medium-high

- Appropriate for most websites
- Prompts before downloading potentially unsafe content
- Unsigned ActiveX controls will not be downloaded

Figure 11-4: The Security Dialog

Step 3: Click the [Custom Level] button and the following Pop-Up window will appear with ActiveX Options in its list.



Security Settings - Internet Zone

Settings

Initialize and script ActiveX controls not marked as safe for scripti

○ Disable (recommended)

○ Enable (not secure)

● Prompt

Only allow approved domains to use ActiveX without prompt

○ Disable

● Enable

Run ActiveX controls and plug-ins

Warning!

Are you sure you want to change the settings for this zone?

*Takes effect after you restart Internet Explorer

Reset custom settings

Reset to: Medium-high (default)

Figure 11-5: The ActiveX Options

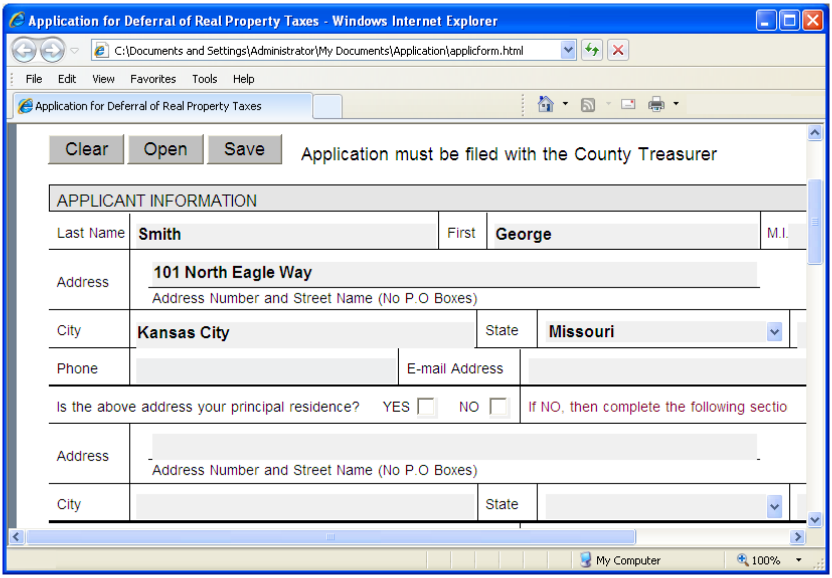
Specifically look for *Initialize and script ActiveX controls not marked as safe.*

Step 4: Take a look at the *Initialize and script ActiveX controls not marked as safe*. If it is "Disabled", change it to "Prompt", then click "Yes" to save your settings.

Step 5: Click [Apply] and [OK] to save your settings and return to the browser.

Using the E-Form Local Save/Open Buttons

It is a good idea to add the buttons to the top of the form. The Open and Save Dialog Boxes will always fall in the upper-left corner of the form, even if the buttons happen to be elsewhere.

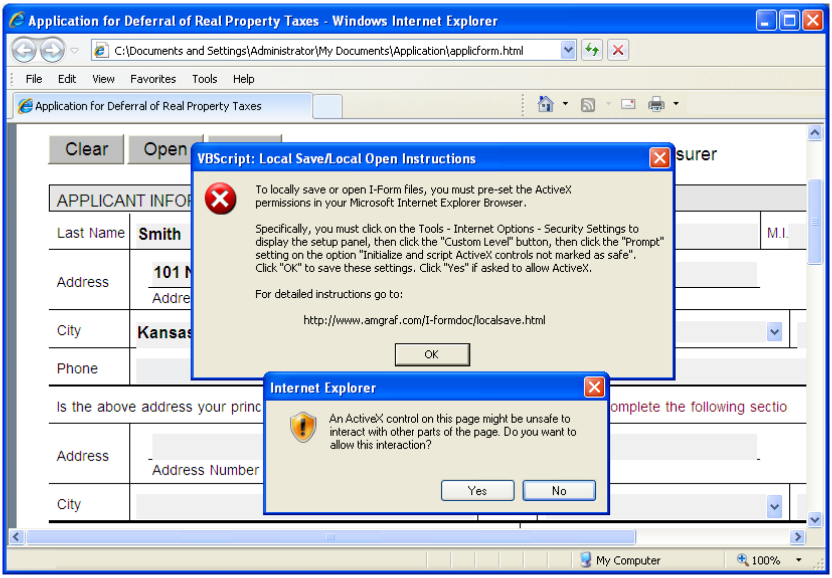

Application for Deferral of Real Property Taxes - Windows Internet Explorer
C:\Documents and Settings\Administrator\My Documents\Application\applicform.html
File
Edit
View
Favorites
Tools
Help
Application for Deferral of Real Property Taxes
Clear
Open
Save
My Computer
100%

Application must be filed with the County Treasurer

APPLICANT INFORMATION				
Last Name	Smith	First	George	M.I.
Address	101 North Eagle Way Address Number and Street Name (No P.O Boxes)			
City	Kansas City	State	Missouri	
Phone		E-mail Address		
Is the above address your principal residence?	YES □ NO □	If NO, then complete the following sectio		
Address	Address Number and Street Name (No P.O Boxes)			
City		State		

Figure 11-6: An E-Form with Local Open/Save Buttons

The first time one of these buttons are used during each session, the following messages will appear. This is caused by the inherent browser characteristic to not allow HTML local harddrive access.


Application for Deferral of Real Property Taxes - Windows Internet Explorer
C:\Documents and Settings\Administrator\My Documents\Application\applicform.html
File Edit View Favorites Tools Help
Application for Deferral of Real Property Taxes
Clear
Open
surer
APPLICANT INFOR
Last Name
Smith
M.I.
Address
101 N
Addre
City
Kansas
Phone
Is the above address your princ
omplete the following sectio
Address
Address Number
City
My Computer
100%

VBScript: Local Save/Local Open Instructions

To locally save or open I-Form files, you must pre-set the ActiveX permissions in your Microsoft Internet Explorer Browser.

Specifically, you must click on the Tools - Internet Options - Security Settings to display the setup panel, then click the "Custom Level" button, then click the "Prompt" setting on the option "Initialize and script ActiveX controls not marked as safe". Click "OK" to save these settings. Click "Yes" if asked to allow ActiveX.

For detailed instructions go to:

http://www.amgraf.com/I-formdoc/localsave.html

Internet Explorer

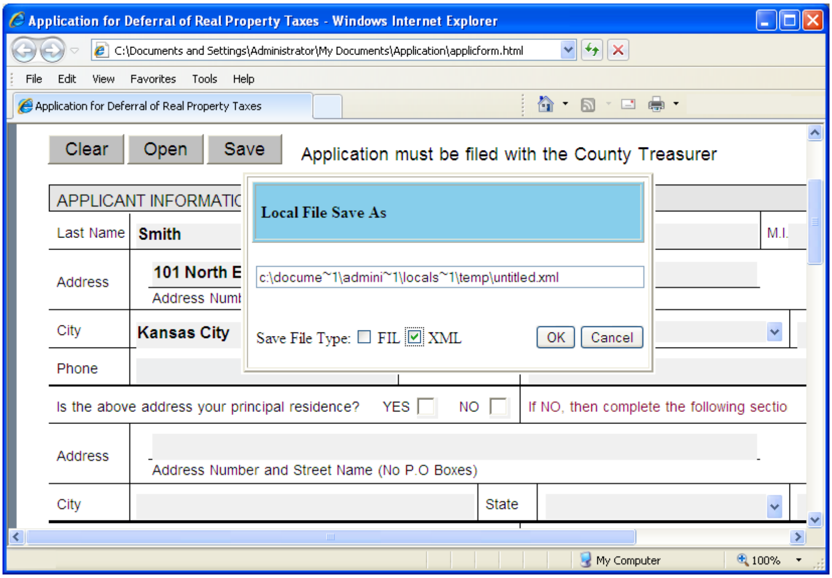
An ActiveX control on this page might be unsafe to interact with other parts of the page. Do you want to allow this interaction?

Figure 11-7: Messages Appear Once During Each Session

If you answer **No,** you will not be allowed to save or open files locally during this session.

Answering **Yes,** will bring up a dialog for saving and naming either a .FIL file or .XML file. These are the type of files that can hold the input data strings.

We are going to save ours as an .XML file.


Application for Deferral of Real Property Taxes - Windows Internet Explorer
C:\Documents and Settings\Administrator\My Documents\Application\applicform.html
File Edit View Favorites Tools Help
Application for Deferral of Real Property Taxes
Clear
Open
Save
My Computer
100%

Application must be filed with the County Treasurer

APPLICANT INFORMATIC

Last Name: Smith — M.I.

