FEITH

Feith Forms iQ Designer
Version 9.2
User Guide

Updated 10/22/2015

Feith Forms iQ Designer
Version 9.2
User Guide

© Copyright 2015 Feith Systems and Software, Inc. All Rights Reserved. No part of this publication may be reproduced, transmitted, stored in a retrieval system, or translated into any language in any form by any means, without the written permission of Feith Systems and Software, Inc.

All information in this work is subject to change and reflects software current at the time of publication.

FDD and the FDD logo are trademarks of Feith Systems and Software, Inc. All other trademarks, product names and company names may be trademarks or registered trademarks of their respective holders.

Feith Systems and Software, Inc.
425 Maryland Drive
Fort Washington, PA 19034
Tel (215) 646-8000
Fax (215) 540-5495
www.feith.com

Licenses

ScintillaNET

ScintillaNET is based on the Scintilla component by Neil Hodgson.

ScintillaNET is released on this same license.

The ScintillaNET bindings are Copyright 2002-2006 by Garrett Serack <gserack@gmail.com>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

GARRETT SERACK AND ALL EMPLOYERS PAST AND PRESENT DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL GARRETT SERACK AND ALL EMPLOYERS PAST AND PRESENT BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The license for Scintilla is as follows:

Copyright 1998-2006 by Neil Hodgson <neilh@scintilla.org>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Table of Contents

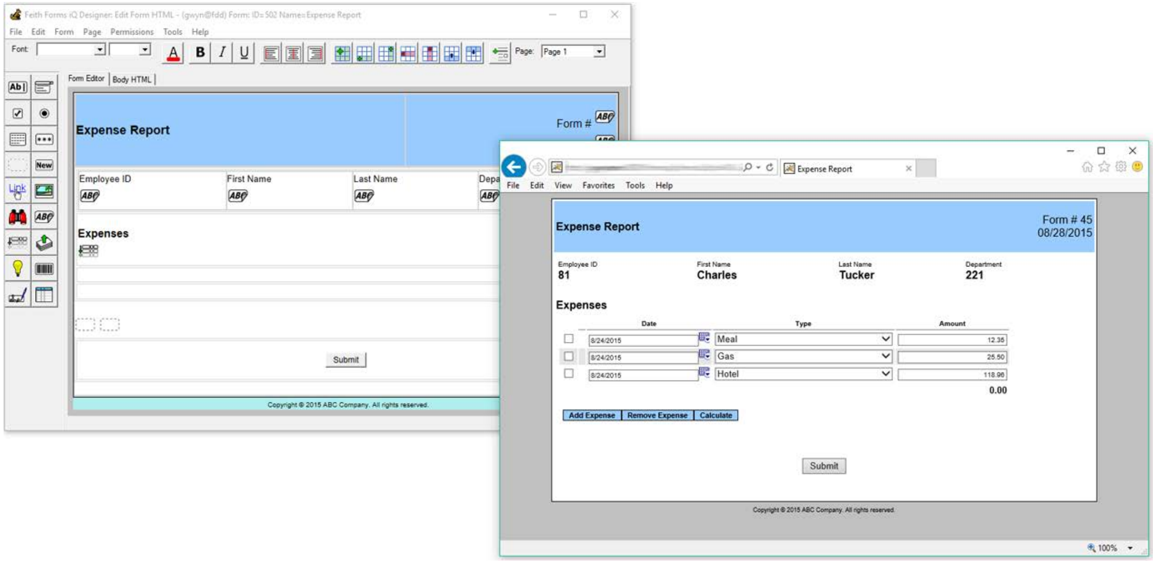
Introduction

Welcome

Feith Forms iQ Designer enables you to create a web form in seconds. You can even start a business process with Feith Workflow iQ. You don't need HTML knowledge to create an interactive, intuitive web form - Forms iQ does it for you! Once you are done designing, view and fill out forms using **Feith Forms iQ Server**.

Get started:

- Create a Form
- Format a Form
- Add Form Fields
- Autofill Form Fields
- Submit a Form


Feith Forms iQ Designer: Edit Form HTML - (gwyn@fdd) Form: ID=502 Name=Expense Report
File Edit Form Page Permissions Tools Help
Font:
Page: Page 1
Form Editor
Body HTML
Expense Report
Form #
Employee ID
First Name
Last Name
Depa
Expenses
Submit
Copyright © 2015 ABC Company. All rights reserved.
Expense Report
File Edit View Favorites Tools Help
Add Expense
Remove Expense
Calculate
Submit
100%

Expense Report

Form # 45
08/28/2015

Employee ID	First Name	Last Name	Department
81	Charles	Tucker	221

Expenses

	Date	Type	Amount
□	8/24/2015	Meal	12.36
□	8/24/2015	Gas	25.50
□	8/24/2015	Hotel	118.96
			0.00

Copyright © 2015 ABC Company. All rights reserved.

What's New?

- **Body HTML Tab:** Modify the HTML of the form right inside the Designer. See Form Tabs for more information.
- **Syntax-Highlighting:** Many of the "code" interfaces, where SQL, JavaScript, and HTML are edited, now have syntax-highlighting.
- **Designer Permissions:** In order to create file cabinets and lookup tables on import of a form, you must be a Feith Admin (on Oracle) or a Database Admin (on MS SQL Server) and have the necessary task permissions. See Import for more information.
- **Line Numbers:** A few "code" interfaces, such as Custom JavaScript, now have line numbers.
- **Interface to Add Open With Apps:** An interface was added inside the Designer to manage the list of editors you like to use for the form's HTML and Custom JavaScript. See Open Page With Other Editor for more information.
- **Change Dynamic List Button Labels:** Change the text labels of the standard dynamic list buttons in the **Dynamic List** properties dialog. See Dynamic List for more information.
- **Better Error Messages:** Error messages in the Designer now provide more information to better help you troubleshoot and report problems.
- **Use Feith Dev CSS and JS:** If you have some CSS and JavaScript objects in Feith Developer, you can use them on your Forms iQ forms too. Just assign them to the form in **Form Properties**. See Set Form Properties for more information.
- **Keep External Editor Open:** If you are changing the form's HTML in another editor, turn on the **Allow continued editing of form after "open with"** option so you don't have to close the editor to see your changes in the form. Just make a change in the editor, save in the editor, then click **Pull HTML** in the Designer to get the updated HTML from the other editor. See Global Defaults and Open Page With Other Editor for more information.
- **Multi-line Comments and Messages:** The fields where you type in a form **Comment**, **Success Message**, and **Sign-In Failure Message** have all been made multi-line.

Frequently Asked Questions

When creating a form, the file cabinet I want to select is not in the list?

> You must have resource permission to the file cabinet in order to build a form on it. Resource permissions are set in Feith Control Panel. See Feith Control Panel User Guide for more information.

When managing forms, I can't find the form I need in the list?

> Check the following:
>
> - The view options in the **Open Form** dialog are set so that the form should be in the list. See Open Form View Options for more information.
> - You must have resource permission to the file cabinet in order to view a form built on it. Resource permissions are set in Feith Control Panel. See Feith Control Panel User Guide for more information.

How do I set up "cascading lookups" where the value in one field determines the list of values to select in another field?

> The field whose list of values change based on the value of another field should have its **Option List Properties** set up with **Advanced** SQL. The SQL would include a field token for the other field. For example:

```
select city from fdd.vendors where state='{state}'
```

The "other table" I want to select for an autofill or storage object is not available in the list?

> The "other" table must be specified in the **Other Tables List** in order to use it for an autofill or storage object. See Maintain Other Tables List for more information.

The external editor I want to use is not listed under Page>Open With?

> Make sure you added your preferred editor to the Designer's "Open With" applications list. See Open Page With Other Editor for more information.
>
> If the editor is already in the Open With applications, make sure the editor's executable file actually exists at the specified path. If it does not, it will not display in the list under **Page>Open With**.

I formatted the form's HTML, indenting with TABs and such, but my formatting was lost?

> Formatting HTML with TAB indents and carriage returns will not be preserved once the form is saved. This is due to the transformation that occurs when a form is saved, which causes extra white space to be removed.

When I view a form in the browser, the wrong form is displayed?

> Is the form identified in the URL by form_name? It's possible another form may have the same name. We recommend giving forms unique names. See Form Sets and Versions for more information.

How do I view a hidden form in the browser?

> If you hid your form, you can look at it in the browser by identifying it with the form_id. See View Form in Browser for more information.

Where did my Action Settings go?

> When you open a form in Designer 9.2.4.1 or later, any **Actions Settings** you previously set up are converted to Custom JavaScript. See Custom JavaScript for more information.

How do I hide and show fields on the form now that the Action Settings is gone?

> Write the hide and show logic in Custom JavaScript. See Hide and Show Examples for more information.

The CDATA section in my Custom JavaScript is commented out with a // and says "Upgrade Warning"?

When you open an existing form in Designer 9.2.4.1 or later, the Designer will look for CDATA sections in your Custom JavaScript and put a ***//*** in front of CDATA start and end tags. This comments out the CDATA in the JavaScript and an "Upgrade Warning" statement is also included in the comment.

The CDATA is commented out so it will be ignored in the JavaScript. However, it will still take effect in the XML. The CDATA probably needs to be ignored in the JavaScript, otherwise the JavaScript may not run properly or at all.

Note that CDATA used to be used as a workaround to avoid errors caused by special characters, such as less than <, but that is no longer necessary as the Designer will now handle such characters appropriately without a CDATA section.

Also note that this change is only for existing forms. If you are creating a new form in Designer 9.2.4.1 or later, and you enter a CDATA section in your Custom JavaScript, the Designer will not change the CDATA at all.

When opening or importing a form, I get a message that my form references Feith Developer objects that are not available?

One or more Feith Developer CSS and JavaScript objects were assigned to the form and do not exist on this FDD database. See Set Form Properties for more information.

Click **Yes** to open the form anyway and lose the references to those objects. Click **No** to cancel.

The Global Defaults option under the Tools menu is disabled?

You must have no form open in order to access Global Defaults. Close the open form and try again.

Tips and Tricks

- Change default logos, messages, and other form settings in **Tools>Global Defaults**. See Global Defaults for more information.

- When naming a form, we recommend giving the form a unique name, as forms can be identified in URLs using their names. See Save Form for more information.

- If you change the view options almost every time you go to open a form, set new defaults to your liking. See Open Form View Options for more information.

- To insert a line break between a label and a form field, type the label, then press **SHIFT+ENTER**, then add the field. This will reduce the amount of space added between the two lines.

- Use the **Field Grid** under the **Form** menu to set some properties for multiple fields at once, such as mandatory, mouseover text, and more. See Common Field Properties for more information.

- Generate form fields from columns in an "other" table to which you want to store using **Form>Add Fields From Table**. See Add Form Fields From Table for more information.

- In "code" fields, you can format your text using **TAB** to indent and **SHIFT+TAB** to un-indent.

- If you have a lot of hidden fields on your form, instead of using the **Hidden** form field, put an invisible table set to display as **none** at the bottom of your form. You can put anything in there and it won't display on the live form, such as textboxes and text labels for your fields, and so on. See Add and Edit Tables for more information.

- You can ask the Designer to automatically create columns in the dynamic list based on the columns in an autofill object's source, such as a lookup table. After selecting the autofill's **Source**, just click **Auto-create List**. See Autofill Dynamic List for more information.

- Use **<br>** tags to make a multi-line **Success Message** or **Sign-In Failure Message**. See Set Form Properties for more information.

- Add a **Save** button to a form with a sign-in page or login permissions, which lets users partially fill out the form and save it with their credentials so they can finish filling it out later. See Add Buttons for more information.

- When you save a form, especially a form in production, we recommend entering a **Comment** listing the changes you made, perhaps along with your initials. See Save Form for more information.

- In the **Open Form** dialog, you can select a form set and click **View Details** to see the user, comments, and more information for all the forms in that form set.

- Hide a form version from end users so they cannot use it. You can do this in the **Open Form** dialog by selecting a form version within a form set and clicking **Hide**.

- If you want to view the hidden version of a form in the browser, choose **Use Form ID** when you select **Form>View in Browser**.

- When providing the link to a form, you may want to use a link that identifies the form by form set ID or form name. That way, when you save a new version of the form, which will be a new form ID, you don't have to send out a new link.

- Set up a custom printer friendly view that users can use after submitting a form. See Define Alternate Print View for more information.

Login

To login to Forms iQ Designer:

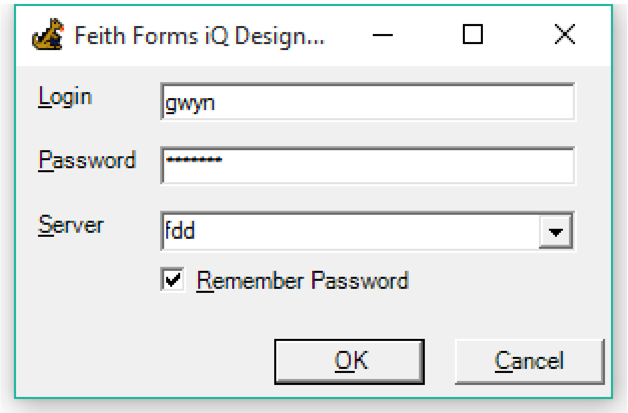
1. In your start menu programs list, select **Forms iQ Designer** under **Feith Systems**. The **Feith Forms iQ Designer Login** dialog opens.

2. Enter your **Login** name.

3. Enter your **Password**.

4. Enter you FDD database **Server**.

5. Optionally turn on **Remember Password** for Forms iQ Designer to remember your password and fill it in for you.



Login gwyn

Password ********

Server fdd

☑ Remember Password

6. Click **OK**. You are logged into Forms iQ Designer.

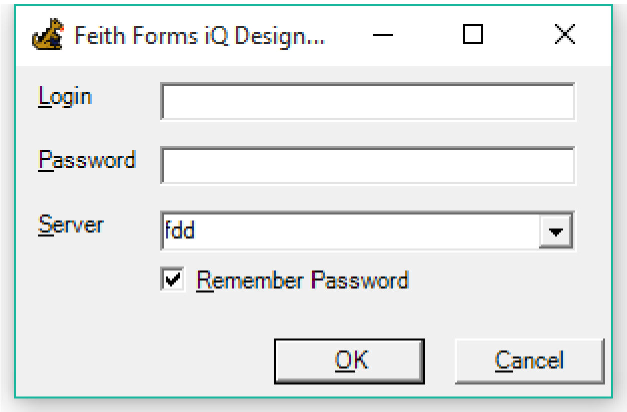
Single Sign-On

To login to Forms iQ Designer if your system is configured for Single Sign-On:

1. In your start menu programs list, select **Forms iQ Designer** under **Feith Systems**. The **Feith Forms iQ Designer Login** dialog opens.

2. Leave the **Login** and **Password** fields empty.

3. Enter you FDD database **Server**.



Server: fdd

☑ Remember Password

4. Click **OK**. You are Single Signed-On to Forms iQ Designer.

Get Started

Create a New Form

Create a new form for your users to fill out.

To create a form:

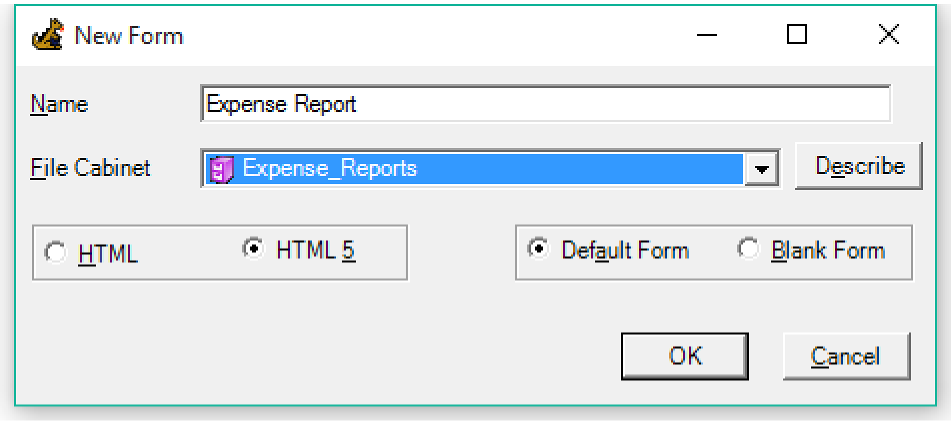
1. Select **File>New**. The **New Form** dialog opens.
2. Enter the form **Name**.

   Tip: We recommend giving the form a unique name, as forms can be identified in URLs using their names.
3. Select the **File Cabinet** on which to build the form. When the form is submitted, it will go into this file cabinet as a document.

   Standard file cabinets are blue, workflow base file cabinets are purple, and virtual file cabinets are red. Click **Describe** to get more information on the file cabinet's structure.