Address: 101 North E — Address Numl

City: Kansas City

Phone

Is the above address your principal residence? YES □ NO □ If NO, then complete the following sectio

Address — Address Number and Street Name (No P.O Boxes)

City — State

Local File Save As

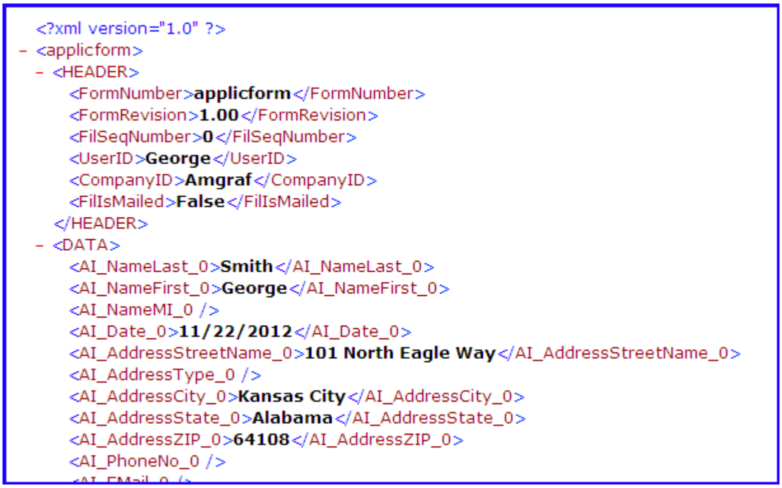
c:\docume~1\admini~1\locals~1\temp\untitled.xml

Save File Type: □ FIL ☑ XML [OK] [Cancel]

Figure 11-8: The Local Save Dialog Box

The default for saving a .XML file will be in the directory Windows Login *Local Settings\Temp folder unless you type in a different path along with the file name.*

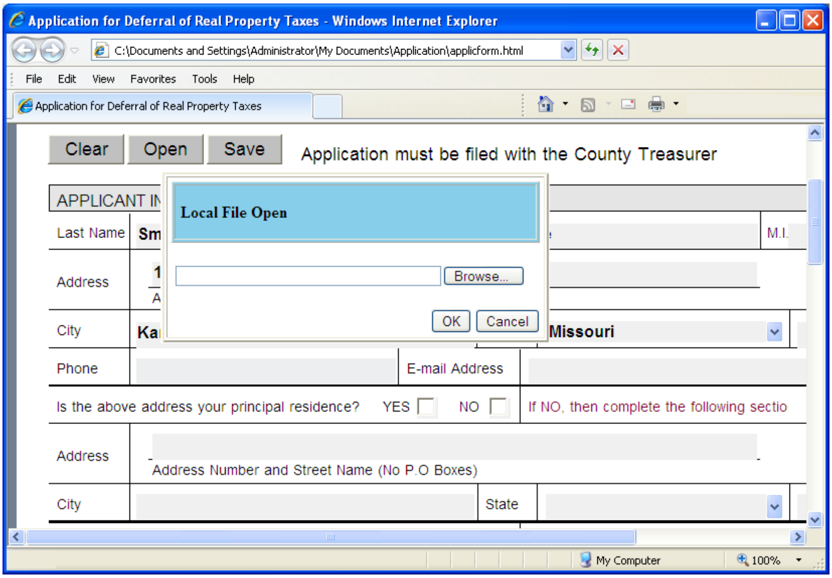
If you want the path, just type over the "untitled" with your file name.



```
   <?xml version="1.0" ?>
 - <applicform>
   - <HEADER>
       <FormNumber>applicform</FormNumber>
       <FormRevision>1.00</FormRevision>
       <FilSeqNumber>0</FilSeqNumber>
       <UserID>George</UserID>
       <CompanyID>Amgraf</CompanyID>
       <FilIsMailed>False</FilIsMailed>
     </HEADER>
   - <DATA>
       <AI_NameLast_0>Smith</AI_NameLast_0>
       <AI_NameFirst_0>George</AI_NameFirst_0>
       <AI_NameMI_0 />
       <AI_Date_0>11/22/2012</AI_Date_0>
       <AI_AddressStreetName_0>101 North Eagle Way</AI_AddressStreetName_0>
       <AI_AddressType_0 />
       <AI_AddressCity_0>Kansas City</AI_AddressCity_0>
       <AI_AddressState_0>Alabama</AI_AddressState_0>
       <AI_AddressZIP_0>64108</AI_AddressZIP_0>
       <AI_PhoneNo_0 />
       <AI_EMail_0 />
```

Figure 11-9: A look at the Saved .XML File

When opening a previously saved file, just [Browse] for the .XML file.


Application for Deferral of Real Property Taxes - Windows Internet Explorer
C:\Documents and Settings\Administrator\My Documents\Application\applicform.html
File
Edit
View
Favorites
Tools
Help
Application for Deferral of Real Property Taxes
Clear
Open
Save
Browse...
OK
Cancel
My Computer
100%

Application must be filed with the County Treasurer

Local File Open

APPLICANT IN

Last Name Sm

M.I.

Address 1

A

City Ka

Missouri

Phone

E-mail Address

Is the above address your principal residence? YES □ NO □

If NO, then complete the following sectio

Address

Address Number and Street Name (No P.O Boxes)

City

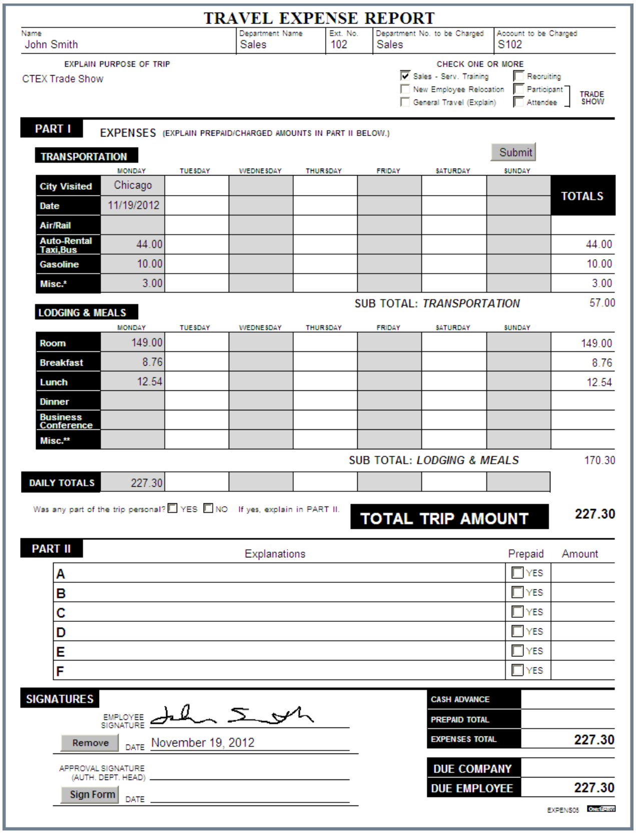
State

Figure 11-10: The Local Open Dialog Box

CHAPTER 12

HTML E-Forms with Signature Fields

The Signature Field is a very unique type of field used for locking a group of visible fields on a form. Signatures provide a method of guaranteeing that the data is locked and has not been altered. If any data changes, the signature will go away and the fields become unlocked. Below is an example of a standard expense form with two signatures available.



TRAVEL EXPENSE REPORT

Name	Department Name	Ext. No.	Department No. to be Charged	Account to be Charged
John Smith	Sales	102	Sales	S102

EXPLAIN PURPOSE OF TRIP

CTEX Trade Show

CHECK ONE OR MORE

☑ Sales - Serv. Training
□ New Employee Relocation
□ General Travel (Explain)
□ Recruiting
□ Participant
□ Attendee
TRADE SHOW

PART I EXPENSES (EXPLAIN PREPAID/CHARGED AMOUNTS IN PART II BELOW.)

TRANSPORTATION

Submit

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	TOTALS
City Visited	Chicago							
Date	11/19/2012							
Air/Rail								
Auto-Rental Taxi,Bus	44.00							44.00
Gasoline	10.00							10.00
Misc.*	3.00							3.00

SUB TOTAL: *TRANSPORTATION* 57.00

LODGING & MEALS

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	
Room	149.00							149.00
Breakfast	8.76							8.76
Lunch	12.54							12.54
Dinner								
Business Conference								
Misc.**								

SUB TOTAL: *LODGING & MEALS* 170.30

DAILY TOTALS	227.30						

Was any part of the trip personal? □ YES □ NO If yes, explain in PART II.

TOTAL TRIP AMOUNT 227.30

PART II

	Explanations	Prepaid	Amount
A		□ YES	
B		□ YES	
C		□ YES	
D		□ YES	
E		□ YES	
F		□ YES	

SIGNATURES

EMPLOYEE SIGNATURE John Smith

Remove DATE November 19, 2012

APPROVAL SIGNATURE (AUTH. DEPT. HEAD)

Sign Form DATE

CASH ADVANCE	
PREPAID TOTAL	
EXPENSES TOTAL	227.30
DUE COMPANY	
DUE EMPLOYEE	227.30

EXPENS05

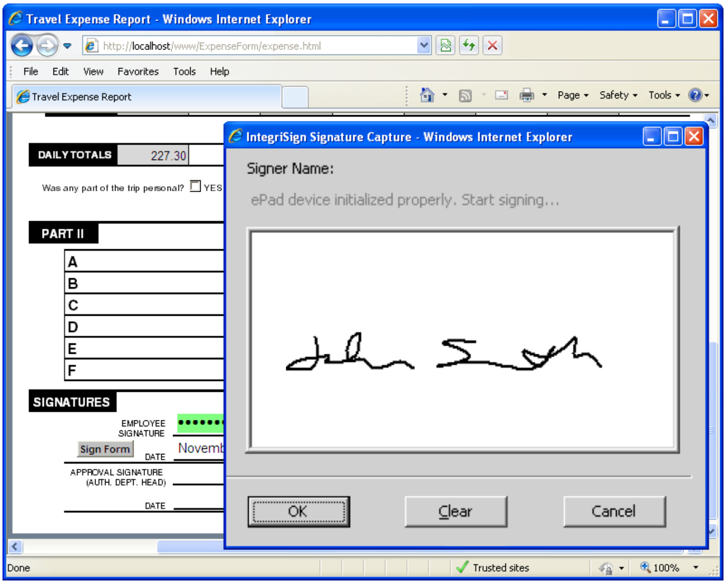
Figure 12-1: Expense Form with Signatures

Once a signature is applied, all fields associated with the signature become locked which makes the information within those fields un-editable.