   Note: Depending on a virtual file cabinet's structure, it may or may not be updatable. Forms should not be built on non-updatable virtual file cabinets. If a form is built on a non-updatable virtual file cabinet, users will receive an error when trying to submit the form.
4. Choose what kind of HTML you want in the form:
   - **HTML:** No HTML doctype defined. We recommend **HTML 5** instead.
   - **HTML 5:** The <!DOCTYPE html> HTML doctype is defined. In the live form, some elements and attributes will be replaced with valid HTML5 elements and attributes.
   - **Tip:** If you change your mind later, you need to adjust the HTML by hand to agree with the HTML doctype.
5. Choose your starting point in the form:
   - **Default Form:** A basic form is created with fields that store to the base file cabinet.
   - **Blank Form:** An empty form with nothing in it is created.

   Note: The form must have at least one field that stores to the file cabinet.



New Form

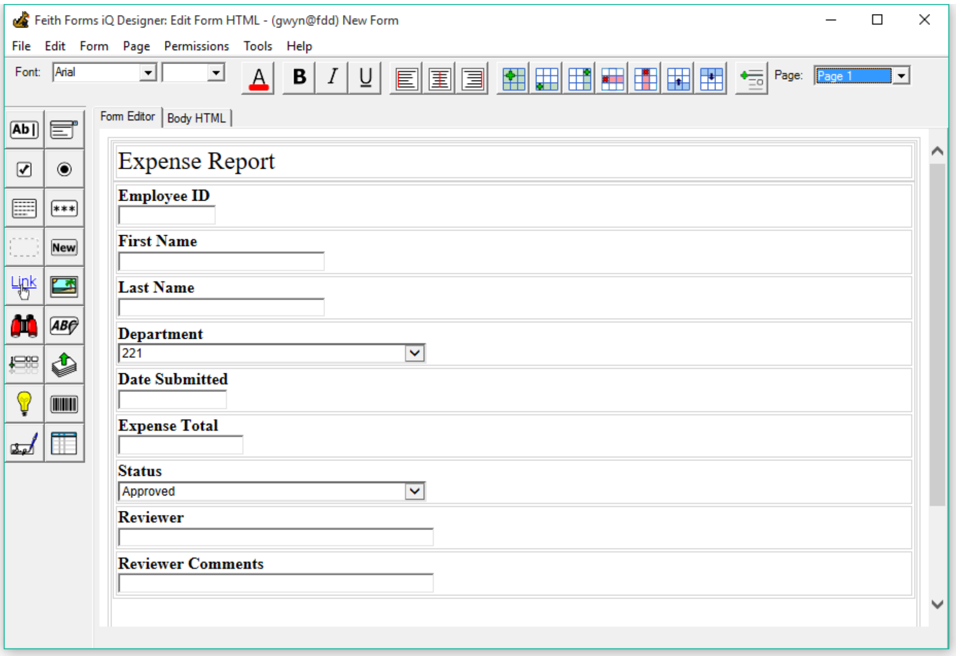
Name: Expense Report

File Cabinet: Expense_Reports

○ HTML ● HTML 5

● Default Form ○ Blank Form

If you chose to generate a **Default Form** with form fields that store to the file cabinet, it would look something like this.



Expense Report

Employee ID

First Name

Last Name

Department: 221

Date Submitted

Expense Total

Status: Approved

Reviewer

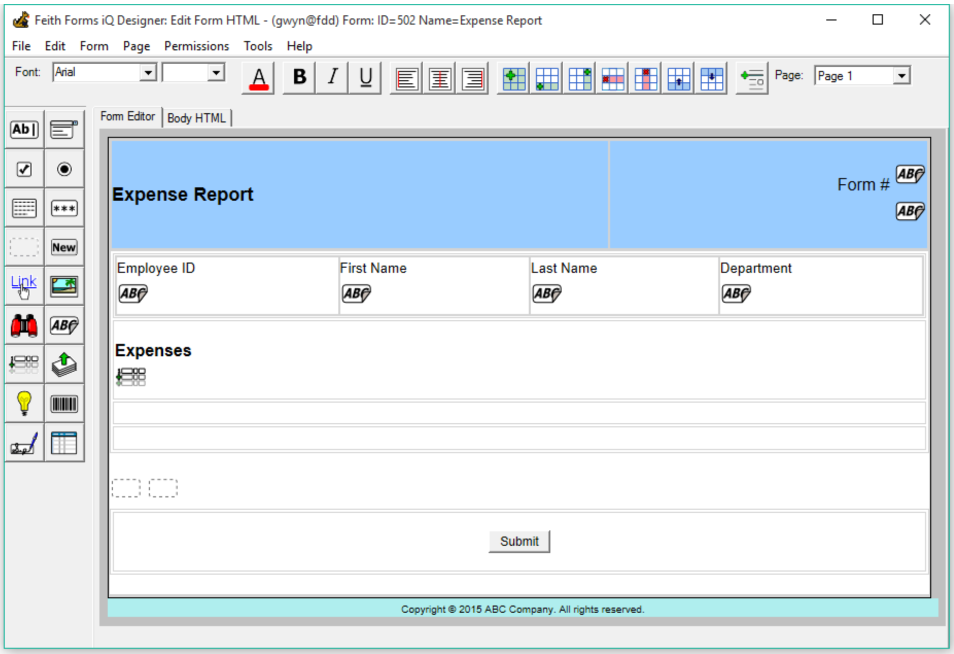
Reviewer Comments

6. Add fields, pages, autofill, storage, and more as desired.

   As you develop the form, save your work by selecting **File>Save Publish** with the **Visible** option unchecked (so that the form is not available to users). See Save Form for more information.

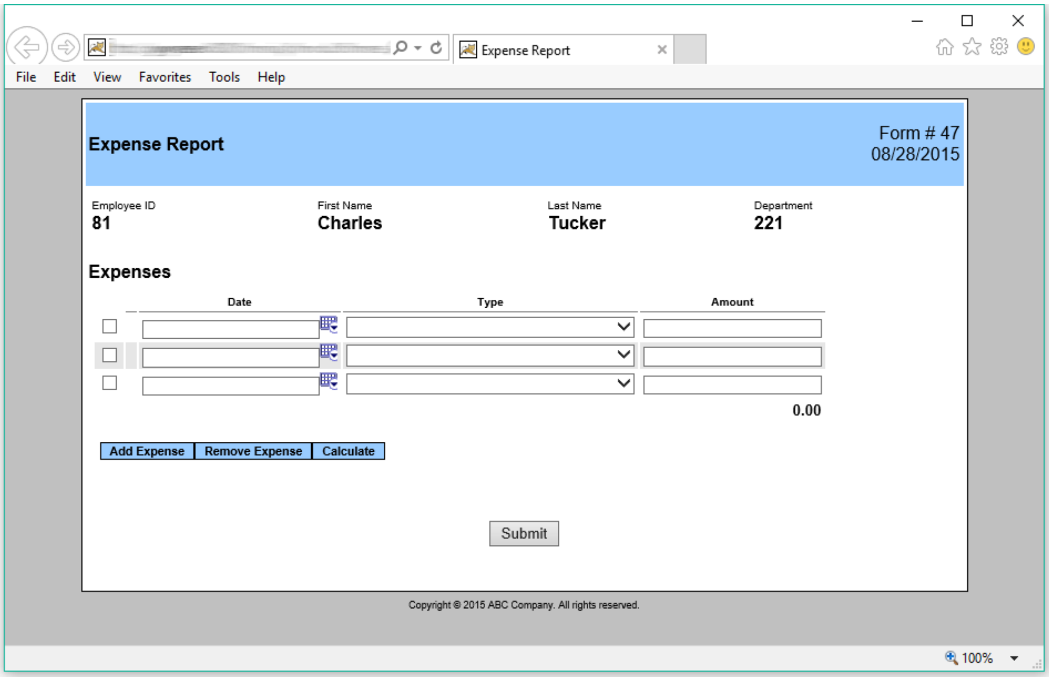
   See relevant sections for information:

   - Add form fields
   - Format the form, add tables, and add buttons
   - Modify the form's HTML directly in the Designer or your preferred editor
   - Add another form page or a Sign-in page
   - Autofill fields with information from your data
   - Store fields' data to various locations
   - Validate information entered in fields
   - Choose who can do what on your form with permissions
   - Calculate the value for a field
   - Customize your form further with your own JavaScript


Feith Forms iQ Designer: Edit Form HTML - (gwyn@fdd) Form: ID=502 Name=Expense Report
File Edit Form Page Permissions Tools Help
Font: Arial
Page: Page 1
Form Editor
Body HTML
Expense Report
Form #
Employee ID
First Name
Last Name
Department
Expenses
Submit
Copyright © 2015 ABC Company. All rights reserved.

7. When your form design is complete and ready for users to fill out, select **File>Save Publish** and make sure the **Visible** option is checked on. See Save Form for more information.

8. We recommend you fill out and submit your form to make sure it works correctly. Select **Form>View in Browser** to view your live form in the browser and fill it out. See Submit Form for more information.



Expense Report

Form # 47
08/28/2015

Employee ID	First Name	Last Name	Department
81	Charles	Tucker	221

Expenses

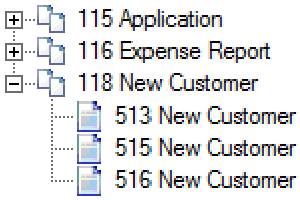
	Date	Type	Amount
□			
□			
□			
			0.00

Copyright © 2015 ABC Company. All rights reserved.

Form Sets and Versions

When you create a new form, a **form set** is created for the new form. A form set can have multiple form **versions** within it.

For example, here are three form sets, one of which is expanded and displaying its three form versions, as shown in the Open Form dialog.



115 Application
116 Expense Report
118 New Customer
513 New Customer
515 New Customer
516 New Customer

When you save changes you made to an existing form, you can choose to:

- Overwrite the form version you modified

  Note: If documents or drafts are saved with the current version of the form, the overwrite option is not available.

- Create a new version in the form set

- Create an entirely new, separate form set

- See Save Form for more information on these options.

The name of a form set is the name of the highest form version in the form set. If you rename the highest version, the form set name will be updated to reflect your change.

When you view a form in the browser, the form is identified in the URL in various ways:

- The name of the form set. The highest, visible form version in the form set will be viewed.

- The ID of the specific form version. The specific form version will be viewed.

  Tip: If you unchecked **Visible** to hide the form from users, but you want to view it yourself in the browser, choose **Use Form ID** so you can see the hidden form.

- The ID of the whole form set. The highest, visible form version in the form set will be viewed.

- See View Form in Browser for more information on these options.

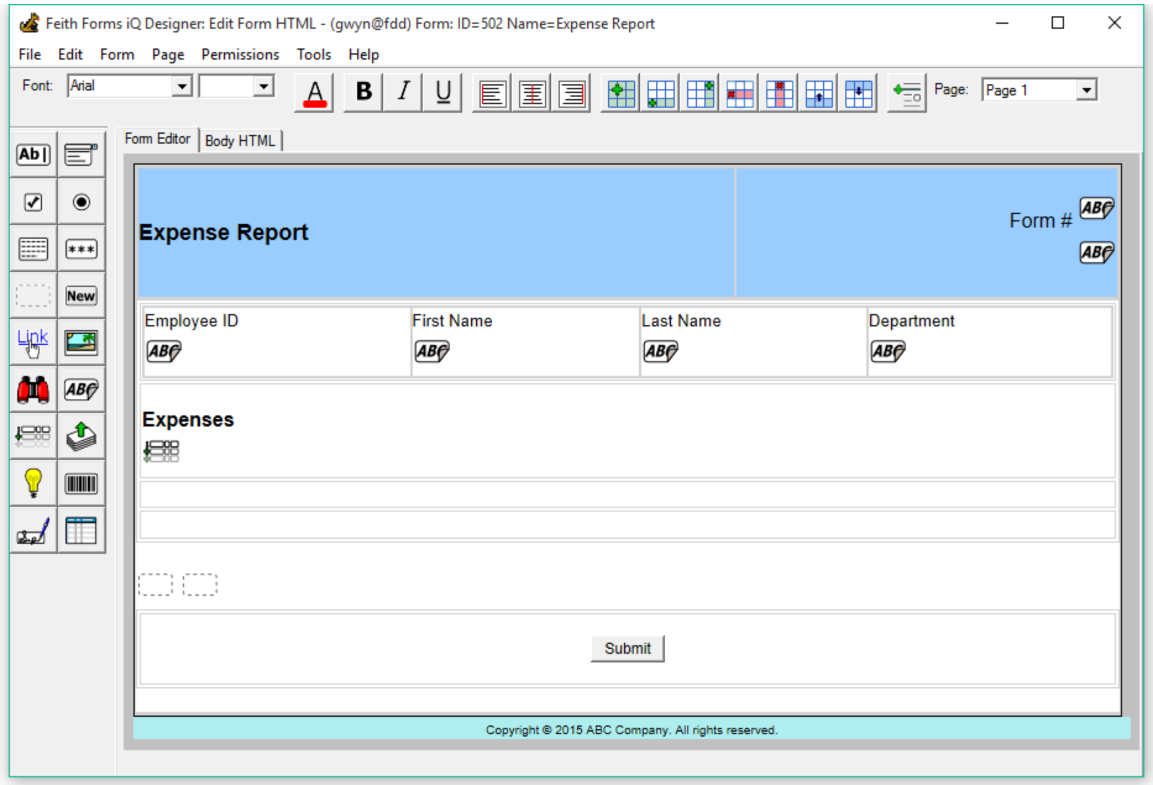
Tip: When providing the link to a form, you may want to use a link that identifies the form by form set ID or form name. That way, when you save a new version of the form, which will be a new form ID, you don't have to send out a new link.

Form Tabs

You can modify your form in the **Form Editor** tab, which is a WYSIWYG interface, or in the **Body HTML** tab, where you can modify the HTML of the form directly.

Form Editor

A WYSIWYG interface for modifying the form. Format the form, add tables, add buttons, and add fields.



Feith Forms iQ Designer: Edit Form HTML - (gwyn@fdd) Form: ID=502 Name=Expense Report

Font: Arial

Page: Page 1

Expense Report

Form #

Employee ID	First Name	Last Name	Department

Expenses

Submit

Copyright © 2015 ABC Company. All rights reserved.

Body HTML

A text editor where you can modify the HTML of the form that is inside the body tag. Click **Commit** to save your changes or click **Revert** to undo your changes since the last time you committed.

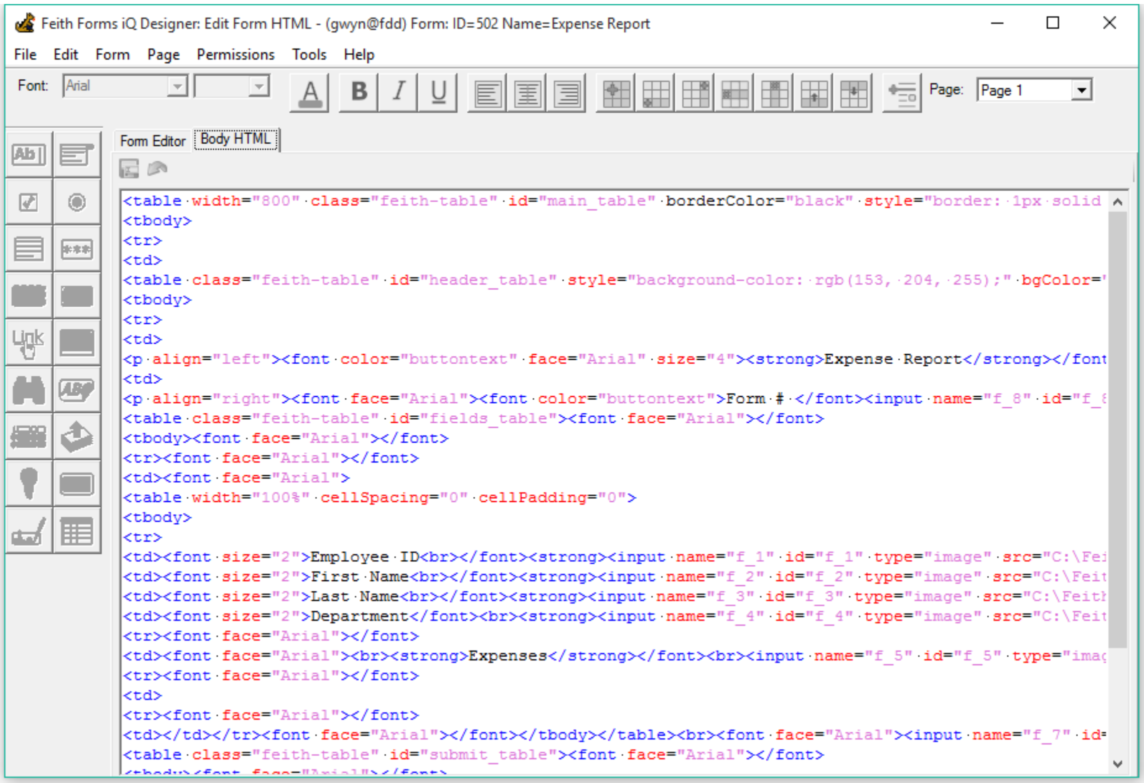
Notes:

- This feature is intended to be used for formatting changes only.

- A form with the HTML doctype **<!DOCTYPE html>** will have some elements and attributes replaced with valid HTML5 elements and attributes on the live form in the browser. The HTML5 elements and attributes are not displayed in the HTML you view in the **Body HTML** tab. See Set Form Properties and Create a New Form for more information.

- Use caution. Editing or deleting a form field's ID or name can invalidate the HTML and may prevent saving of the form. Adding or deleting fields may cause the form to break.

- Formatting HTML with TAB indents and carriage returns will not be preserved once the form is saved.


Feith Forms iQ Designer: Edit Form HTML - (gwyn@fdd) Form: ID=502 Name=Expense Report
File
Edit
Form
Page
Permissions
Tools
Help
Font:
Arial
Page:
Page 1
Form Editor
Body HTML

```
<table width="800" class="feith-table" id="main_table" borderColor="black" style="border: 1px solid
<tbody>
<tr>
<td>
<table class="feith-table" id="header_table" style="background-color: rgb(153, 204, 255);" bgColor="
<tbody>
<tr>
<td>
<p align="left"><font color="buttontext" face="Arial" size="4"><strong>Expense Report</strong></font
<td>
<p align="right"><font face="Arial"><font color="buttontext">Form # </font><input name="f_8" id="f_
<table class="feith-table" id="fields_table"><font face="Arial"></font>
<tbody><font face="Arial"></font>
<tr><font face="Arial"></font>
<td><font face="Arial">
<table width="100%" cellSpacing="0" cellPadding="0">
<tbody>
<tr>
<td><font size="2">Employee ID<br></font><strong><input name="f_1" id="f_1" type="image" src="C:\Fei
<td><font size="2">First Name<br></font><strong><input name="f_2" id="f_2" type="image" src="C:\Feit
<td><font size="2">Last Name<br></font><strong><input name="f_3" id="f_3" type="image" src="C:\Feith
<td><font size="2">Department</font><br><strong><input name="f_4" id="f_4" type="image" src="C:\Feit
<tr><font face="Arial"></font>
<td><font face="Arial"><br><strong>Expenses</strong></font><br><input name="f_5" id="f_5" type="imag
<tr><font face="Arial"></font>
<td>
<tr><font face="Arial"></font>
<td></td></tr><font face="Arial"></font></tbody></table><br><font face="Arial"><input name="f_7" id=
<table class="feith-table" id="submit_table"><font face="Arial"></font>
<tbody><font face="Arial"></font>
```

The Menus

Forms iQ Designer has the following menus: File, Edit, Form, Page, Permissions, Tools, and Help.

File Menu

MENU OPTION	ACTION
New (CTRL+N)	Create a new form. See Create a New Form for more information.
Open (CTRL+O)	Open an existing form. See Open a Form for more information.
Save/Publish (CTRL+S)	Save the form and, if the form is marked **Visible**, publish it as a form available for users to fill out. If you are modifying an existing form, you can create a new version or overwrite the modified version. See Save Form and Form Sets and Versions for more information.
Quick Save	Save the form without any further prompts. Whether Quick Save overwrites an existing version or creates a new version depends on your settings in Global Defaults. See Global Defaults for more information. **Note:** We recommend using Quick Save on a form in heavy development. We recommend using the standard **File>Save/Publish** when making small changes to a form in production, so that you have the opportunity to make comments and choose the appropriate save options.
Close	Close the open form.
Export (to Local Machine)	Export a form design to a .html file, then use the file to import the form design to another FDD database. See Export a Form for more information.
Import (from Local Machine)	Import a .html export file containing a form design to the desired FDD database. See Import a Form for more information.
Forms iQ Report	View a report that details the design of the form, including the form fields' properties, dynamic list columns' properties, permissions, and more.
Print (CTRL+P)	Print the form design.
Reconnect	Close Forms iQ Designer and connect again. This is useful for re-establishing a dropped database connection, or for switching your user login or database.
Exit	Close the Forms iQ Designer application and log out of the FDD database.
Recent Forms	At the bottom of the File menu the recently-modified forms are listed. Select a form from the list to open it.

Edit Menu

MENU OPTION	ACTION
Cut	Move selected text or fields to the clipboard.
Copy	Copy selected text or fields to the clipboard.
Paste	Paste cut or copied text or fields from the clipboard.
Undo	Reverse the last change made to the form design.
Redo	Redo the last change made to the form design that was undone.
Add Copyright Symbol	Insert a copyright symbol.

Form Menu

MENU OPTION	ACTION
Properties	Set various properties on the form that change how it looks of behaves. See Set Form Properties for more information.
Field Grid	View and manage a few basic properties of all the fields on your form in one interface. See Common Field Properties for more information.
Data Storage	Store form field data to the desired location. See Data Storage for more information.
Autofill	Automatically fill fields with data as the user works on the form. See Autofill for more information.
Validations	Validate data entered in form fields to make sure the data is correct before the form is submitted. See Validations for more information.
Images	Add images for use with smart icon fields. See Smart Icons for more information.
Data Cloning Setup	Let users clone data from an existing form to a new form. See Data Cloning for more information.
Add Fields From Table	Generate form fields from an external table to which you want the new form fields to store. See Add Form Fields From Table for more information.
Details	View form details, including the form name, the date the form was created, and the date the form was last modified. See View Form Details for more information.
View in Browser	View the live form in the browser using Forms iQ Server. See View Form in a Browser for more information.

Page Menu

MENU OPTION	ACTION
Open With	Open the form in another editor program, such as a text editor, to format the form as desired. See Open Page With Other Editor for more information.
Define Alternate Views	Define an alternate print view. See Define Alternate Print View for more information.
Add Sign-in Page	Add a sign-in page to the form. See Sign-in Page for more information.
Add Main Page	Add a new page to the form. See Multi-page Forms for more information.
Delete Current Page	Remove the selected page from the form. See Multi-page Forms for more information.

Permissions Menu

MENU OPTION	ACTION
Login	Restrict access to the form so only FDD users can login and submit the form. See Login Permissions for more information.
Go Live	Restrict access to go live to certain users. See Go Live Permission for more information.
History	Restrict access to show history to certain users. See Show History Permission for more information.
Field Set	Restrict access to some fields so that only certain users and groups can read or write to them. See Field Set Permissions for more information

Tools Menu

MENU OPTION	ACTION
User-defined JavaScript	Add custom JavaScript to a form. See Add Custom JavaScript to a Form for more information.
Create/Maintain Other Tables List	Maintain the list of other tables for use with option lists, autofills, and storage objects. See Maintain Other Tables List for more information.
Maintain Open With Apps	Define which applications are available when you select **Page>Open With**. See Open Page With Other Editor for more information.
Global Defaults	Set defaults for various form properties that control how the form looks and behaves. See Global Defaults for more information.

Help Menu

MENU OPTION	ACTION
Feith Forms iQ Designer Help (F1)	View Forms iQ Designer's help file.
About Feith Forms iQ Designer	View Forms iQ Designer version, build, and copyright information.

The Toolbars

Forms iQ Designer has the following toolbars: Standard Toolbar and Field Toolbar.

Standard Toolbar

BUTTON	ACTION
Arial ▾	Change the font of the selected text.
2 (10 pt) ▾	Change the font size of the selected text.
A	Change the font color of the selected text.
B	Boldface the selected text.
I	Italicize the selected text.
U	Underline the selected text.
≡	Left align the selected text.
≡	Center align the selected text.
≡	Right align the selected text.
▦	Add a new table to the form.
▦	Add a new row to the selected table.
▦	Adds a new column to the selected table.
▦	Delete the selected row.
▦	Delete the selected column.
▦	Move the selected row up in the table.

	Move the selected row down in the table.
	Add a horizontal bar to the form.
Page 1 ▾	Change the current page selection.

Field Toolbar

Add fields to your form. Place your cursor in the form where you want to add the field then click the desired button in the Field Toolbar. See Add Form Fields for more information.

BUTTON	ACTION
Ab|	Add textbox. See Textbox for more information.
	Add select field. See Select for more information.
☑	Add check box. See Check Box for more information.
◉	Add radio button. See Radio Button for more information.
	Add text area or rich text editor. See Text Area and Rich Text Editor for more information.
***	Add password field. See Password for more information.
	Add hidden field. See Hidden for more information.
New	Add button. See Add Buttons for more information.
Link	Add hyperlink. See Hyperlink for more information.

🖼	Add image. See Image for more information.
	Add lookup field. See Lookup for more information.
ABC	Add label field. See Label for more information.
	Add dynamic list. See Dynamic List for more information.
	Add file upload field. See File Upload for more information.
💡	Add smart icon. See Smart Icon for more information.
	Add barcode. See Barcode for more information.
	Add signature field. See Signature for more information.
	Add data grid. See Grid for more information.

Properties and Formatting

Set Form Properties

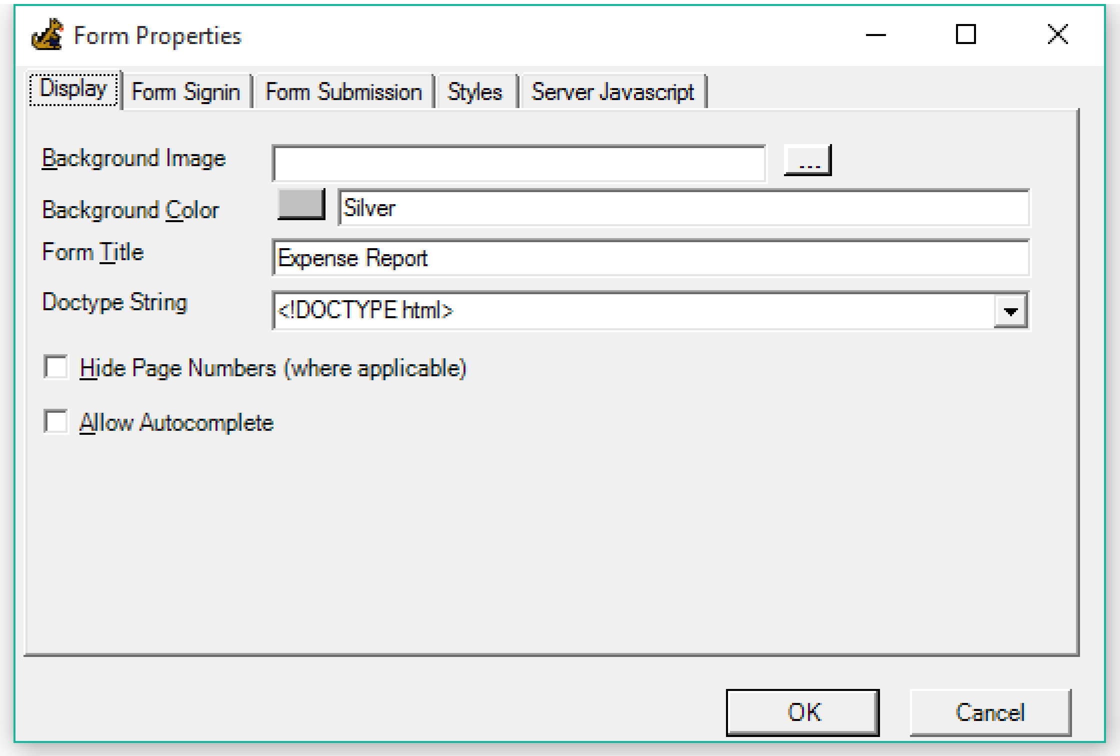
Set various properties on the form that change how it looks of behaves. For example, you can set the form title and customize the success message that appears when a form submits.

To set the form properties:

1. Select **Form>Properties**. The **Form Properties** dialog opens.
2. Change the properties on any of the tabs:
   - Display
   - Form Sign-in
   - Form Submission
   - Styles
   - Server Javascript
3. Click **OK** to save changes.

You can change the default form properties by setting the global defaults. See Global Defaults for more information.

Display

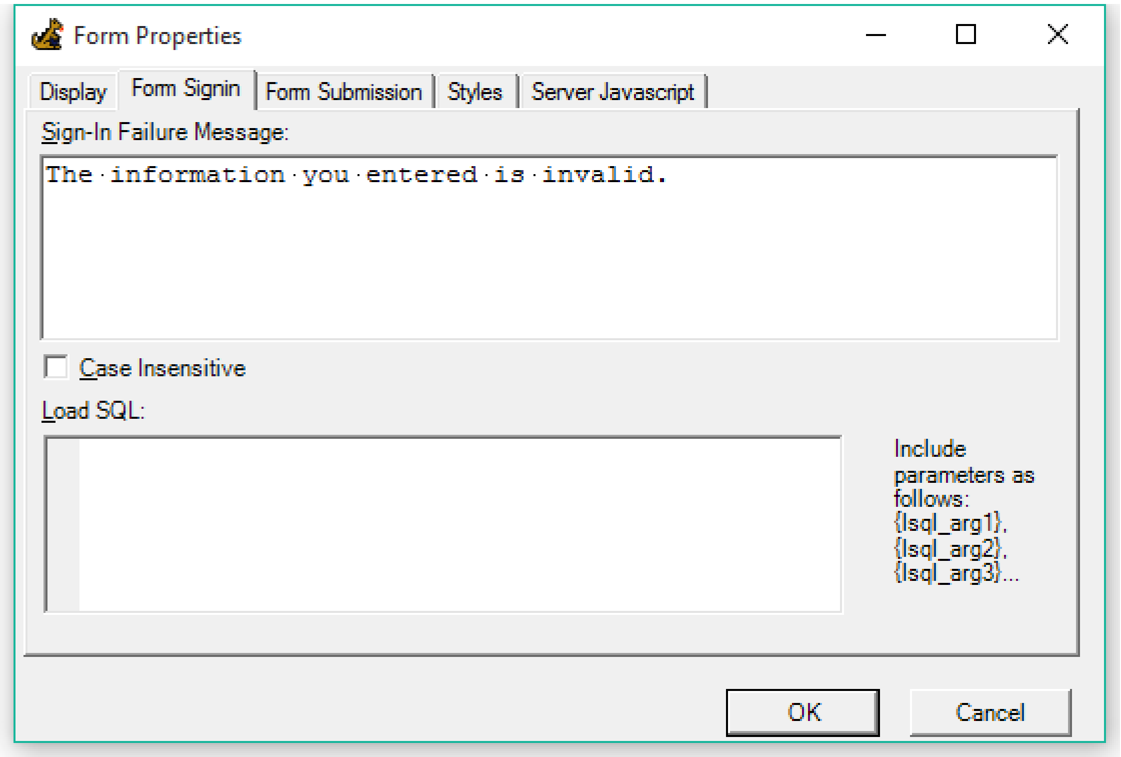


Background Image:
Background Color: Silver
Form Title: Expense Report
Doctype String: <!DOCTYPE html>
□ Hide Page Numbers (where applicable)
□ Allow Autocomplete

The **Display** properties are:

- **Background Image:** Optionally set an image to display in the background of the entire form. Click the browse button to select a .gif or .jpg file to use as the background image.
- **Background Color:** The background color of the entire form. It defaults to white. Click to open the **Color** palette to choose a background color, or type in an HTML color or hex code.
- **Form Title:** The title of the form that displays in the browser's title bar or tab.
- **Doctype String:** The HTML doctype of the form. Select from the provided doctypes or type in your own. The <!DOCTYPE html> doctype, for HTML5, is recommended.
- **Hide Page Numbers:** Turn on and page numbers on multi-page forms will not appear on the form. This setting does not apply to single-page forms. See Multi-page Forms for more information.
- **Allow Autocomplete:** Turn on and, if users have configured autocomplete options in their browser settings, form fields can be autocompleted by the browser. Note autocomplete is a browser-specific setting and may vary between browsers and versions.

Form Sign-in



Sign-In Failure Message:

```
The information you entered is invalid.
```

□ Case Insensitive

Load SQL:

Include parameters as follows: {lsql_arg1}, {lsql_arg2}, {lsql_arg3}...

The **Form Sign-in** properties are:

- **Sign-In Failure Message**: The message displayed to the user if verification fails when attempting to sign in to a form. The sign-in page is optional. This setting defaults to **The information you entered is invalid**. See Sign-in Page for more information.

  Tip: If you want to have a multi-line message, use **<br>** tags.