Signature fields have been automated to work with HTML E-Forms. We will discuss using the Signature Type for an ePad and no Pad. The ePad is with Interlink Electronics, Inc. ePad-Ink Signature Pads Version 7.1.

Using Signatures on HTML E-Forms

When an E-Form with Signatures is opened, the Sign Button for the first signature is available. Clicking the [Sign Form] button causes two actions: focus to the password field so that a pass phrase can be entered, and activate the ePad Signature Pad if available.



Travel Expense Report - Windows Internet Explorer

http://localhost/www/ExpenseForm/expense.html

DAILY TOTALS 227.30

Was any part of the trip personal? □ YES

PART II

A

B

C

D

E

F

SIGNATURES

EMPLOYEE SIGNATURE •••••••

DATE Novemb

APPROVAL SIGNATURE (AUTH. DEPT. HEAD)

DATE

IntegriSign Signature Capture - Windows Internet Explorer

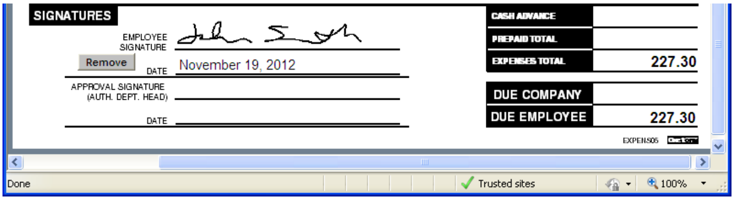
Signer Name:

ePad device initialized properly. Start signing...

Done

Figure 12-2: Signing the Form with the ePad

When the password field becomes available, type in your pass phrase and [Tab]. If the ePad Signature Pad is found, signing the pad will cause the signature image to appear within its capture window and place it on the form. If using No Pad, then the image is replaced with "Signature on File" text. Then the signature date and all fields on the form of the same security lock level will become read-only. It will immediately replace the [Sign Form] Button with the [Remove] Button.



SIGNATURES

EMPLOYEE SIGNATURE

DATE November 19, 2012

APPROVAL SIGNATURE (AUTH. DEPT. HEAD)

DATE

CASH ADVANCE	
PREPAID TOTAL	
EXPENSES TOTAL	227.30
DUE COMPANY	
DUE EMPLOYEE	227.30

EXPENS05

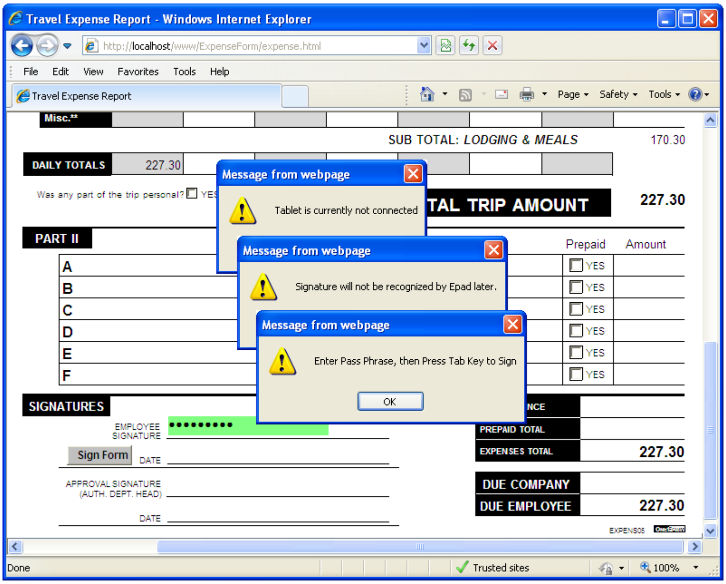
Done

Figure 12-3: [Remove] Button Becomes Available

Clicking on the [Remove] Button causes the action of focusing to the password field. If the password is correct, then the signature is removed and the fields on the form are unlocked. Also, the [Sign Form] Button returns for signing again later. Removing multiple signatures must be removed consecutively from last applied to first.

Applying a Password without a Signature Image

If using the ePad type and the Signature Pad is not found, the following messages will appear letting you know that the signature cannot be recognized. At that time the password field will become available. Type in your pass phrase and [Tab]. This will lock the signature with the "Signature on File" text, and lock all fields on the form of the same security lock level.



Travel Expense Report - Windows Internet Explorer

http://localhost/www/ExpenseForm/expense.html

Message from webpage: Tablet is currently not connected

Message from webpage: Signature will not be recognized by Epad later.

Message from webpage: Enter Pass Phrase, then Press Tab Key to Sign

EMPLOYEE SIGNATURE: ●●●●●●●●●

SUB TOTAL: LODGING & MEALS 170.30

DAILY TOTALS 227.30

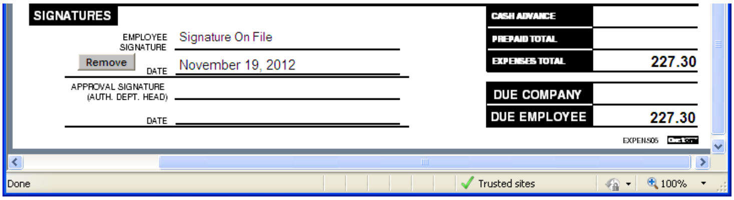
TOTAL TRIP AMOUNT 227.30

EXPENSES TOTAL 227.30

DUE EMPLOYEE 227.30

Figure 12-4: ePad Signature Pad Not Found

Clicking on the [Remove] Button causes the action of focusing to the password field. If the password is correct, then the "Signature on File" text is removed and the fields on the form are unlocked. Also, the [Sign Form] Button returns.



EMPLOYEE SIGNATURE: Signature On File

DATE: November 19, 2012

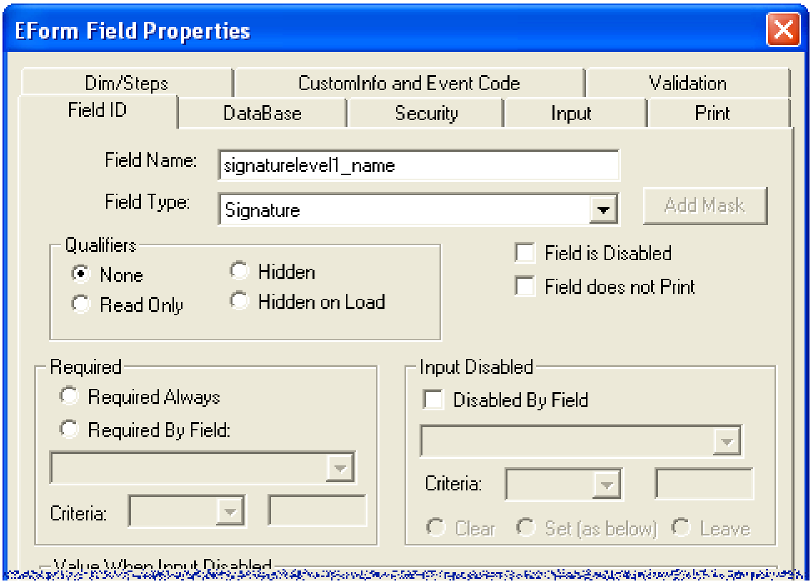
EXPENSES TOTAL 227.30

DUE EMPLOYEE 227.30

Figure 12-5: Example of No Pad "Signature on File"

The Signature Field Type

The Signature Field Type is unique and used for locking all visible fields associated with its security lock level.



Field Name: signaturelevel1_name

Field Type: Signature

Qualifiers

● None ○ Hidden

○ Read Only ○ Hidden on Load

□ Field is Disabled

□ Field does not Print

Required

○ Required Always

○ Required By Field:

Criteria:

Input Disabled

□ Disabled By Field

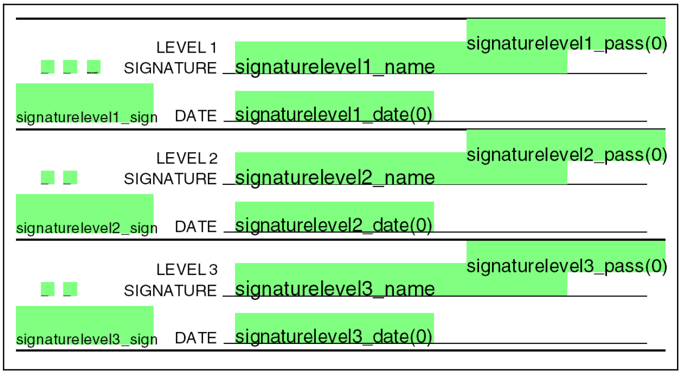
Criteria:

○ Clear ○ Set (as below) ○ Leave

Value When Input Disabled

Figure 12-6: Signature Field Type

Signature fields do not work alone. They must be accompanied by other fields of the same security level to make them work. In Figure 12-7 below, you can see that it takes many fields to produce three signatures. Each Signature requires six fields sharing the same Security Lock Level. They are created through the Server Preferences Signature Tab.