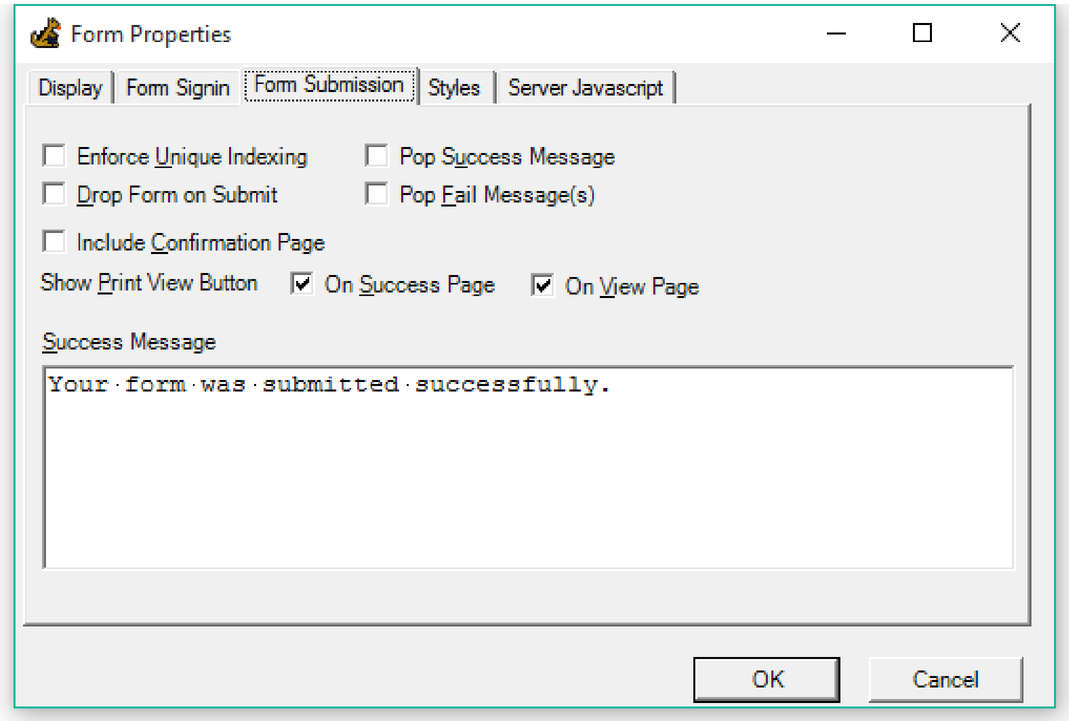
- **Case Insensitive**: Turn on and sign-in field verification will be case-insensitive.

- **Load SQL:** An optional feature to run SQL upon launching a form. The SQL parameters defined here are called from a URL configured with the SQL parameters. Contact Feith Support for more detailed information on how to configure Load SQL.

  Notes:

  - Load SQL can only be run on a form with a sign-in page.
  - Load SQL always runs, even when the launched form presents an error.

Form Submission



□ Enforce Unique Indexing
□ Pop Success Message
□ Drop Form on Submit
□ Pop Fail Message(s)
□ Include Confirmation Page
Show Print View Button ☑ On Success Page ☑ On View Page

Success Message

Your·form·was·submitted·successfully.

The **Form Submission** properties are:

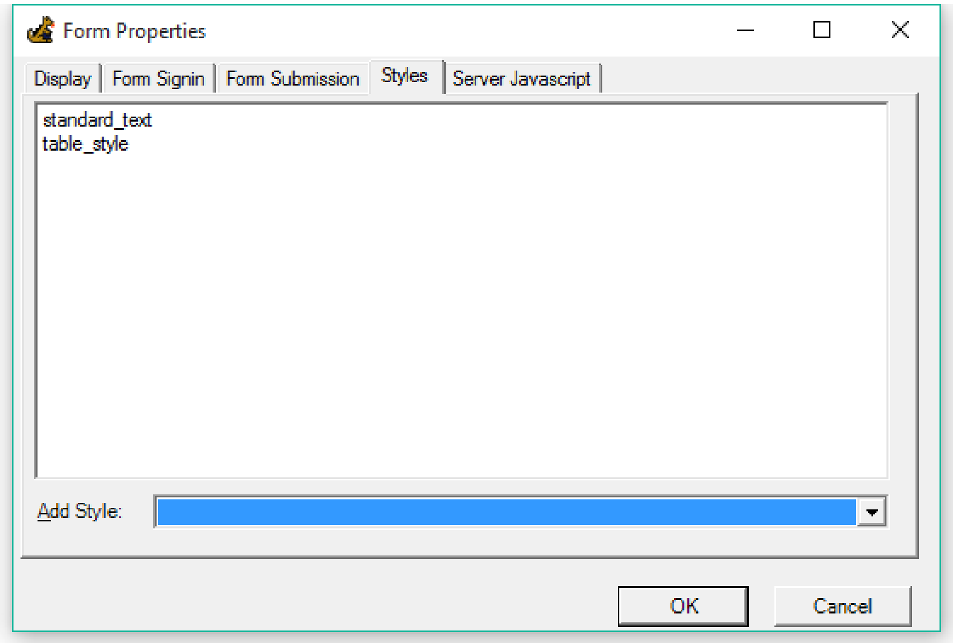
- **Enforce Unique Indexing:** Turn on and when a form is submitted with indexing values identical to an existing FDD document, the user will receive a "**Duplicate Values are not Allowed**" message. If this option is not selected, a submitted form with indexing values identical to an existing FDD document will be appended to the document as a new page.
- **Drop Form on Submit:** Turn on and the form page will be dropped on submission and will not be stored as part of the new FDD document. This option may be useful if you want the forms submission process to create an FDD document that contains only the uploaded files on the form (see File Upload for more information).

  Note: This option cannot be used if your form is configured to store field data in the form Document. In this case, you will receive a warning (when saving your form design) that the form will be saved without this option. See Data Storage and File Upload for more information.
- **Include Confirmation Page:** Turn on and a confirmation page will display after the form is submitted.
- **Pop Success Message:** Turn on and a pop-up success message displays once the form is submitted.
- **Pop Fail Message(s):** Turn on and one or more pop-up failure messages display if the form fails to submit with errors (e.g. mandatory field was not filled in).
- **Show Print View Button:** The print view button displays a printer-friendly view of the form.
  - **On Success Page:** Turn on and the print view button will appear on the form submission success page.
  - **On View Page:** Turn on and the print view button will appear on the form page when viewed in FDD or WebFDD.
- **Success Message:** The message displayed to the user after the submitted form is successfully saved as an FDD document. This setting defaults to **Your form was submitted successfully**.

  Tip: If you want to have a multi-line message, use **<br>** tags.

Styles

Add CSS created in Feith Developer and apply it to your form. In the **Add Style** field, select the desired CSS and it is added to you form. Select as many CSS objects as you want and you can reorder them using click-and-drag.

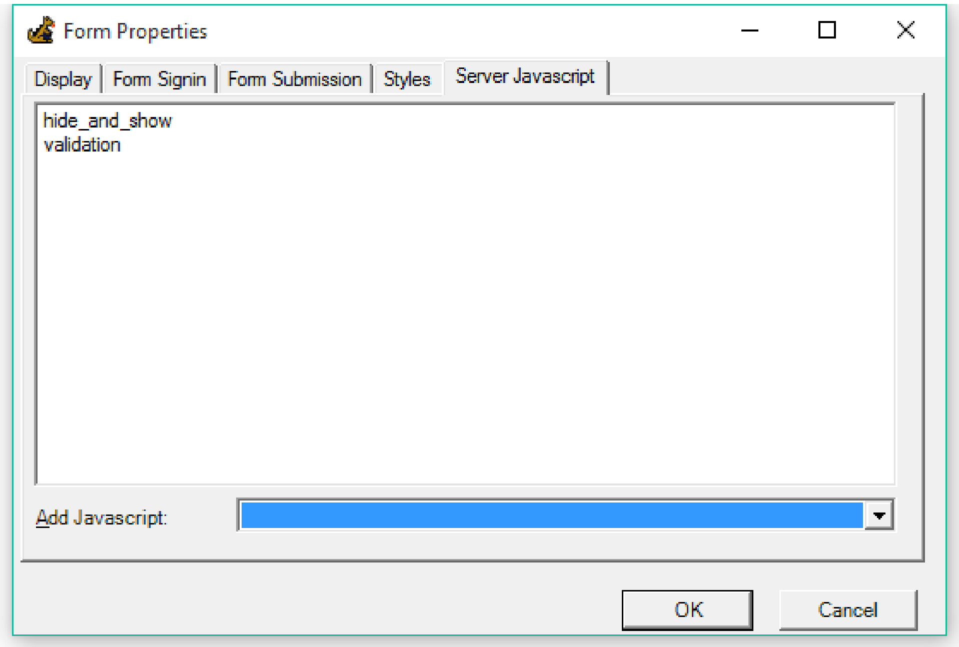


standard_text
table_style

Add Style:

Server Javascript

Add Javascript created in Feith Developer and apply it to your form. In the **Add Javascript** field, select the desired Javascript and it is added to you form. Select as many Javascript objects as you want and you can reorder them using click-and-drag.



hide_and_show
validation

Add Javascript:

Global Defaults

Set defaults for various form properties that control how the form looks and behaves. See Set Form Properties for more information. For example, for every form you create you can set the default background color and default success message.

To set the global defaults:

1. Make sure you have no form open (the option is disabled when a form is open in the Designer).

2. Select **Tools>Global Defaults**. The **Form Property Defaults** dialog opens.

3. Change the properties on any of the tabs. See the following for more information:

   - Set the defaults for the settings found in Form Properties

   - Global-Only Options

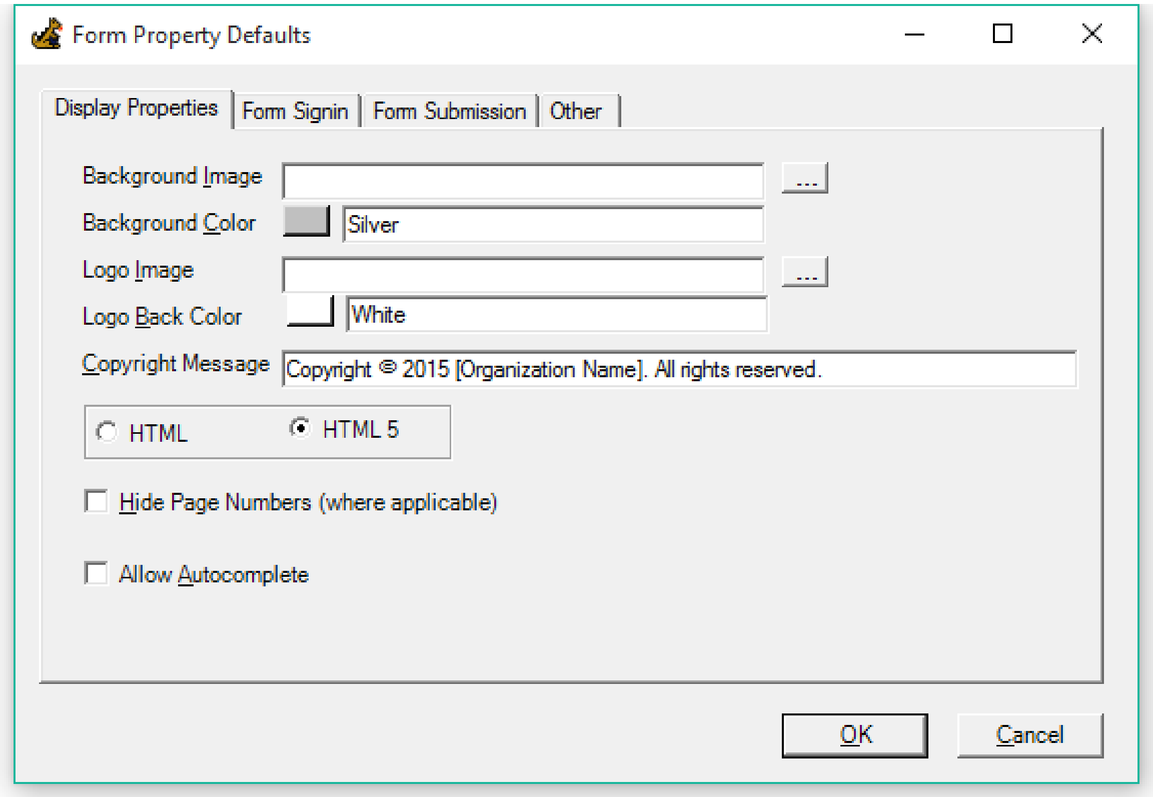
   - Other Options

4. Click **OK** to save changes. The next time you create a form, the form will be created with these defaults.

Global-Only Options

The following display options apply to all forms and cannot be changed in the properties for an individual form. These options only appear under the global defaults:

- **Logo Image**: The logo to display on your forms. Click the browse button to select a .gif or .jpg file to use as the logo.
- **Logo Back Color**: The background color of the logo. This setting defaults to white. Click to open the Color palette to choose a logo background color, or type in an HTML color or hex code.
- **Copyright Message**: The default copyright message that displays on your forms. The copyright message can be changed per form by editing its text on the form in the Designer.
- **HTML and HTML 5:** Sets whether **HTML** or **HTML 5** is the default when creating new forms. See Create a New Form for more information.



Background Image:
Background Color: Silver
Logo Image:
Logo Back Color: White
Copyright Message: Copyright © 2015 [Organization Name]. All rights reserved.

○ HTML ● HTML 5

□ Hide Page Numbers (where applicable)

□ Allow Autocomplete

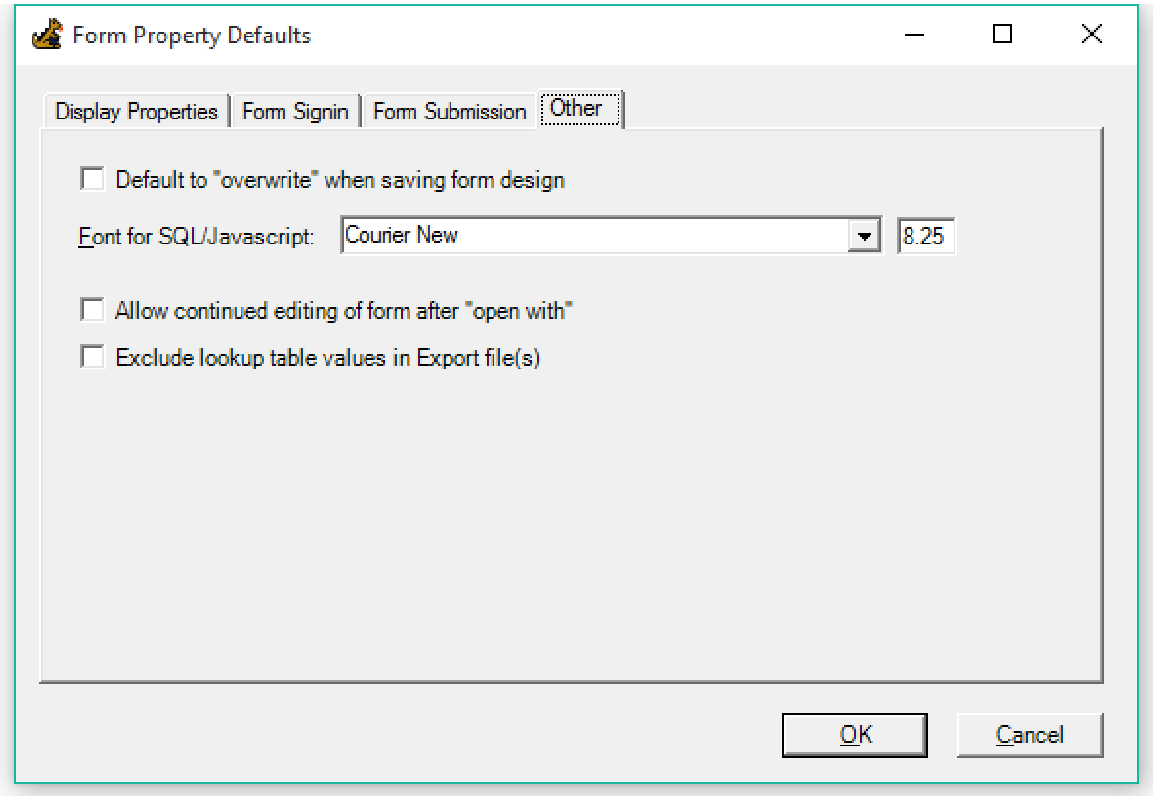
Note: The following options can only be set in the properties for an individual form; they do not appear under the global defaults:

- **Form Title**
- **Load SQL**

Other Options

The **Other** tab on the **Form Property Defaults** dialog includes the following options:

- **Default to "overwrite" when saving form design:** If this option is checked, the **Overwrite** option will be selected by default instead of **New Version** when saving changes to an existing form. See Modify Form for more information on editing forms.
- **Font for SQL/Javascript**: The font of SQL and JavaScript text displayed in Forms iQ Designer. For example, you may prefer a fixed-width font for the SQL you write in an autofill.
- **Allow continued editing of form after "open with":** If this option is checked, the **Pull HTML** button becomes available in the toolbox on the left. After opening a form in another editor and making changes, click the **Pull HTML** button to bring the changes from the other editor into Forms iQ Designer. This option enables you to switch between Forms iQ Designer and the other editor as desired, without closing the other editor. See Open Page with Other Editor for more information.
- **Exclude lookup table values in Export file(s):** If this option is checked, when you export a form the lookup table values are excluded from the export. Therefore, if you generate a lookup table when importing the form, the lookup table itself will be created but it will contain no values. See Export for more information.



□ Default to "overwrite" when saving form design

Font for SQL/Javascript: Courier New 8.25

□ Allow continued editing of form after "open with"

□ Exclude lookup table values in Export file(s)

Format a Form

Format your form so it looks exactly how you want. For instructions on adding, editing and deleting tables, see Add Tables.

To change the background color:

1. Select **Form>Properties**. The **Form Properties** dialog opens.
2. Click the **Background Color** icon ☐ and pick a color, or you can type in a color name or hex code in the **Background Color** field.
3. Click **OK**. The new background color is applied.

To set or change the background image:

1. Select **Form>Properties**. The **Form Properties** dialog opens.
2. Click the browse button next to the **Background Image** field, and select the .jpg or .gif file to use as the background image.
3. Click **OK**. The new background image is applied.

To change the font:

1. Select the text to format.
2. Choose a font from the drop-down list Arial ▾ on the toolbar. The font is applied.

Note: The font list retains the last selection. To apply the selected font to text on the form, highlight the text and click the font list.

To change the font size:

1. Select the text to format.
2. Choose a new font size from the drop-down list 2 (10 pt) ▾ on the toolbar. The new font size is applied.

Note: The font size list retains the last selection. To apply the selected font size to text on the form, highlight the text and click the font size list.

To change the font color:

1. Select the text to format.
2. Click the color palette icon A on the toolbar. The **Color** palette opens.
3. Choose a color and click **OK**. The new font color is applied.

To bold text:

1. Select the text to format.
2. Click the **Bold** button B on the toolbar. The text is bolded.

To italicize text:

1. Select the text to format.
2. Click the **Italicize** button I on the toolbar. The text is italicized.

To underline text:

1. Select the text to format.
2. Click the **Underline** button on the toolbar. The text is underlined.

To change the text alignment:

1. Select the text to format.
2. Click either the left , center or right alignment button on the toolbar. The text is aligned.

To undo the last formatting change:

- Select **Edit>Undo**. The last change is undone.

To add a horizontal bar:

1. Click on the form where you want to add the horizontal bar.
2. Click the **Add Horizontal Bar** button in the toolbar. A horizontal bar is added to the form.

To move a field:

- Click-and-drag the field to the desired location.

Alternatively:

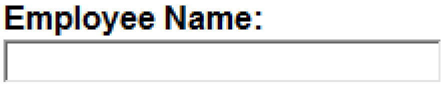
1. Select the desired field.
2. Select **Edit>Cut**. The field is cut and put on the clipboard.
3. Click in the location where you want the field to be placed.
4. Select **Edit>Paste**. The field is pasted in the cursor's location.

To copy a field:

1. Select the desired field.
2. Select **Edit>Copy**. The field is copied to the clipboard.
3. Click in the location where you want the copy to be placed.
4. Select **Edit>Paste**. A copy of the field is pasted in the cursor's location. The copy has the same name as the original field with a "2" on the end.

   Note: If you paste the copy before the original field, the second field *in order* is considered the copy. E.g. if you paste a copy of "invoice_number" before the original "invoice_number", the first, pasted field (the result of the copy) is called "invoice_number" and the second, original field is changed to "invoice_number_2".

Tip: To insert a line break between a label and a form field, type the label, then press **SHIFT+ENTER**, then add the field. This will reduce the amount of space added between the two lines. For example:



Employee Name:

Add Tables

Use tables to add structure to the layout of your form.

To add a table:

1. Click on the form where you want to add the table.
2. Click the **Add a Table** button in the toolbar. A new table is added to the form.

To add a row to a table:

1. Click on the form within the table.
2. Click the **Add a Row** button in the toolbar. A row is added to the bottom of the table.

To add a column to a table:

1. Click on the form within the table.
2. Click the **Add a Column** button in the toolbar. A column is added to the right of the table.

To delete a row:

1. Click on the form in the row you want to delete.
2. Click the **Delete Current Row** button in the toolbar. The row is deleted.

To delete a column:

1. Click on the form in the column you want to delete.
2. Click the **Delete Current Column** button in the toolbar. The column is deleted.

To move a row up:

1. Click on the form in the row you want to move.
2. Click the **Move Row Up** button in the toolbar. The row is moved up within the table.

To move a row down:

1. Click on the form in the row you want to move.
2. Click the **Move Row Down** button in the toolbar. The row is moved down within the table.

To change the table properties (e.g. width, border, color):

1. Right-click on the table and select **Properties**. The **Table Properties** dialog opens.

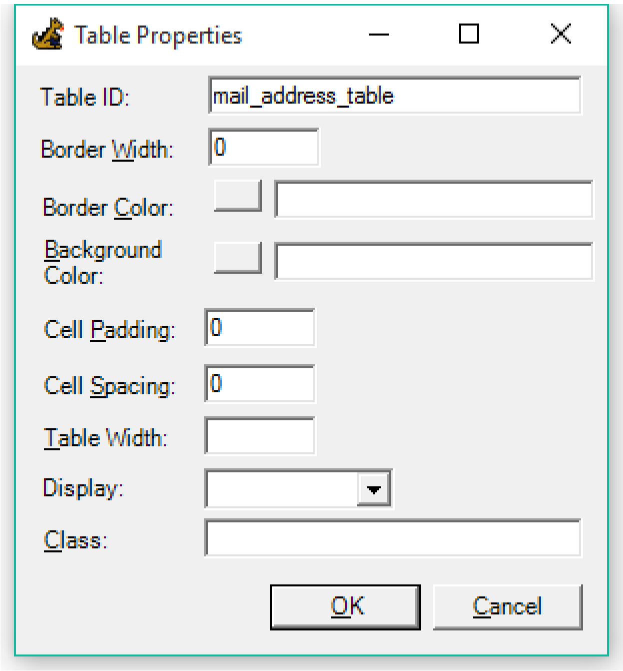


Table Properties

Table ID: mail_address_table

Border Width: 0

Border Color:

Background Color:

Cell Padding: 0

Cell Spacing: 0

Table Width:

Display:

Class:

2. Change any of the following table properties:

    - **Table ID:** The ID of the table. This needs to be set when you want to hide and show tables using Custom Javascript. See Hide and Show Examples more information.
    - **Border Width:** The width, in pixels, of the table border. Set to **0** for no border.
    - **Border Color:** The color of the table border. To choose a color, click the button to open the **Color** palette or type in a color name or hex code.

      Note: For the table border to display, the **Border Width** must be set to a value greater than **0** and the **Border Color** must be set.
    - **Background Color:** The background color of the table. To choose a color, click the button to open the **Color** palette or type in a color name or hex code.
    - **Cell Padding:** The amount, in pixels, of white space between the cell content and its borders.
    - **Cell Spacing:** The distance, in pixels, between the cells.
    - **Table Width:** The width of the table. Either a percentage (e.g. 100%) or a set width in pixels (e.g. 800px), may be entered.
    - **Display:** The display property of the table. If set to **none**, the table will be hidden by default. See Hide and Show Examples more information.
    - **Class:** The HTML class of the table.

3. Click **OK**. The changes are applied.

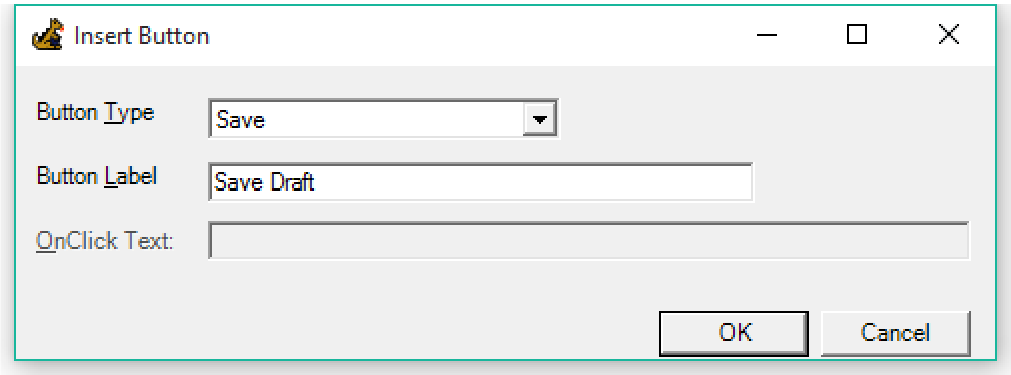
Add Buttons

A **Submit** button is added to all forms by default. In addition, you can add the following types of buttons to a form page.

BUTTON TYPE	ACTION
Submit	Submit the form.
Reset	Reset the fields to what they were when the form first loaded.
Next Page	Go to the next page in a multi-page form. See Multi-page Forms for more information.
Prev Page	Go to the previous page in a multi-page form. See Multi-page Forms for more information.
Save	Save a draft of the form. With this option, the user can login or sign-in again later to open the saved draft and complete the form. A **Save** button can only be added to forms with login or a sign-in page. See Login Permissions or Sign-In Page for more information.
New Form	Start a new blank form. The **New Form** button appears on the success page after the form has been submitted.
Retrieve Submitted Forms	Retrieve forms submitted as the signed-in user. With this option, the user can enter their sign-in credentials then click the button to view a list of forms they previously submitted. This button can only be placed on a sign-in page. See Sign-In Page for more information.
Custom OnClick	Create your own button that has your custom OnClick command that will run when a user clicks it.
View	View the configured page in the alternate print view. With this option, the user can navigate to another page in the alternate print view. See Define Alternate Print View for more information. This button can only be placed on an alternate print view page.

To add a button:

1. Click on your form where you want to add the button.
2. Click the **Add Button** tool in the toolbar. The **Insert Button** dialog opens.
3. Choose the **Button Type**.
4. Enter the **Button Label**. The label defaults to the name of the button type.
5. If you are creating a **Custom OnClick** button type, enter the **OnClick Text**.



Button Type: Save

Button Label: Save Draft

OnClick Text:

6. Click **OK**. The button is added.

To modify a button's label:

1. Double-click the button. The **Edit Button** dialog opens.
2. Change the **Button Label** as desired.
3. Click **OK**. The button's label is modified.

Tip: The same button type can be added multiple times to the same form page; for example, you can add a **Next Page** button at the top and bottom of the page.

Open Page With Other Editor

Open the form in another editor program, such as a text editor, to format the form as desired. Alternatively, you can edit the HTML in the **Body HTML** tab right in Forms iQ Designer. See Form Tabs more information.

Note: This feature is intended to be used for formatting changes only.

To open a form page with another editor:

1. Select **Page>Open With** and choose one of the listed editors. See Maintain Open With Apps for more information on adding your preferred editor.
2. You are prompted with a warning that any field additions, field deletions, or field property changes may cause the form to break.
3. Click **OK** to proceed and open the page in the selected editor.
4. Make the desired changes.
5. Save your changes in the editor then close the editor. You are returned to Forms iQ Designer and the changes you made in the other editor are applied to the form.

   Alternatively, if you want to leave the editor open and go back and forth between the editor and Forms iQ Designer rapidly, turn on **Allow continued editing of form after "open with"** in global defaults (see Global Defaults for more information).
   With this option on, instead: Save your changes in the editor, leaving it open, and click **Pull HTML** in Forms iQ Designer. The changes you made in the other editor are applied to the form.

When editing a form page using another editor, please note the following:

- Always use the **Open With** option to edit a form page in another application. Do not open an exported form in another application; the form export is intended to be imported back into the Forms iQ Designer only.
- A form with the HTML doctype **<!DOCTYPE html>** will have some elements and attributes replaced with valid HTML5 elements and attributes on the live form in the browser. The HTML5 elements and attributes are not displayed in the HTML you view in the other editor. See Set Form Properties and Create a New Form for more information.
- Editing or deleting the form field ID or name can invalidate the HTML. If the HTML is invalidated, Forms iQ Designer may be unable to save the form design.
- Adding or deleting fields may cause the form to break.
- Any Microsoft FrontPage embedded images added to the form page will not display when the page is saved back into Forms iQ Designer.
- Formatting HTML with TAB indents and carriage returns will not be preserved once the form is saved.

Maintain Open With Apps

Define which applications are available when you select **Page>Open With**.

To add an application:

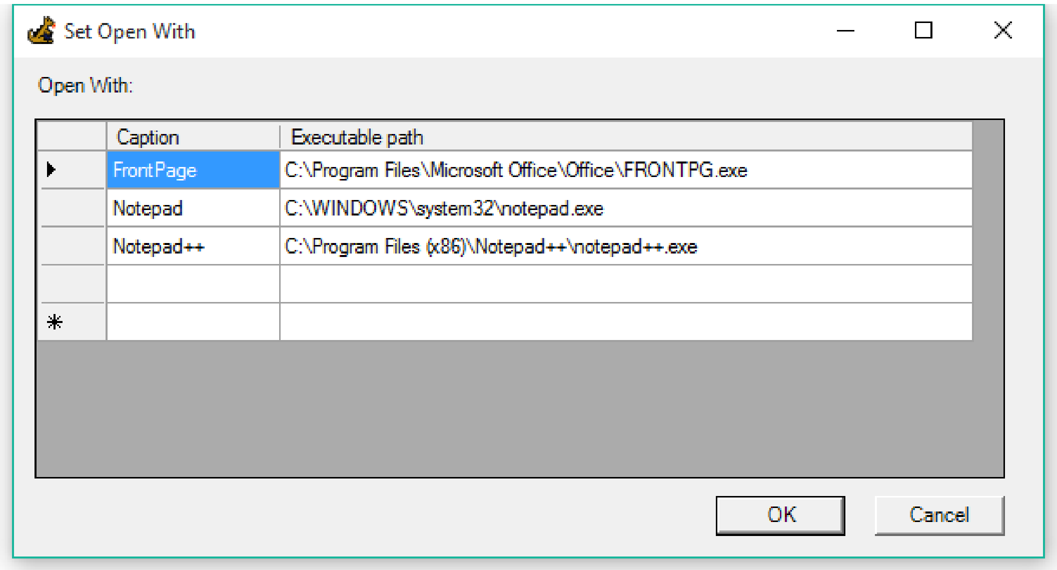
1. Select **Tools>Maintain Open With Apps**. The **Set Open With** dialog opens.

2. At the bottom of the list in an empty row, double-click in the **Caption** column.

3. Type in a friendly name for the application that will display in the **Page>Open With** sub-menu.

4. In the same row, double-click in the **Executable path** column.

5. Type or paste in the full path to the application's executable file, including the name of the executable file.



	Caption	Executable path
▶	FrontPage	C:\Program Files\Microsoft Office\Office\FRONTPG.exe
	Notepad	C:\WINDOWS\system32\notepad.exe
	Notepad++	C:\Program Files (x86)\Notepad++\notepad++.exe
✱		

6. Click **OK**. The application is added.

To delete an application:

1. Select **Tools>Maintain Open With Apps**. The **Set Open With** dialog opens.

2. Select the row of the application you want to delete by clicking on the gray area to the left of the row.

3. Hit the **DELETE** key. The application is deleted from the list.

4. Click **OK**. Your changes are saved.

Sign-In Page

A sign-in page prompts users to enter sign-in data before accessing the main form. Optionally add a sign-in page to your form in order to:

- Restrict access to the form. Users will only be access the form if they enter valid sign-in credentials. You decide what valid sign-in credentials are by setting up verification of the data.

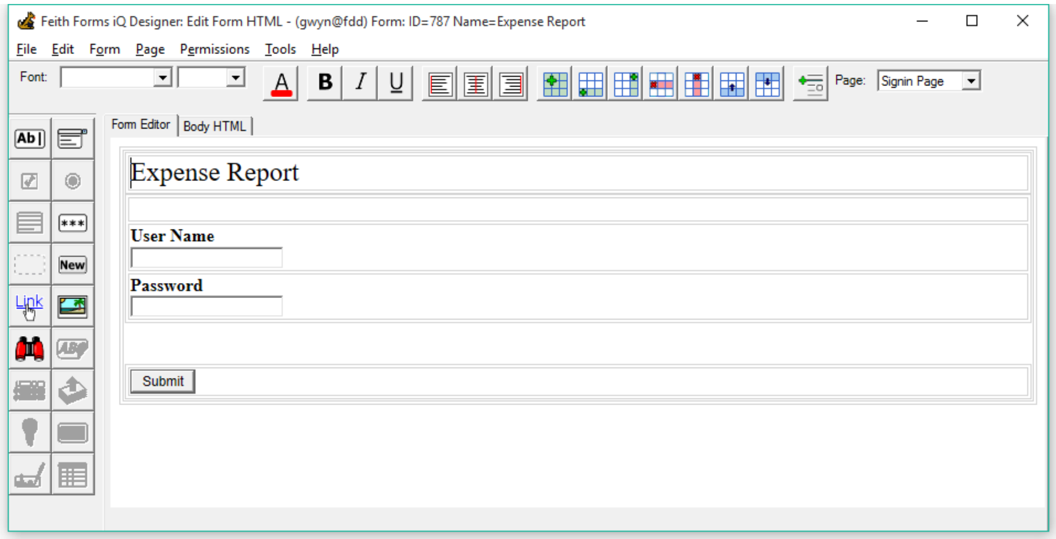
  Tip: If you want to restrict the form to FDD users only, try setting login permissions on the form instead of using a sign-in page. See Login Permissions for more information.

- Autofill data on the form. Fields on the main form will be autofilled based on the entered sign-in value(s). See Autofill from Sign-In for instructions on configuring autofill of a field based on a sign-in field value.

To add a sign-in page:

- Select **Page>Add Sign-In Page**. A sign-in page is added to the form and displayed in the Designer.

  The default sign-in page includes a textbox labeled **User Name** and a password field labeled **Password**. You can customize the sign-in page as needed, modifying these default fields or adding fields of your own.