LEVEL 1 SIGNATURE signaturelevel1_name signaturelevel1_pass(0)

signaturelevel1_sign DATE signaturelevel1_date(0)

LEVEL 2 SIGNATURE signaturelevel2_name signaturelevel2_pass(0)

signaturelevel2_sign DATE signaturelevel2_date(0)

LEVEL 3 SIGNATURE signaturelevel3_name signaturelevel3_pass(0)

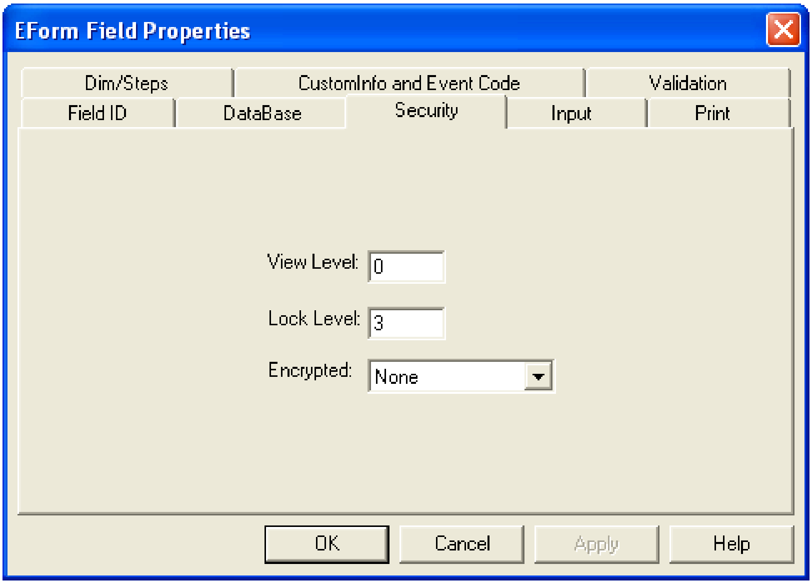
signaturelevel3_sign DATE signaturelevel3_date(0)

Figure 12-7: Closeup of the Signature Fields

Applying Security Lock Levels

Security Lock Levels can be applied to any field through the Field Properties **Security** tab.

When using signatures, all fields on the form should be given a security lock level. The security lock level applied to the Signature field is important. All other fields with a security lock level ***equal to or lower than*** the Signature field will be locked, once the password has been applied.



View Level: 0

Lock Level: 3

Encrypted: None

Figure 12-8: Security Tab

Security levels are 0 through 15, with 0 being the lowest. Use lock level 1 for all fields being locked when the first signature is applied, 2 for the second, and so on. Fields left with a 0 will be locked when the first signature is applied. Fields with a number higher than all signatures on the form will never be locked.

When there is more than one signature, the Signature Field lock levels must run consecutively, always starting with 1.

You can use the Group option to select all fields for a particular lock level and change them all at once.

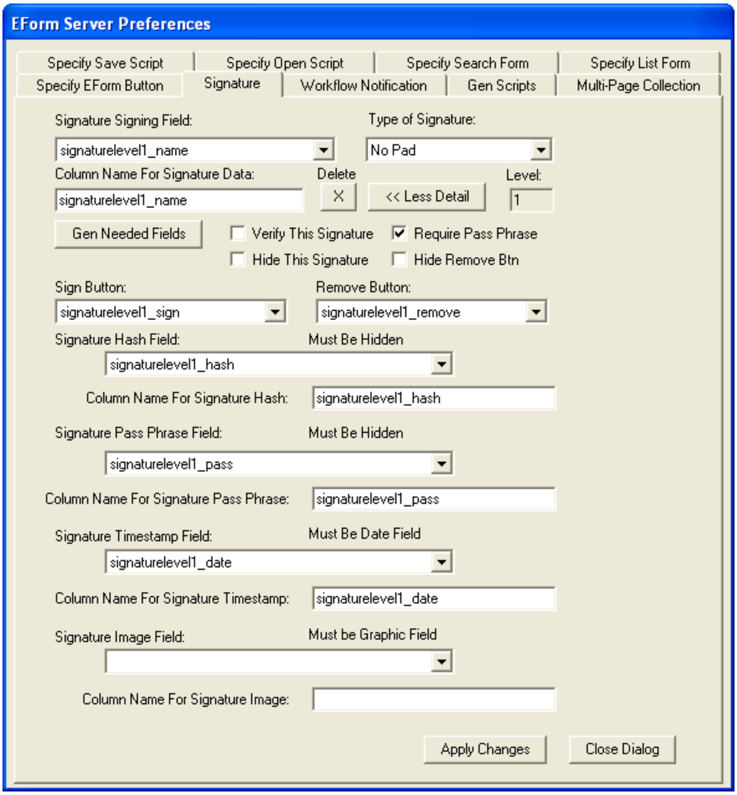
The E-Forms Signature Tab Dialog

General Information

Each Signature requires six fields sharing the same Security Lock Level:

Field Use	Field Type	Field Suffix
Signature Image Level	Signature Field	_name
Sign Form Button	Button Field	_sign
Remove Signature Button	Button Field	_remove
Signature Hash Value	Hidden Text Field	_hash
Signature Pass Phrase	Text Field	_pass
Signature Timestamp	Date Field	_date

These fields are linked through the EForm Server Preferences Signature Tab dialog. Clicking [Apply] will read the Signature Field's Security Lock Level and apply it into the Form's Preferences.



Signature Signing Field: signaturelevel1_name

Type of Signature: No Pad

Column Name For Signature Data: signaturelevel1_name

Delete

Level: 1

□ Verify This Signature ☑ Require Pass Phrase

□ Hide This Signature □ Hide Remove Btn

Sign Button: signaturelevel1_sign

Remove Button: signaturelevel1_remove

Signature Hash Field: Must Be Hidden
signaturelevel1_hash

Column Name For Signature Hash: signaturelevel1_hash

Signature Pass Phrase Field: Must Be Hidden
signaturelevel1_pass

Column Name For Signature Pass Phrase: signaturelevel1_pass

Signature Timestamp Field: Must Be Date Field
signaturelevel1_date

Column Name For Signature Timestamp: signaturelevel1_date

Signature Image Field: Must be Graphic Field

Column Name For Signature Image:

Figure 12-9: EForm Server Preferences Signature Tab Dialog

When a form is signed, the signature along with any visible data fields with that signature's security lock level, are combined and hashed creating a unique hash value. These fields are then locked. All locked fields will appear gray and the information within those fields will be un-editable.

If any data changes and the signature hash value cannot be validated, the signature will go away and the fields become unlocked.

The two sign and remove buttons, along with the pass phrase field, allows the signer to apply and remove the signature to the form.

Note: Only visible data fields are used to compute the signature data. Hidden fields do not apply.

Filling Out the Signature Tab Dialog

Signature Type – This is the type of Signature that is being applied. Each type behaves differently. The types available are: *No Pad, ePad Interlink Driver, CIC Signature, Voice Signature,* and *Appligent SecureSign.* Using the "No Pad" type requires no additional software or drivers. All others require their proprietary software and drivers. This chapter discusses only the first two types.

Signature Signing Field – This will list all Signature Field Types on the form. Its lock level will be read when you select it. Recommended field name suffix is _name. This field's data is stored in the Primary Output Table. This field needs to be created and assigned a lock level.

The six fields necessary for each signature to work must be linked to the form through this dialog. These fields must share the same lock level. Clicking on the [Generate Needed Fields] button will automatically create each field and place them on the form surrounding the Signature field.

Sign Button – This is a Button Field used to activate the signing process. Create a button field and label it appropriately. Recommended field name suffix is _sign.

Remove Button – This is a Button Field used to activate the removal of a signature. Create a button field and label it appropriately. Recommended field name suffix is _remove.

Signature Hash Field – This is a Text Field that is hidden. It is used to hold the signature's unique hash value. Recommended field name suffix is _hash. This field's data is stored in the Primary Output Table.

Signature Pass Phrase Field – This is a Text Field that is used to input the password or pass phrase locking the signature. Recommended field name suffix is _pass. This field's data is stored in the Primary Output Table.

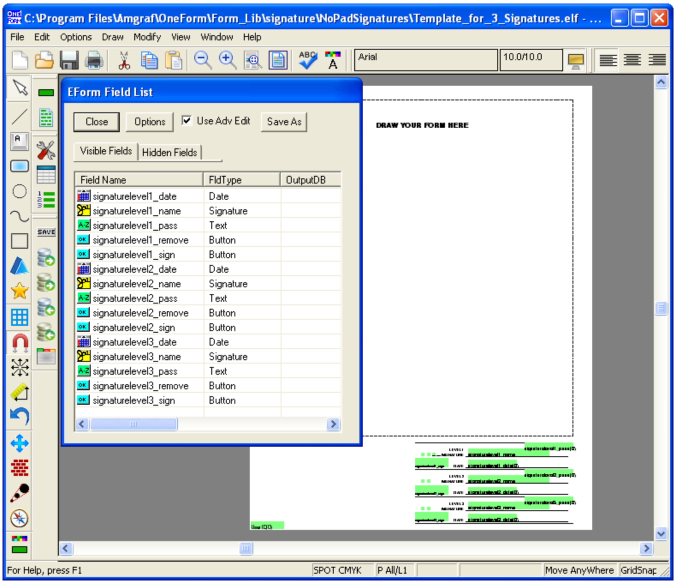
Signature Timestamp Field – This is a Date Field used to keep track of the date the signature is applied. Recommended field name suffix is _date. This field's data is stored in the Primary Output Table.