Feith Forms iQ Designer: Edit Form HTML - (gwyn@fdd) Form: ID=787 Name=Expense Report

Page: Signin Page

Expense Report

User Name

Password

To add sign-in fields and set verification:

1. Click on the form where you want to add the field.

2. Click the button for the desired field type in the toolbar. You can add a textbox, password, select, lookup, hyperlink, and hidden fields to a sign-in page. The field is added to the form.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, and more. See Common Field Properties for more information.

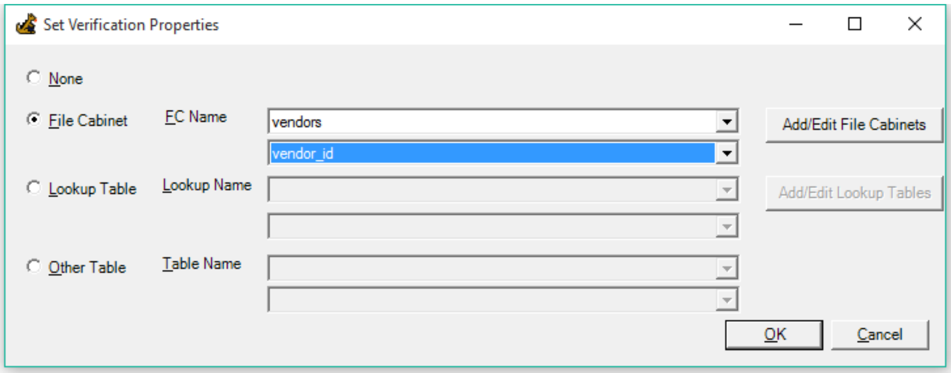
5. Set the properties specific to this field type. See the following for details:

- Textbox
- Password
- Select
- Lookup
- Standard or Custom Hyperlink
- Hidden

6. Optionally click **Set Verification Properties** to check the data the user enters in the sign-in page. If they enter incorrect data, they are not allowed to access the main form. Options are:

    - **None:** Sign-in data is verified against nothing. The user can enter anything and gain access to the main form.
    - **File Cabinet:** Sign-in data is verified against values in a file cabinet. Select the **FC Name** and the file cabinet field containing the values you want to check against. The user must enter a value that exists in the file cabinet field in order to access the main form.
    - **Lookup Table:** Sign-in data is verified against values in a lookup table. Select the **Lookup Name** and the lookup column containing the values you want to check against. The user must enter a value that exists in the lookup table column in order to access the main form.
    - **Other Table:** Sign-in data is verified against values in another table. Select the **Table Name** and the table column containing the values you want to check against. The user must enter a value that exists in the table column in order to access the main form.

      Note: The table must be in your Other Tables List in order to select it here. See Maintain Other Tables List for more information.



Set Verification Properties

○ None

● File Cabinet — FC Name: vendors; vendor_id

○ Lookup Table — Lookup Name

○ Other Table — Table Name

7. Click **OK**. The verification properties are saved.

8. Click **OK**. The field properties are saved.

To delete a sign-in page:

1. Select the **Signin Page** using the page selector in the upper left. The sign-in page displays.

2. Select **Page>Delete Current Page**. The sign-in page is deleted.

Do more with your sign-in page:

- Set **Form Sign-In Properties**, such as the message that displays when a user enters incorrect sign-in data. See Set Form Properties for more information.

- Add a **Save** button to a form with a sign-in page, which lets users partially fill out the main form and save it with their sign-in credentials so they can finish filling it out later. See Add Buttons for more information.

- Add a **Retrieve Submitted Forms** button to the sign-in page, which lets users retrieve forms they submitted previously. See Add Buttons for more information.

Configure Save Draft

Allow users to partially fill out their forms and save a draft to come back and complete later. When the user signs in or logs in later, they can access their saved drafts and finish filling out the forms.

To configure a form to allow saving drafts:

1. Set login permissions or add a sign-in page to the form. See Login Permissions or Sign-in Page for more information.
2. Add a **Save** button to the main page of the form using the **Add Button** tool in the toolbar. See Add Buttons for more information.
3. Continue designing the form as needed and save when done.

When a user saves a draft with sign-in, the next time they sign in they will be prompted to finish the form or they can choose to start a new form. When a user saves a draft with login, they can access their drafts in Forms iQ Server's My Forms interface.

Define Alternate Print View

Customize the printer friendly view your users see when they click the **Printer Friendly View** button after filling out a form.

The default print view shows all form fields and, if it's a multi-page form, all pages separated by horizontal bars. Creating an alternate print view replaces the default one, and you can do things like customize the layout of the fields or remove some fields. You can also remove your custom print view to return to the default print view.

To define an alternate print view:

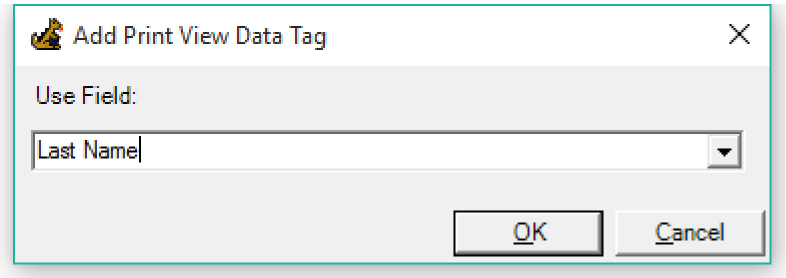
1. Select **Page>Define Alternate Views**. The alternate print view displays.

2. Add fields to the print view:

    a. Click in the location where you want the field to go.

    b. Click the **Add Print Data** button in the toolbar. The **Add Print View Data Tag** dialog opens.

    c. Select the desired field.



Use Field: Last Name

    d. Click **OK**. A label field, which will display the data from the field you selected, is placed in the cursor's location.

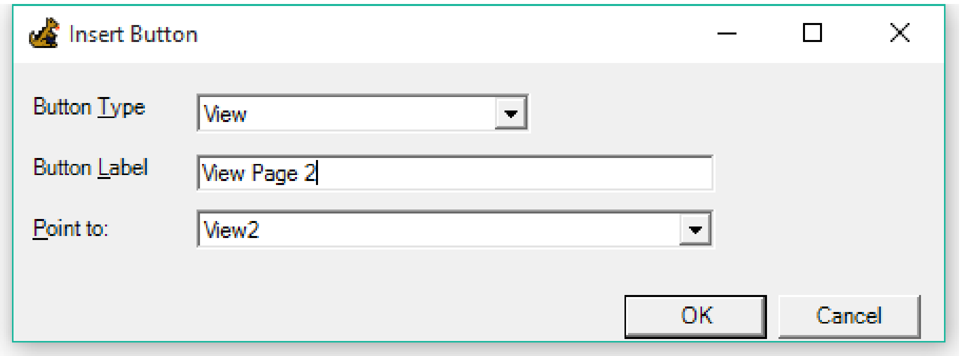
3. Optionally add more pages to the print view:

    a. Select **Page>Add View Page**. The page is added and displayed.

    Navigate between the print view's pages using the view page number drop-down list View1 in the toolbar in the upper right.

    b. Add fields to the page as desired.

    c. Add **View** buttons for users to navigate between the print view's pages. See Add Buttons for more general information on adding buttons.



Button Type: View

Button Label: View Page 2

Point to: View2

4. Once the alternate print view is defined, return to the main form by selecting **Page>Back to Main Screen**.
5. Save your form with its new alternate print view. Users who click **Printer Friendly View** on a submitted form will see the new alternate print view.

To delete an alternate print view and return to using the default print view:

1. Select **Page>Define Alternate Views**. The alternate print view displays.
2. Select a view page and select **Page>Delete Current Page**. The alternate print view is deleted and users who click **Printer Friendly View** after submitting the form will see the default print view.

   If there are multiple pages, repeat this process for each view page in the alternate print view until all are deleted.

Multi-page Forms

If you are designing a large form, you may want to design the form with multiple pages. To do this, add new pages to your form, then customize each page of the form by adding fields and formatting the page as needed.

When the user views the form in a browser window, they will see the first page of the form. Numbered hyperlinks to the other form pages will display at the bottom of the form. In addition, **Next Page** and **Previous Page** buttons can be added to each page of form.

To add a page:

- Select **Add Main Page** from the **Page** menu. A new page is added to the form.

To change the page selection:

- Choose the page number from the page number drop-down list in the toolbar. The selected page is displayed.

To delete a page:

1. Choose the page number from the page number drop-down list in the toolbar. The selected page is displayed.
2. Select **Delete Current Page** from the **Page** menu. The page is deleted.

If you delete a page, the pages are renumbered so that the page sequence does not skip numbers.

To add Next Page and Previous Page navigation buttons to a page:

1. Click on your form where you want to add the button.
2. Click **Add Button** in the toolbox. The **Add Button** dialog opens.
3. For the **Button Type**, choose **Next Page** or **Previous Page**, depending on what you want.
4. Enter the **Button Label**.
5. Click **OK**. The button is added.

See Add Buttons for more information.

Add Form Fields

Add Form Fields

A field is an object on the form that contains data. The data may be entered or viewed by a user, or storing a hidden, internal value. There are a variety of field types to choose from.

Notes:

- All field types can be added to a standard "main" form page. Only some field types can be added to a sign-in page. See below table and Sign-in Page for details.
- See Common Field Properties for the settings common to most field types that you can set.
- See Add Buttons for more information on buttons.

Tip: Generate form fields from columns in an "other" table to which you want to store using **Form>Add Fields From Table**. See Add Form Fields From Table for more information.

TOOLBAR ICON	FIELD TYPE	AVAILABLE FOR SIGN-IN PAGE?
Ab|	Textbox Enter a single line of text as input.	Yes
▤	Select Provide the user with a list of values to select from.	Yes
☑	Check Box Allow the user to make one or more selections from a set of options.	No
◉	Radio Button Limit the user to a single selection from a set of options.	No
▦	Text Area and Rich Text Editor Enter multiple lines of text as input.	No
***	Password Similar to a textbox except characters entered are displayed as asterisks (*) for privacy.	Yes
⬚	Hidden Store data that is not visible on the form.	Yes
Link	Hyperlink Add a hyperlink to the form. Available hyperlink types include: Standard Hyperlink, Custom Hyperlink, and Logout Hyperlink.	Yes, Standard and Custom hyperlinks only
🖼	Image Add an image to the form. The image may be a standard image (from a file on your computer) or a linked image.	Yes

	Lookup Provide the user with a list of values. Two columns of data can be displayed in a lookup field.	Yes
ABC	Label Display text that cannot be edited. The value of the label is set by an autofill.	No
	Dynamic List Allow the user to enter multiple lines of detail pertaining to a single item.	No
	File Upload Allow the user to browse and select a file to upload with the form.	No
	Smart Icon Display one of many images based on a rule. Rules can be based on form fields or SQL queries.	No
	Barcode Display a barcode on the form. The barcode value can be the document ID, the page ID, a selected field value, or a constant value.	No
	Signature Allow the user to sign the form. The signature is based on standard PKI (Public Key Infrastructure).	No
	Grid Display data in report format. Grid provides several helpful features such as the ability to reorder columns, print, and export to Microsoft Excel.	No

Common Field Properties

The following field properties are common to most field types. If the property is unavailable for your field type, then it cannot be set for that field type.

This properties are usually set one field at a time, but you can also use the Field Grid to set some field properties for many fields at once in a single interface.

FIELD PROPERTY	DESCRIPTION
Name	The name of the field. The field name is shown in the Designer's interface. For example, the field name is shown when selecting field tokens for use in calculations. A new field is automatically assigned the default name of **Field 1**. If the default name is not changed, the number increments, so that the next new field is named **Field 2** and so on.
Title	The title of the field. The field title is shown to users in error messages when submitting the form.
Data Type	The data type of the field. The data type options are **CHAR**, **DATE**, **DATETIME**, **INTEGER**, **SMALLINT**, **DECIMAL**, **MONEY**, **SIGNATURE**, **STRING-LIST**, **DATE-LIST**, and **NUMBER-LIST**. **Note:** For some field types, such as file upload fields and smart icons, the **Data Type** setting does not apply and is automatically set to **N/A**.
Field Type	The field type. When adding a field, this value is automatically set based on the button you clicked in the toolbar. However, the field type can be changed if needed. E.g. you can change a field added as a textbox to a lookup field later on. **Notes:** • A field of the data type **SIGNATURE** must be the field type **Signature**. • A field of the data type **STRING-LIST**, **DATE-LIST**, or **NUMBER-LIST** must be the field type **Text Area** or **Lookup**.
HTML Name	The name of the field in the HTML.
Visible Width	The width of the field. This number of characters displays in the field without scrolling.
Maximum Characters	The maximum number of characters accepted for input. A number will be suggested based on where the field's data will be stored. **Note:** For **DECIMAL** and **MONEY** fields, the maximum number of characters counts *all* characters in the field, including numbers before and after the decimal point as well as the decimal point itself.
Minimum Characters	The minimum number of characters accepted for input.
Scale	The number of digits to the right of the decimal point accepted for input. This setting applies to **DECIMAL** and **MONEY** fields only.
Mouseover Text	Text to display on mouseover of the field when the form is viewed in a browser.

Class	The HTML class of the field. To assign multiple classes, separate each one with a space.
Mandatory	Turn on and the field will require input. Users will not be able to submit the form without entering a value in the field.
Read Only	Turn on and the field value will be read-only. Users will not be able to edit the field value.
Storage	The data storage location for the field. Storage options are a file cabinet field, an other table column, or the document. Click **View/Edit Form Storage** to set where the field's data is stored. See Data Storage for more information. **Notes:** • A field of the data type **SIGNATURE** must be stored to the file cabinet with a file cabinet field type of **Signature**. • A field of the data type **STRING-LIST**, **DATE-LIST**, or **NUMBER-LIST** must be stored to the document or to the file cabinet with a file cabinet field type of **List of Strings**, **List of Dates**, or **List of Numbers**.
Autofill	The optional autofill configuration for the field. With the exception of **Autofill from Sign-in**, autofill is set by clicking **View/Edit Form Autofill**. See Autofill for more information.
Autofill from Sign-in	If selected, the form field will be autofilled with the value of the selected sign-in page field. **Note:** This setting is only available if your form includes a sign-in page. See Sign-In Page for more information.
Calculation Properties	Populate a field with a calculated value. See Calculation Properties for more information.

Field Grid

The field grid allows you to view and manage a few basic properties of all the fields on your form in one interface.

Note: The field grid does not include signature fields, dynamic lists and their fields, or sign-in fields.

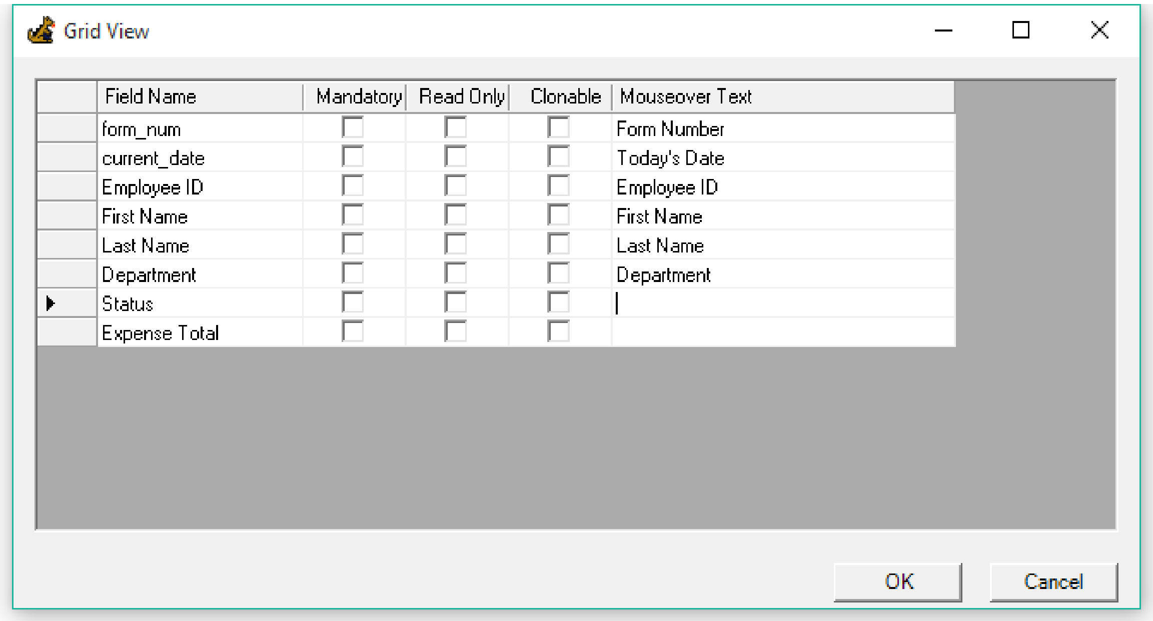
Select **Form>Field Grid** to view the field grid.

To edit the Field Name or Mouseover Text:

1. Click in the cell containing the field name or mouseover text you want to change.
2. Edit the field name or mouseover text as desired.
3. Click **OK**.