Signature Image Field – This is the image field for the image being captured in the ePad Interlink signature type. Recommended field name suffix is _image. This field's data is stored in the Primary Output Table.

Adding Signatures to Your E-Form

Signature Templates

The easiest way to place a signature on your form is to use OneForm's Forms Library signature templates.



C:\Program Files\Amgraf\OneForm\Form_Lib\signature\NoPadSignatures\Template_for_3_Signatures.elf - ...

EForm Field List

☑ Use Adv Edit

Field Name	FldType	OutputDB
signaturelevel1_date	Date	
signaturelevel1_name	Signature	
signaturelevel1_pass	Text	
signaturelevel1_remove	Button	
signaturelevel1_sign	Button	
signaturelevel2_date	Date	
signaturelevel2_name	Signature	
signaturelevel2_pass	Text	
signaturelevel2_remove	Button	
signaturelevel2_sign	Button	
signaturelevel3_date	Date	
signaturelevel3_name	Signature	
signaturelevel3_pass	Text	
signaturelevel3_remove	Button	
signaturelevel3_sign	Button	

Figure 12-10: Example of the Three Signature Template

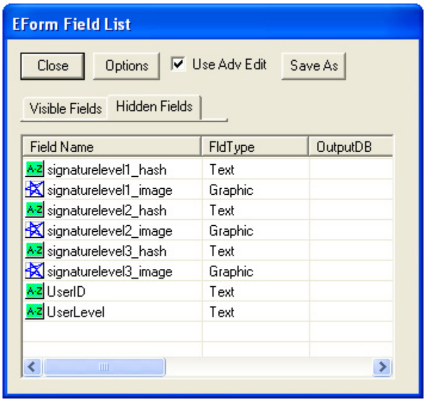
There are signature template files which can expedite adding signatures to your document. All of the necessary fields are drawn and ready to be used. You can draw directly onto a saved version of the template and have the signature fields with their levels already applied and ready to use, or combine the fields into your existing form and apply the copied fields to the Form's Preferences.

Which way you decide will probably depend on the customization already in place on your form.

Things to Remember if Copying the Template Fields

Several of the fields are on top of each other and the hash field is hidden. (See Figure 12-11 on the next page.) Be sure to group all six of the fields needed for each signature. It will not work if one of them is missing.

After copying the signature fields into your existing form, be sure to apply the properties of the fields through the Signature Tab dialog. Also, remember to include them in the Primary Output Table.



☑ Use Adv Edit

Field Name	FldType	OutputDB
signaturelevel1_hash	Text	
signaturelevel1_image	Graphic	
signaturelevel2_hash	Text	
signaturelevel2_image	Graphic	
signaturelevel3_hash	Text	
signaturelevel3_image	Graphic	
UserID	Text	
UserLevel	Text	

Figure 12-11: Notice the Hidden Signature Fields

Things to Remember if Copying Existing Forms

When combining or copying a form into a signature template file, any Custom Programming within the Form Preferences, and Tab Order of the fields, will not be copied. These will need to be copied over separately from the Form Preferences.

Even though the templates are setup in the Spot Color model, any Spot Color Layer Colors will not be combined automatically. Use Options, Combine Layer Colors to locate your file. The dialog will show the existing colors versus the colors within the selected file. Choose the colors needed and select okay.

The templates are setup in the Spot Color model. For those needing a Process Color model, change the drawing properties from Spot Color to Process Color first, then save as a new file. Exit OneForm and reopen. When the new form is opened the color model will be process color.

E-Form Signature Examples in the Forms Library

Within the **signature** folder you will find not only three template files, but there are also two examples of completed forms with signatures. Just by opening the HTML file locally in Internet Explorer you will be able to test the first signature on each form.

The *ExpenseFormExercise* folder holds the expense form used to produce the samples in this chapter. Opening the **expense.html** will allow you to try the first signature and lock the form's fields with or without the signature image.

The *Requisition* folder has a **req_3sigs.html** file which can be opened with Internet Explorer and signed.

When testing locally without serving the form, only the first signature will work. When multiple signatures are applied, the second signature does not become active until the form is stored in the Primary Output Database Table and retrieved again. As the form is opened again, the second signature's [Sign Form] Button becomes available.

Both folders are completed projects with signatures applied. If you copy the entire folder to your web server and create the Primary Output Database Table, you will be able to submit and retrieve between each signature making the next level available.

Expense E-Form Exercise

Within the *ExpenseFormExercise* folder there is an ELF file without signatures. It is the *expense_sample.elf.* We will use it in the following exercise.

The Expense form has custom calculations within the Form Preferences to control the calculations of the weekly expenses. For that reason it would be easier to "Copy and Paste" the Signature Fields from the 2 Signature Template for an employee signature and approval signature.

All of the fields on the Expense form have a Security Level of 0. This means that the first signature will lock all field data when signed.

1. ***Copy the Signature Fields:*** Open the Template_for_2_Signatures.elf in OneForm Designer Plus and use the Edit, Group Select Intersect option to identify the items needed. Group the fields along with their text labels. Do not forget the necessary hidden fields. Then click on the Clipboard Copy Tool in the top toolbar. The grouped items are now stored in memory.

2. ***Paste the Signature Fields:*** Open the expense_sample.elf and Zoom into the signature location at the bottom. Click on the Clipboard Paste Tool in the top toolbar and then pick the upper-left corner of where the items are to be placed. They will appear and be grouped. Move them into location before using the Edit, Group Unselect All.

3. ***Edit the Text Labels:*** With the Select Cursor, edit the text next to each signature to the appropriate Employee and Approval labels.

4. ***Restart Tab Order:*** Fields brought into a file through the "Copy and Paste" method will not have any tabbing order, so take the time to restart the tabbing order from the last entry field to the Sign Form Buttons.

5. ***Signature Tab Dialog:*** Now apply the signatures to the form through the EForm Server Preferences Signature Tab Dialog. Link all of the fields for signature level 1 (see Figure 12-9) and select [Apply]. Then link all of the fields for signature level 2 and select [Apply]. Close the dialog.

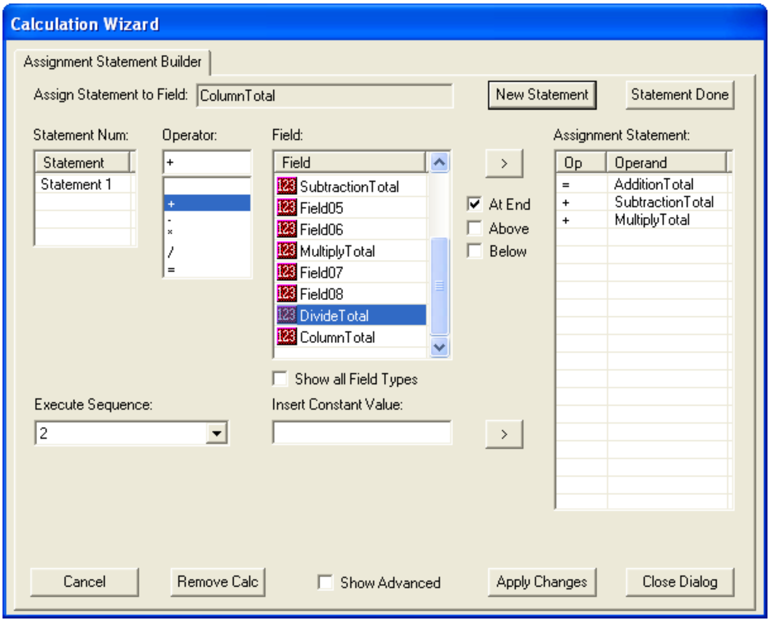
6. ***Test First Signature Locally:*** Save the form and then Generate an Electronic Form - HTML. Check "Run EForm" and test the locking of the input fields through the first Employee signature.

7. ***Prepare the E-Form for a Server:*** The form is now ready for the standard procedures for creating an E-Form as described in *Chapter 10: Creating HTML E-Forms.* This includes attaching the output database primary table and generating CGI scripts.

CHAPTER 13

The Calculation Wizard Option

In this section we will discuss how to use the Calculation Wizard option to do simple math (such as adding or subtracting), with numeric fields and the "Assign Statement Builder" Tab.



Assign Statement to Field: ColumnTotal

Statement
Statement 1

Operator: +

+
-
*
/
=

Field
SubtractionTotal
Field05
Field06
MultiplyTotal
Field07
Field08
DivideTotal
ColumnTotal

☑ At End
□ Above
□ Below

Op	Operand
=	AdditionTotal
+	SubtractionTotal
+	MultiplyTotal

□ Show all Field Types

Execute Sequence: 2

□ Show Advanced

Figure 13-1: The Calculation Wizard Dialog

The Assign Statement Builder Tab is organized to easily choose an Operator and Field Name to be copied [>} to the resulting Assignment Statement column. The JavaScript code is created when the HTML E-Form is generated. It can be previewed, by clicking "Show Advanced".

Assign Statement to Field: This is the field that holds the calculation. It is usually a total field.