To edit Mandatory, Read Only, or Clonable settings:

1. Click in the cell containing checkbox you want to toggle
2. Check or uncheck the checkbox as desired.
3. Click **OK**.



	Field Name	Mandatory	Read Only	Clonable	Mouseover Text
	form_num	☐	☐	☐	Form Number
	current_date	☐	☐	☐	Today's Date
	Employee ID	☐	☐	☐	Employee ID
	First Name	☐	☐	☐	First Name
	Last Name	☐	☐	☐	Last Name
	Department	☐	☐	☐	Department
▶	Status	☐	☐	☐	
	Expense Total	☐	☐	☐	

Textbox

Enter a single line of text as input.

To add a textbox:

1. Click on the form where you want to add the field.

2. Click **Add Textbox Field** in the toolbar. A textbox is added to the form.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.

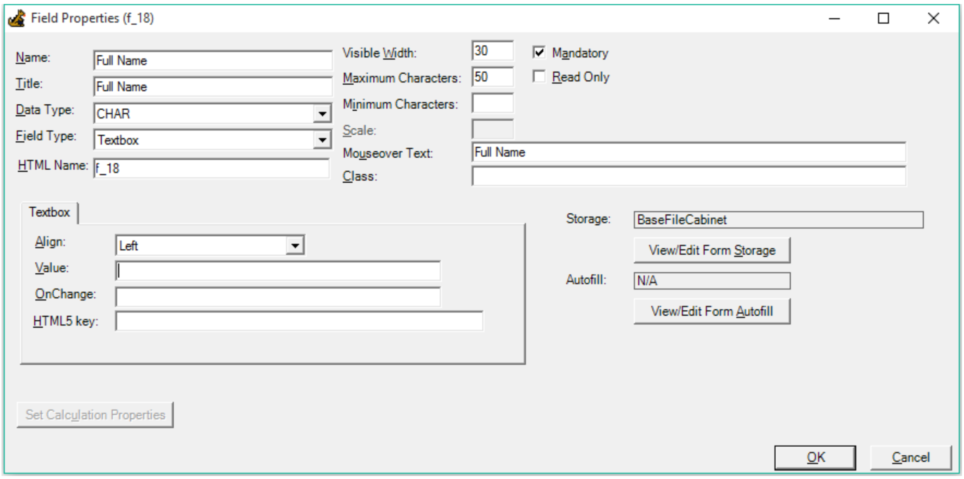
5. Set the properties specific to this field type:

   - **Align**: The alignment of text within the field. Options are **Left**, **Right** and **Center**.

   - **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

   - **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.

   - **HTML5 key:** The HTML input type of the field (e.g. email).



Name: Full Name
Title: Full Name
Data Type: CHAR
Field Type: Textbox
HTML Name: f_18
Visible Width: 30
Maximum Characters: 50
Minimum Characters:
Scale:
Mouseover Text: Full Name
Class:
☑ Mandatory
☐ Read Only

Align: Left
Value:
OnChange:
HTML5 key:

Storage: BaseFileCabinet
Autofill: N/A

6. Click **OK**. The field properties are saved.

Select

Provide the user with a list of values to select from. It's also called a drop-down list.

To add a select field:

1. Click on the form where you want to add the field.

2. Click **Add List Select Field (Dropdown Box)** in the toolbar. A select field is added to the form.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill** and **Storage**. See Common Field Properties for more information.

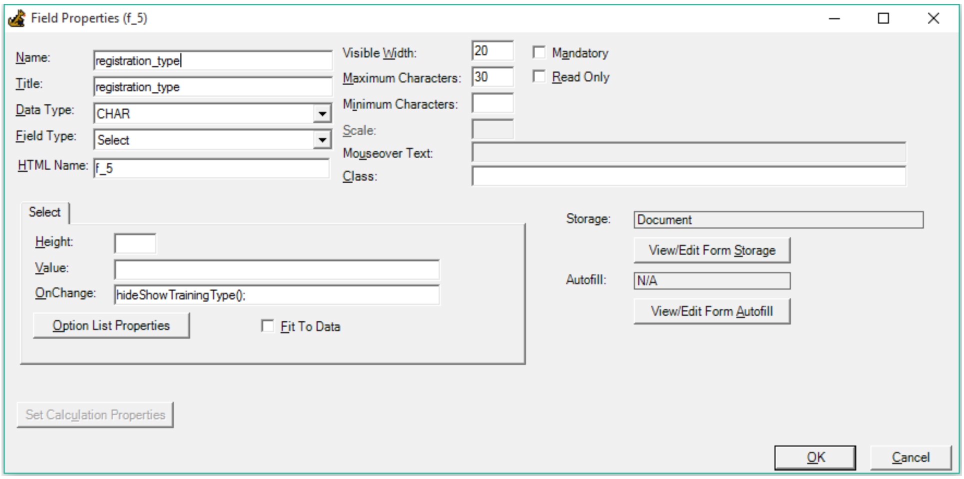
5. Set the properties specific to this field type:

   - **Height**: The height of the field.

   - **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

   - **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.

   - **Fit to Data**: Turn on and the width of the field will adjust to fit the longest value in the list.

   Field Properties (f_5)

   Name: registration_type

   Title: registration_type

   Data Type: CHAR

   Field Type: Select

   HTML Name: f_5

   Visible Width: 20

   Maximum Characters: 30

   Minimum Characters:

   Scale:

   Mouseover Text:

   Class:

   □ Mandatory

   □ Read Only

   Select

   Height:

   Value:

   OnChange: hideShowTrainingType();

   □ Fit To Data

   Storage: Document

   Autofill: N/A

6. Click **Option List Properties** to configure the select field's list using the following options:

   - **Include a Blank Value**: Turn on and a blank value will be included in the option list.

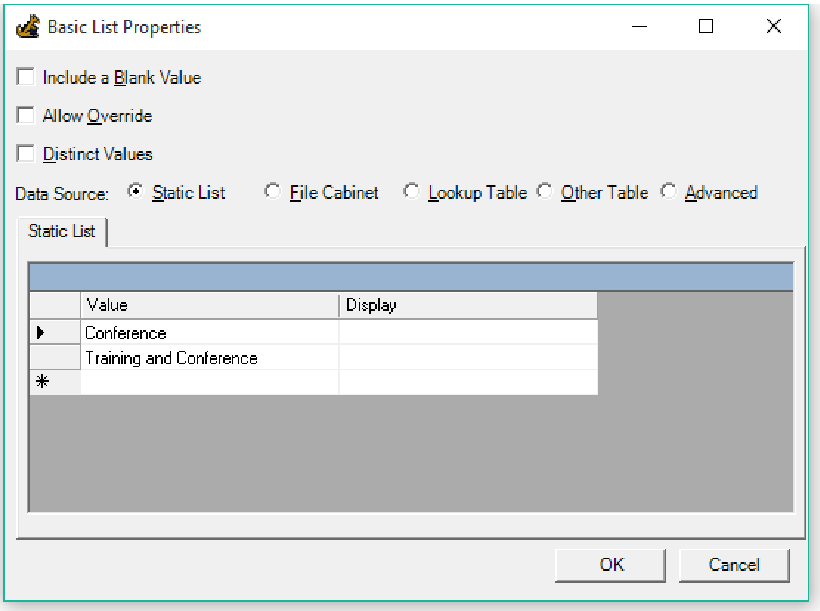
     Note: This option is disabled if the field is mandatory.

   - **Distinct Values**: Turn on and the option list will include only the distinct values from the selected value column in the data source.

     Note: This option is disabled if you choose the **Advanced** SQL **Data Source**.

- **Data Source**: Select the data source for the list. Choices include:
  - **Static List:** Enter values for the list.

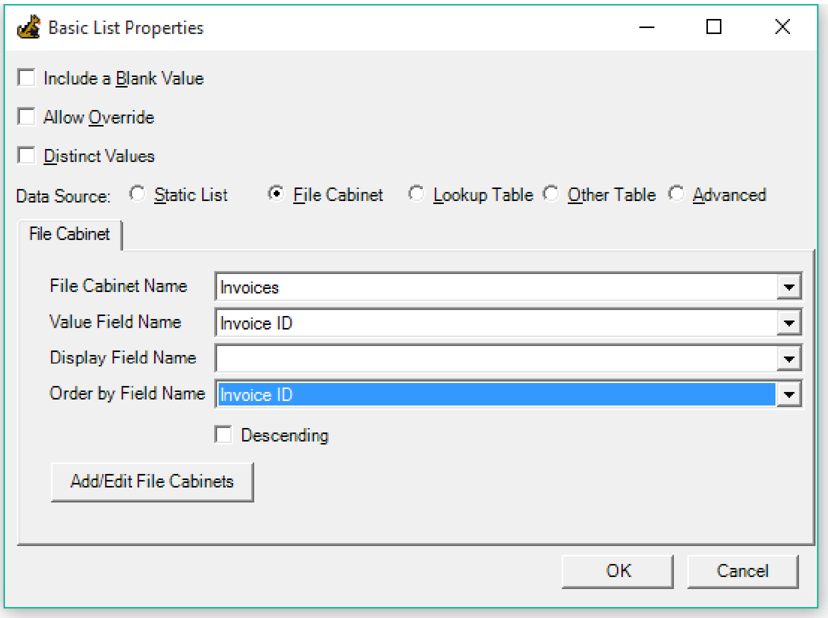
    Click in the row and enter the desired **Value**. Optionally enter a friendly **Display** value to show to the user in place of the **Value**.



Value	Display
Conference	
Training and Conference	

  - **File Cabinet:** Get values from a file cabinet.

    Select the **Field Cabinet Name** and **Value Field Name**. Optionally select a friendly **Display Field Name** to show to the user in place of the value. Optionally select an **Order by Field Name** to order the values; the order is ascending by default but you can choose **Descending** instead.

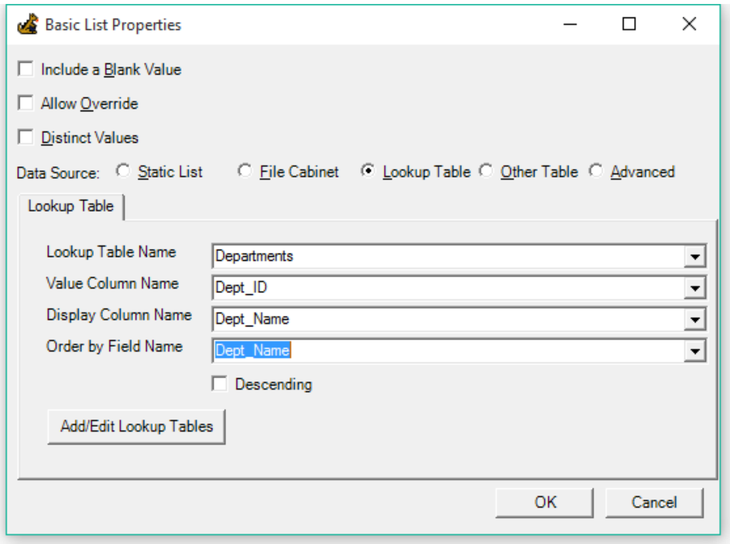


File Cabinet Name: Invoices
Value Field Name: Invoice ID
Display Field Name:
Order by Field Name: Invoice ID
☐ Descending

- **Lookup Table:** Get values from a lookup table.

  Select the **Lookup Table Name** and **Value Column Name**. Optionally select a friendly **Display Column Name** to show to the user in place of the value. Optionally select an **Order by Field Name** to order the values; the order is ascending by default but you can choose **Descending** instead.

  Alternatively, you can hand-type into these fields to use any table or view in your FDD database as the list. Make sure to type in the table/view name and column names correctly.



Basic List Properties

☐ Include a Blank Value

☐ Allow Override

☐ Distinct Values

Data Source: ○ Static List ○ File Cabinet ● Lookup Table ○ Other Table ○ Advanced

Lookup Table

Lookup Table Name: Departments

Value Column Name: Dept_ID

Display Column Name: Dept_Name

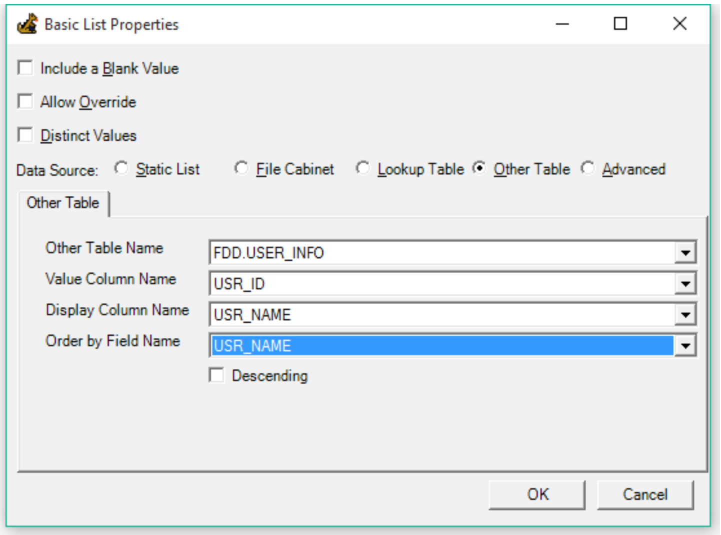
Order by Field Name: Dept_Name

☐ Descending

- **Other Table:** Get values from another table.

  Select the **Other Table Name** and **Value Column Name**. Optionally select a friendly **Display Column Name** to show to the user in place of the value. Optionally select an **Order by Field Name** to order the values; the order is ascending by default but you can choose **Descending** instead.

  Note: The table must be in your Other Tables List in order to select it here. See Maintain Other Tables List for more information.



Basic List Properties

☐ Include a Blank Value

☐ Allow Override

☐ Distinct Values

Data Source: ○ Static List ○ File Cabinet ○ Lookup Table ● Other Table ○ Advanced

Other Table

Other Table Name: FDD.USER_INFO

Value Column Name: USR_ID

Display Column Name: USR_NAME

Order by Field Name: USR_NAME

☐ Descending

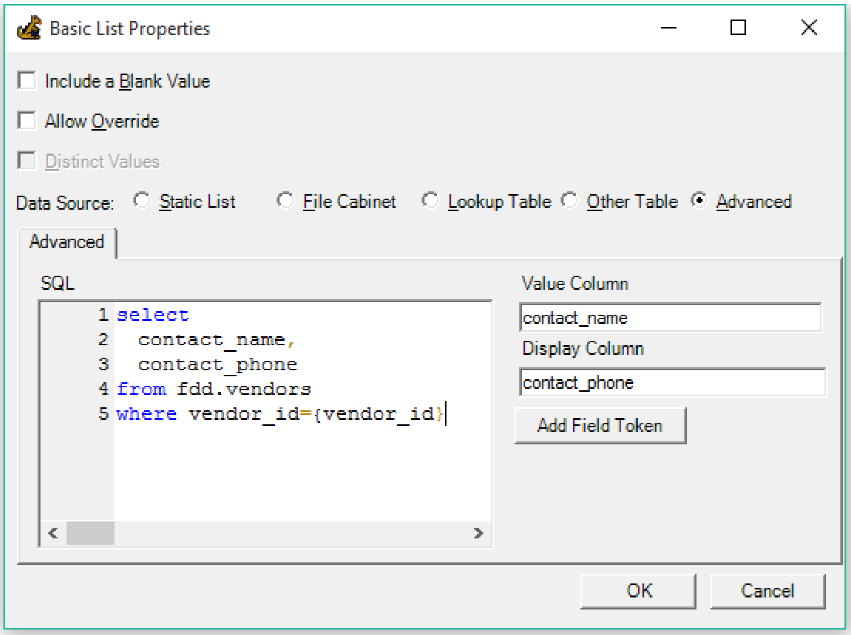
- **Advanced:** Get values from a SQL query.

  Enter the **SQL** query, in which you may **Add a Field Token**. Also enter the **Value Column**. Optionally enter a friendly **Display Column** to show to the user in place of the value.

  Tip: You can format your text using **TAB** to indent and **SHIFT+TAB** to un-indent.

  Notes:

  - This option requires SQL knowledge and the **Edit Forms iQ SQL** task permission set in Feith Control Panel.
  - Make sure any field token you use is or is not enclosed in single quotes, as is appropriate for the type of data.
  - Using a token of a list field (e.g. STRING-LIST) is not supported.



□ Include a Blank Value

□ Allow Override

□ Distinct Values

Data Source: ○ Static List ○ File Cabinet ○ Lookup Table ○ Other Table ● Advanced

SQL

```
select
  contact_name,
  contact_phone
from fdd.vendors
where vendor_id={vendor_id}
```

Value Column: contact_name

Display Column: contact_phone

Add Field Token

- **Note:** When you set a **Display** in addition to a **Value**, the Display is shown to the user in place of the Value when they are filling out the form, but it is the Value that is actually stored.

7. Click **OK**. The option list properties are saved.

8. Click **OK**. The field properties are saved.

Check Box

Allow the user to make one or more selections from a set of options.

To add a check box:

1. Click on the form where you want to add the field.

2. Click **Add Checkbox Field** in the toolbar. A check box is added to the form.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.