[New Statement] and [Statement Done]: Clicking the [New Statement] button will issue a Statement Number. These buttons start and stop each statement. Normally a total field will only need one statement. An example of multiple statements would be adding up many columns of fields. If you need to add an entire page of fields, it is easier to organize them into statements. Clicking [New Statement] will clear the "Assign Statement Column", start a new statement number and let you continue the calculation while staying in the same total field. *See Figure 13-7.*

Statement Number: For multiple statement calculations, clicking on the Statement Number will display the Assignment Statement Column for that number.

Operators are the math being applied to the following field chosen from the list. The available operators are:

+ addition
- subtraction
* multiplication
/ division
= equals

Note: The very first operator assignment will always be an equal.

[>] Copy Button: Copy the selected operator and field to the assignment statement column. There are choices as to where it falls, above or below an existing highlighted item. By default it attaches to the end of the statement.

Execute Sequence Number: This number controls the order that statements appear in the JavaScript code. It is important for forms with subtotals and grand totals. Subtotals need executed first, with the grand total executed second. There are 10 execute levels.

Insert Constant Value [>]: This allows you to apply a constant value to an assignment. For example, if you need a percentage you divide the total by 100. Examples of usage are at the end of this chapter.

Show Advanced: When this is checked, the advanced functions appear along with a preview of the JavaScript Code the statement will generate. *See Figure 13-2.*

Functions Available under "Use Function": The list of functions available is derived from an external XML file named *FormFunctionList.xml,* read during the startup of OneForm.

SumSteps: This function is chosen when you are adding a column of stepped fields.

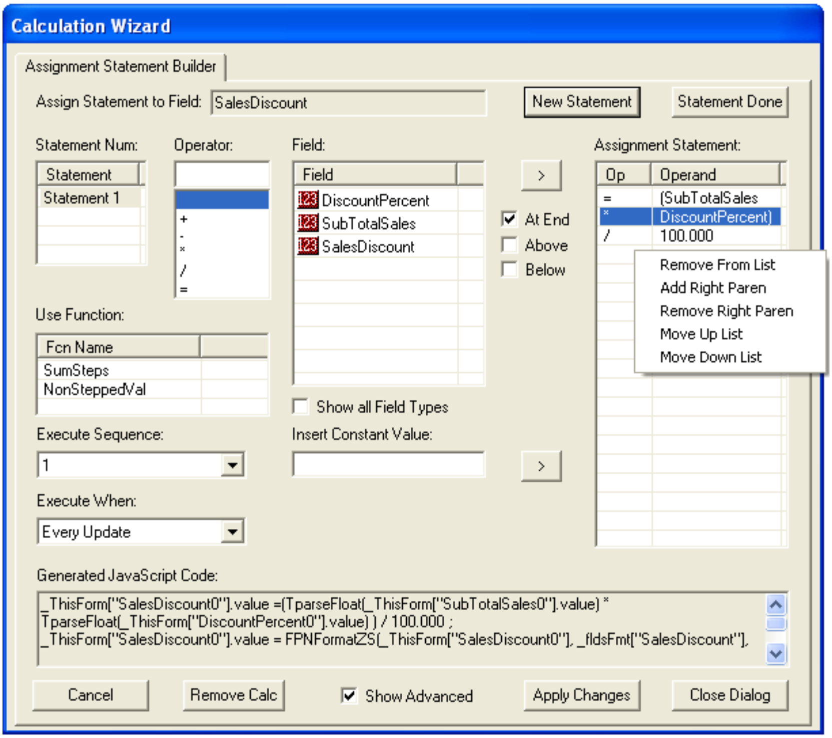
NonSteppedVal: This function is chosen to assist the program when you need to specify that the field picked is not stepped, and yet all others in the calculation are stepped. (By default is looks that the first field assignment and determines whether stepping is applied or not.) This option allows an exception to the rule, per field.

Execute When: This has two options for the location of the JavaScript Code. The first one places the code in MainCalculation, where the second one puts it in the field itself.

Every Update: This is the default. It means to place the JavaScript Code into the "function MainCalculation" and run it for every field update.

During Event: When this option is chosen, it means to place the JavaScript Code into the lost focus event of the total field itself and only execute it during that specific lost focus event.

Assignment Statement Popup Menu: Highlighting an existing assignment statement and clicking the right-mouse will bring up a pop-up menu with options for removing, moving up and down the list, and adding grouping parentheses.



Assign Statement to Field: SalesDiscount

Statement Num: Statement 1

Field: DiscountPercent, SubTotalSales, SalesDiscount

☑ At End ☐ Above ☐ Below

☐ Show all Field Types

Execute Sequence: 1

Execute When: Every Update

Assignment Statement:

Op	Operand
=	(SubTotalSales
*	DiscountPercent)
/	100.000

Generated JavaScript Code:

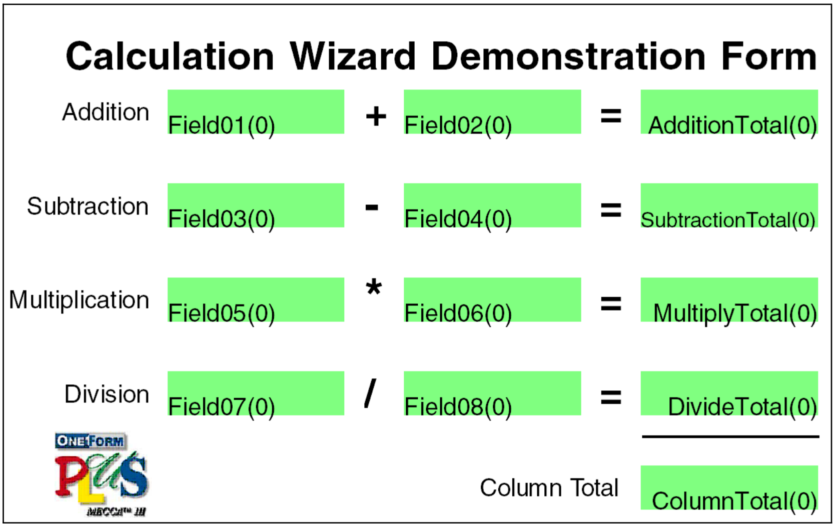
```
_ThisForm["SalesDiscount0"].value =(TparseFloat(_ThisForm["SubTotalSales0"].value) *
TparseFloat(_ThisForm["DiscountPercent0"].value) ) / 100.000 ;
_ThisForm["SalesDiscount0"].value = FPNFormatZS(_ThisForm["SalesDiscount0"], _fldsFmt["SalesDiscount"],
```

☑ Show Advanced

Figure 13-2: Advanced Options with JavaScript Code Preview

Getting Started with Calculations

We will be using the *Examples* file *simple_calculations.elf.* Open it in OneForm and view the field names **(View Menu, EForm Fields, Show Field Names)** to follow along and reference the fields with the steps below.



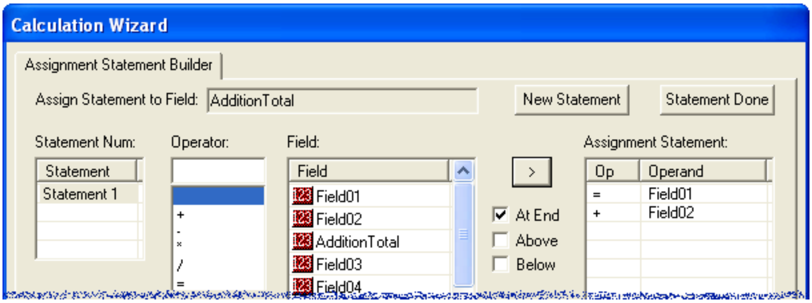
Calculation Wizard Demonstration Form

Addition	Field01(0)	+	Field02(0)	=	AdditionTotal(0)
Subtraction	Field03(0)	-	Field04(0)	=	SubtractionTotal(0)
Multiplication	Field05(0)	*	Field06(0)	=	MultiplyTotal(0)
Division	Field07(0)	/	Field08(0)	=	DivideTotal(0)
				Column Total	ColumnTotal(0)

Figure 13-3: Examples/simple_calculation.elf

In our example we will do the math to each row and then add the third column of totals. Each type of math uses operators to express the action for the field following it.

We will start with the AdditionTotal(0) Field for row 1. Our examples show an *English Description,* the *Field it is Being Assigned To,* and then what appears in the *Assignment Statement* column.



Calculation Wizard

Assignment Statement Builder

Assign Statement to Field: AdditionTotal

☑ At End
☐ Above
☐ Below

Assignment Statement:

Op	Operand
=	Field01
+	Field02

Figure 13-4: Assignment Statement for Addition in Row 1

Note: You will always be selecting the total field that needs to hold the statement. The calculation statement will be placed under the field's *Custom Info Tab.*

Step-by-Step for Simple Calculations

1. To get started use the **Select Cursor Tool** to click on the first total field **AdditionTotal(0),** and go to the popup menu and select **Calculation Wizard.**

This will bring up the dialog for creating a calculation statement for the "AdditionTotal" total field.

English Description: "The AdditionTotal(0) Field equals Field01(0) plus Field02(0)."

Assign Statement to Field: AdditionTotal

2. Click the **[New Statement]** Button. It will place "Statement 1" in the Statement Number Column.

3. Click the **= Operator** as the first statement assignment.

4. Select the field from the list to be added, this would be **Field01.**

5. Click on the **[>] Button** to copy this assignment to the Assignment Statement column.

Assignment Statement: = Field01

6. Click the **+ Operator** as the second statement assignment.

7. Select the field from the list to be added, this would be **Field02.**

8. Click on the **[>] Button** to copy this assignment to the Assignment Statement column.

Assignment Statement: + Field02

9. Click on **[Statement Done]** and the dialog will close. Your calculation is complete for the addition of row 1.

10. Repeat Steps 2 through 9 to create the following statement for row 2 by selecting the **SubtractionTotal(0)** Field with the **Select Cursor** and going to the **Calculation Wizard.**

English Description: "The SubtractionTotal(0) Field equals Field03(0) minus Field04(0)."

Assign Statement to Field: SubtractionTotal

Assignment Statement: = Field03
– Field04

11. Repeat Steps 2 through 9 to create the following statement for row 3 by selecting the **MultipyTotal(0)** Field with the **Select Cursor** and going to the **Calculation Wizard.**

English Description: "The MultiplyTotal(0) Field equals Field05(0) times Field06(0)."

Assign Statement to Field: MultiplyTotal

Assignment Statement: = Field05
* Field06

12. Repeat Steps 2 through 9 to create the following statement for row 4 by selecting the **DivideTotal(0)** Field with the **Select Cursor** and going to the Calculation Wizard.

    English Description: "The DivideTotal(0) Field equals Field07(0) divided by Field08(0)."

    Assign Statement to Field: DivideTotal

    Assignment Statement: = Field07
    / Field08

13. Repeat Steps 2 through 8 to create the following statement for column 3 by selecting the **ColumnTotal(0)** Field with the **Select Cursor** and going to the **Calculation Wizard.**

    English Description: "The ColumnTotal(0) Field equals the Addition Total(0) Field plus SubtractionTotal(0) Field plus MultiplyTotal(0) Field plus DivideTotal(0) Field."

    Assign Statement to Field: ColumnTotal

    Assignment Statement: = AdditionTotal
    + SubtractionTotal
    + MultiplyTotal
    + DivideTotal

14. The **Execute Sequence Number** for the ColumnTotal(0) Field needs to be **2.** This will guarantee that it falls after the row totals (which have an execute sequence of 1) when the JavaScript code is generated.

15. Click on **[Statement Done]** and the dialog will close. Your calculations are complete.

View Menu for Fields with Calculations

The View Menu for calculations displays fields in Purple and shows the Execute Sequence Number.

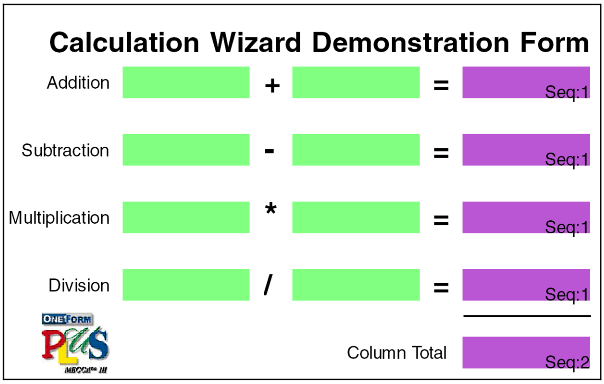


Figure 13-5: Calculation Targets

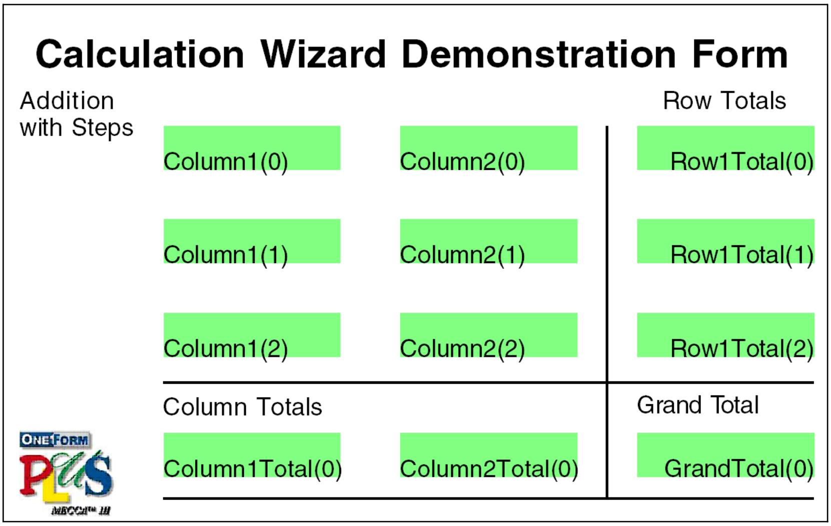
Wizard Generated Code

When a field has calculations applied, XML BeginCalculation/EndCalculation code is placed within that field's Custom Information.

When the HTML or PDF E-Form is generated the XML code is processed to produce JavaScript. Following is an example of the JavaScript code that will be placed in the MainCalculation function of the E-Forms.

```
// Wizard generated code
_ThisForm["AdditionTotal0"].value = TparseFloat(_ThisForm["Field010"].value) +
   TparseFloat(_ThisForm["Field020"].value) ;
_ThisForm["AdditionTotal0"].value = FPNFormatZS(_ThisForm["AdditionTotal0"],
   _fldsFmt["AdditionTotal"]);
_ThisForm["SubtractionTotal0"].value = TparseFloat(_ThisForm["Field030"].value) -
   TparseFloat(_ThisForm["Field040"].value) ;
_ThisForm["SubtractionTotal0"].value = FPNFormatZS(_ThisForm["SubtractionTotal0"],
   _fldsFmt["SubtractionTotal"]);
_ThisForm["MultiplyTotal0"].value = TparseFloat(_ThisForm["Field050"].value) *
   TparseFloat(_ThisForm["Field060"].value) ;
_ThisForm["MultiplyTotal0"].value = FPNFormatZS(_ThisForm["MultiplyTotal0"],
   _fldsFmt["MultiplyTotal"]);
_ThisForm["DivideTotal0"].value = TparseFloat(_ThisForm["Field070"].value) /
   TparseFloatDiv(_ThisForm["Field080"].value) ;
_ThisForm["DivideTotal0"].value = FPNFormatZS(_ThisForm["DivideTotal0"],
   _fldsFmt["DivideTotal"]);
_ThisForm["ColumnTotal0"].value = TparseFloat(_ThisForm["AdditionTotal0"].value) +
   TparseFloat(_ThisForm["SubtractionTotal0"].value) +
   TparseFloat(_ThisForm["MultiplyTotal0"].value) +
   TparseFloat(_ThisForm["DivideTotal0"].value) +
_ThisForm["ColumnTotal0"].value = FPNFormatZS(_ThisForm["ColumnTotal0"],
   _fldsFmt["ColumnTotal"]);
// End Wizard generated code
```

Working with Stepped Columns



Calculation Wizard Demonstration Form

Addition with Steps			Row Totals
	Column1(0)	Column2(0)	Row1Total(0)
	Column1(1)	Column2(1)	Row1Total(1)
	Column1(2)	Column2(2)	Row1Total(2)
	Column Totals		Grand Total
	Column1Total(0)	Column2Total(0)	GrandTotal(0)

Figure 13-6: Examples/stepped_calculations.elf

Step-by-Step for Stepped Columns

1. To get started use the **Select Cursor Tool** to click on the first total field **Row1Total(0),** and go to the popup menu and select **Calculation Wizard.**

   This will bring up the dialog for creating a calculation statement for the "Row1Total" total field.

   English Description: "The Row1Total(0) Field equals Column1(0) plus Column2(0)."

   Assign Statement to Field: Row1Total

2. Click the **= Operator** as the first statement assignment.

3. Select the field from the list to be added, this would be **Column1.**

4. Click on the **[>] Button** to copy this assignment to the Assignment Statement column.

   Assignment Statement: = Column1

5. Click the **+ Operator** as the second statement assignment.

6. Select the field from the list to be added, this would be **Column2.**

7. Click on the **[>] Button** to copy this assignment to the Assignment Statement column.

   Assignment Statement: + Column2

8. Click on **[Statement Done]** and the dialog will close. Since all three fields involved are stepped, your calculation is complete for the addition of all stepped rows.

9. Now use the **Select Cursor Tool** to click on the first column total field **Column1Total(0),** and go to the popup menu and select **Calculation Wizard.**

This will bring up the dialog for creating a calculation statement for the "Column1Total" total field.

English Description: "The Column1Total(0) Field equals the sum of all steps of Column1(0)."

Assign Statement to Field: Column1Total

10. Working with stepped fields requires that you **check "Show Advanced"** features to expand the dialog.

11. Click the **[New Statement]** Button. It will place "Statement 1" in the Statement Number Column.

12. Click the **= Operator** as the first statement assignment.

13. Click on the **SumSteps** Function. (This will tell it to add every stepped field found in the next pick.)

14. Select the stepped field from the list to be added, this would be **Column1.**

15. Click on the **[>] Button** to copy this assignment to the Assignment Statement column.

Assignment Statement: = SumSteps(Column1)

16. The **Execute Sequence Number** for the Column1Total(0) Field needs to be a **2.** This will guarantee that it falls after the row totals (which have an execute sequence of 1) when the JavaScript code is generated.

17. Click on **[Statement Done]** and the dialog will close. Your calculation is complete for the Addition of all stepped fields in column 1.

18. Repeat steps 10 through 17 for Column2Total(0).

English Description: "The Column2Total(0) Field equals the sum of all steps of Column2(0)."

Assign Statement to Field: Column2Total

Assignment Statement: = SumSteps(Column2)

19. Repeat steps 2 through 8 for GrandTotal(0).

English Description: "The GrandTotal(0) Field equals Column1Total(0) plus Column2Total(0)."

Assign Statement to Field: GrandTotal

Assignment Statement: = Column1Total
+ Column2Total

20. The **Execute Sequence Number** for the GrandTotal(0) Field needs to be a **3.** This will guarantee that it falls after the column totals when the JavaScript code is generated.

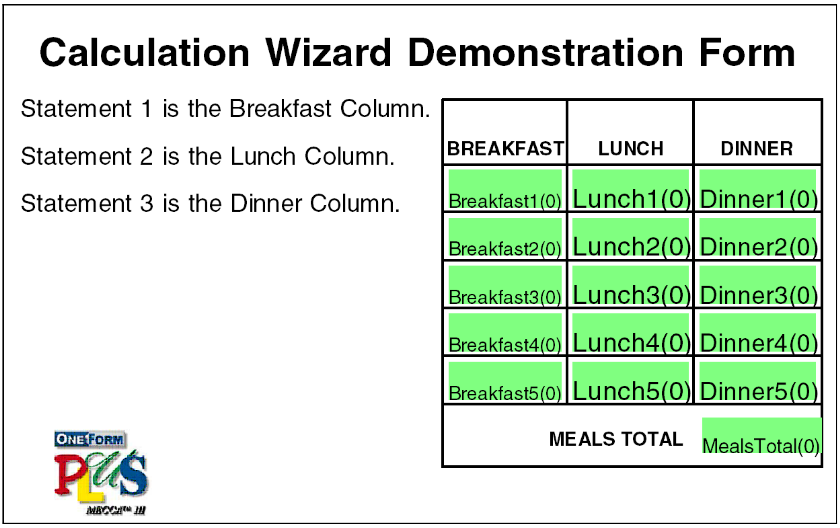
21. Click on **[Statement Done]** and the dialog will close. Your calculations are complete.

Wizard Generated Code

When the HTML or PDF E-Form is generated the following JavaScript code will be placed in the MainCalculation function of the E-Forms.

```
// Wizard generated code
for (var _I = 0; _I < 3; _I++) {
   _ThisForm["Row1Total" + _I].value = TparseFloat(_ThisForm["Column1" + _I].value)
      + TparseFloat(_ThisForm["Column2" + _I].value) ;
   _ThisForm["Row1Total" + _I].value = FPNFormatZS(_ThisForm["Row1Total" + _I],
      _fldsFmt["Row1Total"]);
}
_ThisForm["Column1Total0"].value = SumSteps("Column1", 3) ;
_ThisForm["Column1Total0"].value = FPNFormatZS(_ThisForm["Column1Total0"],
   _fldsFmt["Column1Total"]);
_ThisForm["Column2Total0"].value = SumSteps("Column2", 3) ;
_ThisForm["Column2Total0"].value = FPNFormatZS(_ThisForm["Column2Total0"],
   _fldsFmt["Column2Total"]);
_ThisForm["GrandTotal0"].value = TparseFloat(_ThisForm["Column1Total0"].value) +
   TparseFloat(_ThisForm["Column2Total0"].value) ;
_ThisForm["GrandTotal0"].value = FPNFormatZS(_ThisForm["GrandTotal0"],
   _fldsFmt["GrandTotal"]);
// End Wizard generated code
```

Working with Multiple Statements



Calculation Wizard Demonstration Form

Statement 1 is the Breakfast Column.

Statement 2 is the Lunch Column.

Statement 3 is the Dinner Column.

BREAKFAST	LUNCH	DINNER
Breakfast1(0)	Lunch1(0)	Dinner1(0)
Breakfast2(0)	Lunch2(0)	Dinner2(0)
Breakfast3(0)	Lunch3(0)	Dinner3(0)
Breakfast4(0)	Lunch4(0)	Dinner4(0)
Breakfast5(0)	Lunch5(0)	Dinner5(0)
	MEALS TOTAL	MealsTotal(0)

Figure 13-7: Examples/statements_calculation.elf

In this example the fields are not stepped, but individual. We will be creating a statement for each column to organize the calculation, since many individual fields need added. When the JavaScript Code is generated, the statements will be executed in their statement number order.

1. Go through the standard procedures of creating calculations for the MealsTotal(0) total field for Statement 1. The only difference will be that you will NOT click the [Done Statement] until you have completely finished all three statements.

English Description Statement 1: MealsTotal(0) Field equals the first column of Breakfast1(0) plus Breakfast2(0) plus Breakfast3(0) plus Breakfast4(0) plus Breakfast5(0).

Assign Statement to Field: MealsTotal

Assignment Statement: = Breakfast1
+ Breakfast2
+ Breakfast3
+ Breakfast4
+ Breakfast5

2. Now click **[New Statement].** This will save Statement 1 then show Statement 2 as the current statement. It also clears the Assignment Statement Column.

3. Go through the standard procedures of creating calculations for the MealsTotal(0) total field for Statement 2. Again, do NOT click the [Done Statement] until you have completely finished all three statements.

English Description Statement 2: MealsTotal(0) Field equals MealsTotal(0) (itself - which would be the current value it holds from statement 1), plus Lunch1(0) plus Lunch2(0) plus Lunch3(0) plus Lunch4(0) plus Lunch5(0).

Assign Statement to Field: MealsTotal

Assignment Statement: = MealsTotal
+ Lunch1
+ Lunch2
+ Lunch3
+ Lunch4
+ Lunch5

4. Now click **[New Statement].** This will save Statement 2 then show Statement 3 as the current statement. It also clears the Assignment Statement Column.

5. Go through the standard procedures of creating calculations for the MealsTotal(0) total field for Statement 3.

English Description Statement 3: MealsTotal(0) Field equals MealsTotal(0) (itself), plus Dinner1(0) plus plus Dinner2(0) plus Dinner3(0) plus Dinner4(0) plus Dinner5(0).

Assign Statement to Field: MealsTotal

Assignment Statement: = MealsTotal
+ Dinner1
+ Dinner2
+ Dinner3
+ Dinner4
+ Dinner5

6. Click on **[Statement Done]** and the dialog will close. Your calculation is complete for the Addition of all three columns.

Note: When reopening the Calculation Wizard, it will always display the last statement. Just click on the Statement Number to see the assignments of each one. You can add more statements simply by clicking the [New Statement] button.

Wizard Generated Code

When the HTML or PDF E-Form is generated the following JavaScript code will be placed in the MainCalculation function of the E-Forms.

```
// Wizard generated code
_ThisForm["MealsTotal0"].value = TparseFloat(_ThisForm["Breakfast10"].value) +
   TparseFloat(_ThisForm["Breakfast20"].value) +
   TparseFloat(_ThisForm["Breakfast30"].value) +
   TparseFloat(_ThisForm["Breakfast40"].value) +
   TparseFloat(_ThisForm["Breakfast50"].value) ;
_ThisForm["MealsTotal0"].value = TparseFloat(_ThisForm["MealsTotal0"].value) +
   TparseFloat(_ThisForm["Lunch10"].value) +
   TparseFloat(_ThisForm["Lunch20"].value) +
   TparseFloat(_ThisForm["Lunch30"].value) +
   TparseFloat(_ThisForm["Lunch40"].value) +
   TparseFloat(_ThisForm["Lunch50"].value) ;
_ThisForm["MealsTotal0"].value = TparseFloat(_ThisForm["MealsTotal0"].value) +
   TparseFloat(_ThisForm["Dinner10"].value) +
   TparseFloat(_ThisForm["Dinner20"].value) +
   TparseFloat(_ThisForm["Dinner30"].value) +
   TparseFloat(_ThisForm["Dinner40"].value) +
   TparseFloat(_ThisForm["Dinner50"].value) ;
_ThisForm["MealsTotal0"].value = FPNFormatZS(_ThisForm["MealsTotal0"],
   _fldsFmt["MealsTotal"]);
// End Wizard generated code
```

Miscellaneous Information and Examples

Inserting Wizard Generated Code into Form Preferences

If you need more advanced code than the Calculation Wizard currently supplies, you can continue to insert custom calculations in the Form Preferences of your form. You can include any Wizard Generated Code into the "function MainCalculation" by inserting the following comment :

```
    //Insert Calculation Point 1
```

This allows you to control where the Wizard Generated Code falls within the custom "function MainCalculation" script.

Examples of Insert Constant Value Option

Insert Constant Value [>]: This allows you to apply a constant value to an assignment. Select the Operator first, then type in the value and click the "Insert Constant Value" [>] button.

Example 1: Calculating mileage expenses with the rate being 48 cents.

English Description: "MileageAmount(0) Field equals MilesDriven(0) multipled by 0.480."

Assign Statement to Field: MileageAmount

Assignment Statement: = MilesDriven
* 0.480

Example 2: Calculate a percent of discount on total sales. Remember to select the Operator first, then type in the discount value and click the "Insert Constant Value" [>] button.

Assign Statement to Field: SalesDiscount

Assignment Statement: = (TotalSales
* DiscountPercent)
/ 100

In this example, notice that we have placed Left and Right Parentheses around two fields to group the operation before the next assignment takes place. This was done by highlighting the field in the statement column and doing a right-mouse click to get to the pop-up menu and selecting "Add Left Paren". Do the same with "Add Right Paren". *See Figure 13-2.*

INDEX