   Note: If autofill is configured for a check box, the check box will be selected if the autofill value matches the field value.

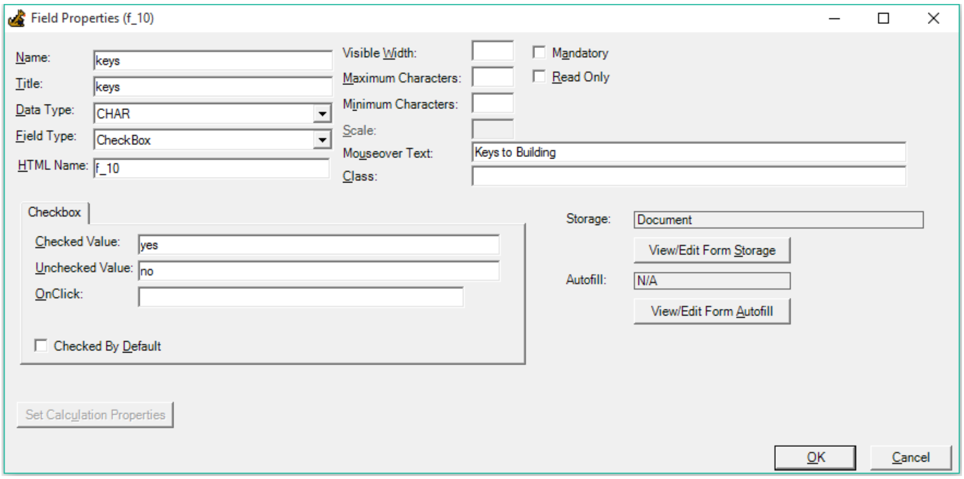
5. Set the properties specific to this field type:

   - **Checked Value**: Enter the value of the check box if checked.

   - **Unchecked Value**: Enter the value of the check box if unchecked.

   - **OnClick**: Optionally define an OnClick event that calls Custom JavaScript when the field is clicked. See Add Custom JavaScript for more information.

   - **Checked By Default**: Turn on and the check box will be checked by default.



Field Properties (f_10)

Name: keys
Title: keys
Data Type: CHAR
Field Type: CheckBox
HTML Name: f_10
Visible Width:
Maximum Characters:
Minimum Characters:
Scale:
Mouseover Text: Keys to Building
Class:
☐ Mandatory
☐ Read Only

Checkbox

Checked Value: yes
Unchecked Value: no
OnClick:
☐ Checked By Default

Storage: Document
Autofill: N/A

6. Click **OK**. The field properties are saved.

Radio Button

Limit the user to a single selection from a set of options. Radio buttons are added as a group and there is a single storage location for that group; the value of the selected radio button is stored.

To add a radio button:

1. Click on the form where you want to add the field.

2. Click **Add Radio Field** in the toolbar. A radio button is added to the form.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.

   Notes:

   - The **Title** is shared for all radio buttons in a group. If you change it in one radio button, it will change for the others.

   - If autofill is configured for a radio button, the radio button will be selected if the autofill value matches the field value.

5. Set the properties specific to this field type:

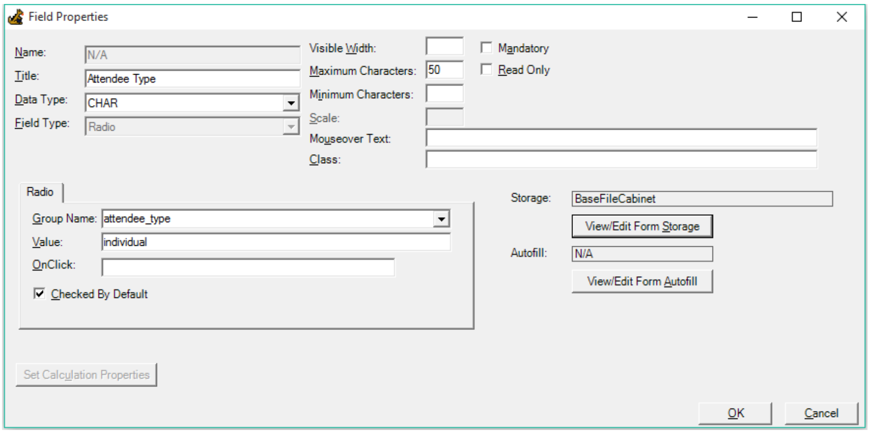
   - **Group Name**: Either enter the name of the radio button group or select an existing group from the list.

   - **Value**: Enter the value of the radio button.

   - **OnClick**: Optionally define an OnClick event that calls Custom JavaScript when the field is clicked. See Add Custom JavaScript for more information.

   - **Checked By Default**: Turn on and this radio button will be checked by default.

     Note: Only one radio button in a group can be Checked By Default.



Name: N/A
Title: Attendee Type
Data Type: CHAR
Field Type: Radio
Visible Width:
Maximum Characters: 50
Minimum Characters:
Scale:
Mouseover Text:
Class:
☐ Mandatory
☐ Read Only
Group Name: attendee_type
Value: individual
OnClick:
☑ Checked By Default
Storage: BaseFileCabinet
Autofill: N/A

6. Click **OK**. The field properties are saved.

Text Area

Enter multiple lines of text as input.

To add a text area:

1. Click on the form where you want to add the field.

2. Click **Add Text Area Field** in the toolbar and select **Standard Textarea**. A text area is added to the form.

3. Double-click the field to open the field properties.

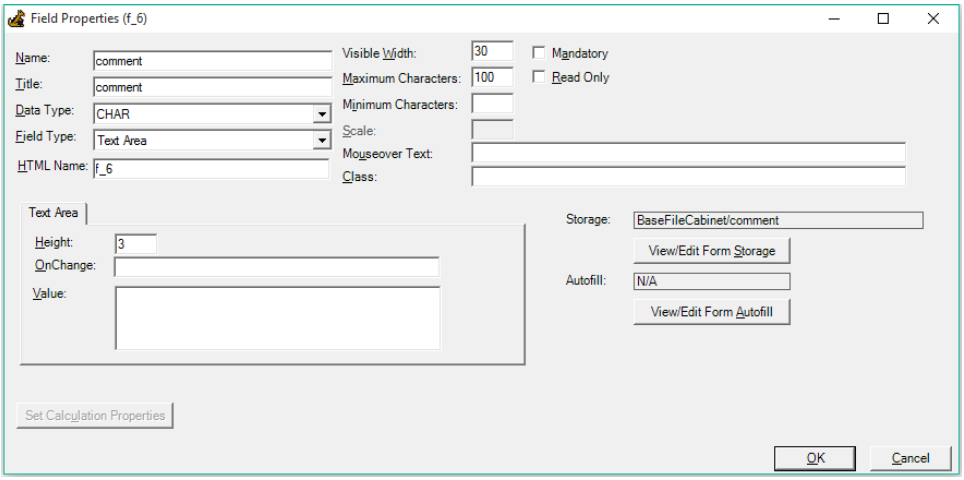
4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.

5. Set the properties specific to this field type:

   - **Height**: The height of the text area

   - **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.

   - **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.



Field Properties (f_6)

Name: comment
Title: comment
Data Type: CHAR
Field Type: Text Area
HTML Name: f_6
Visible Width: 30
Maximum Characters: 100
Minimum Characters:
Scale:
Mouseover Text:
Class:
☐ Mandatory
☐ Read Only

Text Area
Height: 3
OnChange:
Value:

Storage: BaseFileCabinet/comment
Autofill: N/A

6. Click **OK**. The field properties are saved.

Rich Text Editor

Enter multiple lines of text as input. The text can be formatted and is stored as rich text.

To add a rich text editor:

1. Click on the form where you want to add the field.

2. Click **Add Text Area Field** button in the toolbar and select **Rich Text Editor**. A rich text editor is added to the form.

   In the Designer, the rich text editor is represented by an icon . The rich text editor will display as a text area with formatting options on the live form when viewed in a browser.

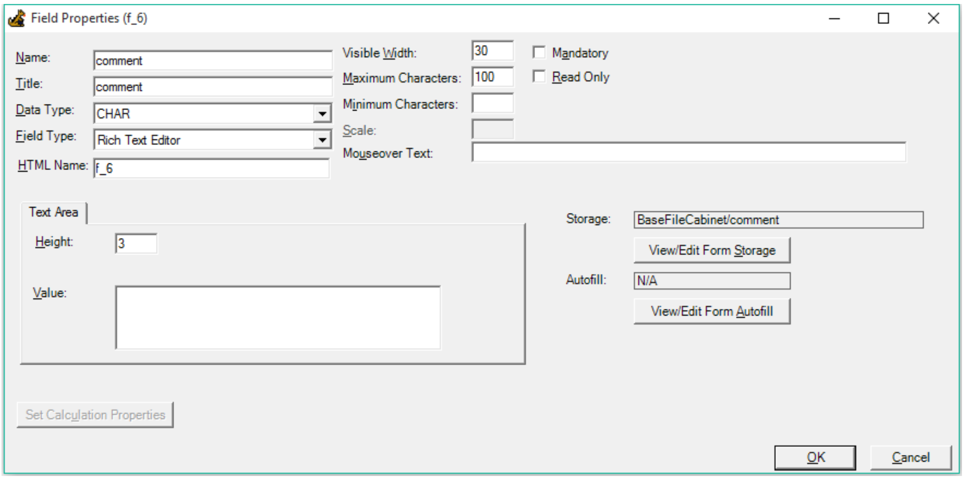
3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.

5. Set the properties specific to this field type:

   - **Height**: The height of the rich text editor.

   - **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.



Field Properties (f_6)

Name: comment
Title: comment
Data Type: CHAR
Field Type: Rich Text Editor
HTML Name: f_6

Visible Width: 30
Maximum Characters: 100
Minimum Characters:
Scale:
Mouseover Text:

□ Mandatory
□ Read Only

Text Area
Height: 3
Value:

Storage: BaseFileCabinet/comment
Autofill: N/A

6. Click **OK**. The field properties are saved.

Password

Similar to a textbox except characters entered are displayed as asterisks (*) for privacy.

To add a password field:

1. Click on the form where you want to add the field.

2. Click **Add Password Field** in the toolbar. A password field is added to the form.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.

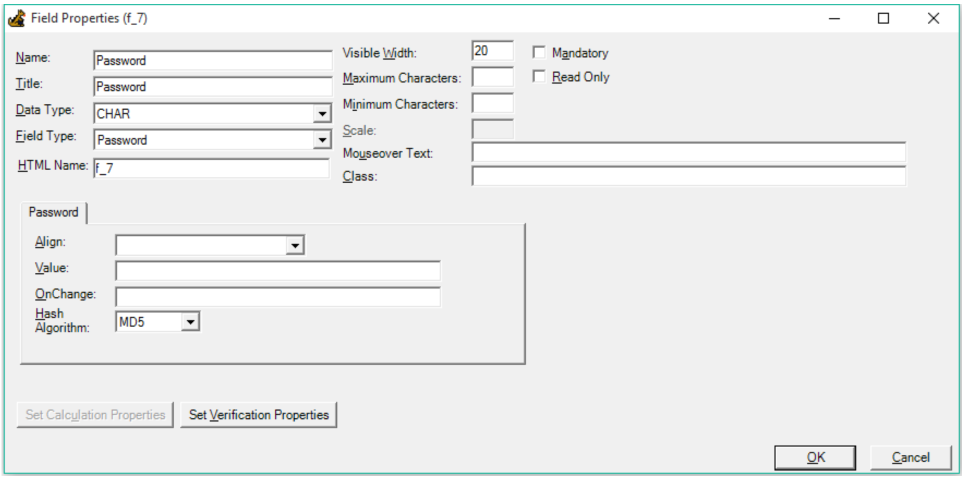
5. Set the properties specific to this field type:

   - **Align**: The alignment of text within the field. Options are **Left**, **Right** and **Center**.

   - **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

   - **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.

   - **Hash Algorithm**: Choices includes no hash (blank), **MD5**, or **SHA1**.



Name: Password
Title: Password
Data Type: CHAR
Field Type: Password
HTML Name: f_7
Visible Width: 20
Maximum Characters:
Minimum Characters:
Scale:
Mouseover Text:
Class:
☐ Mandatory
☐ Read Only

Password
Align:
Value:
OnChange:
Hash Algorithm: MD5

6. Click **OK**. The field properties are saved.

Hidden

Store data that is not visible on the form. Since users will not be able to enter a value into a hidden field, the field should be configured with either a default value, calculation, or autofill.

Warning: Do *not* use hidden fields to store sensitive data. Although hidden fields do not display on the form, hidden fields and their values can be viewed in the form's source HTML.

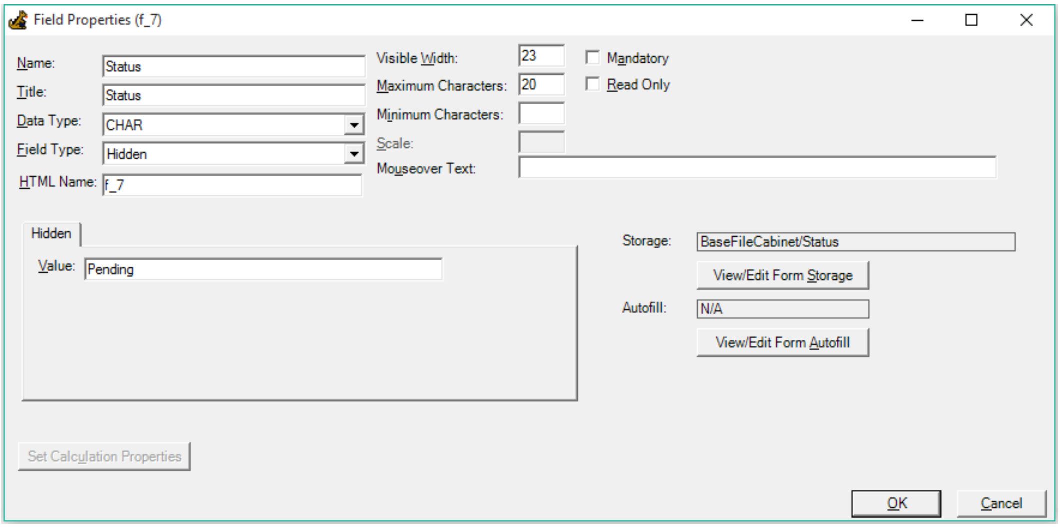
Tip: If you have a lot of hidden fields on your form, instead of using the **Hidden** form field, put an invisible table set to display as **none** at the bottom of your form. You can put anything in there and it won't display on the live form, such as textboxes and text labels for your fields, and so on. See Add Tables for more information.

To add a hidden field:

1. Click on the form where you want to add the field.
2. Click **Add Hidden Field** in the toolbar. A hidden field is added to the form.

   In the Designer, the hidden field is represented by an icon ⬚. The hidden field will not display on the live form when viewed in a browser.
3. Double-click the field to open the field properties.
4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.
5. Optionally set a default **Value** for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

   Note: If you do not set a value, you need to set a calculation or autofill, because otherwise the hidden field will be blank.



Field Properties (f_7)

Name: Status
Title: Status
Data Type: CHAR
Field Type: Hidden
HTML Name: f_7
Visible Width: 23
Maximum Characters: 20
Minimum Characters:
Scale:
Mouseover Text:
□ Mandatory
□ Read Only

Hidden
Value: Pending

Storage: BaseFileCabinet/Status
Autofill: N/A

6. Click **OK**. The field properties are saved.

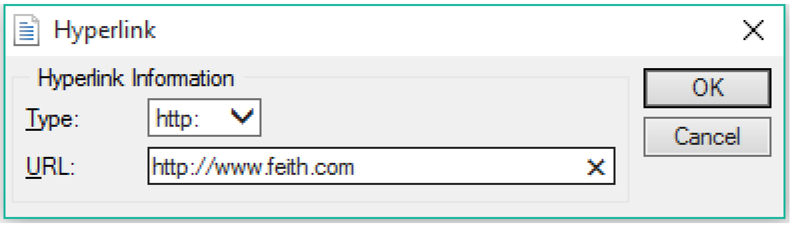
Hyperlink

Standard Hyperlink

A basic hyperlink with no additional settings.

To add a standard hyperlink:

1. Click on the form where you want to add the link.
2. Click **Add Hyperlink** in the toolbar and select **Standard Hyperlink**. The **Hyperlink** dialog opens.
3. Select the hyperlink **Type**.
4. Enter the hyperlink **URL**.



Hyperlink Information

Type: http:

URL: http://www.feith.com

5. Click **OK**. The standard hyperlink is added to the form with the **URL** as the link's text.
6. To change the display text of the link, position the cursor within the text and edit the text as needed.

Go to feith.com

To modify a standard hyperlink:

1. Place your cursor within the text of the standard hyperlink.
2. Click **Add Hyperlink** in the toolbar and select **Standard Hyperlink**. The **Hyperlink** dialog opens for the selected standard hyperlink.
3. Make changes as desired.
4. Click **OK**. The standard hyperlink is modified.

Custom Hyperlink

A hyperlink you can customize with various settings, including form field tokens.

To add a custom hyperlink:

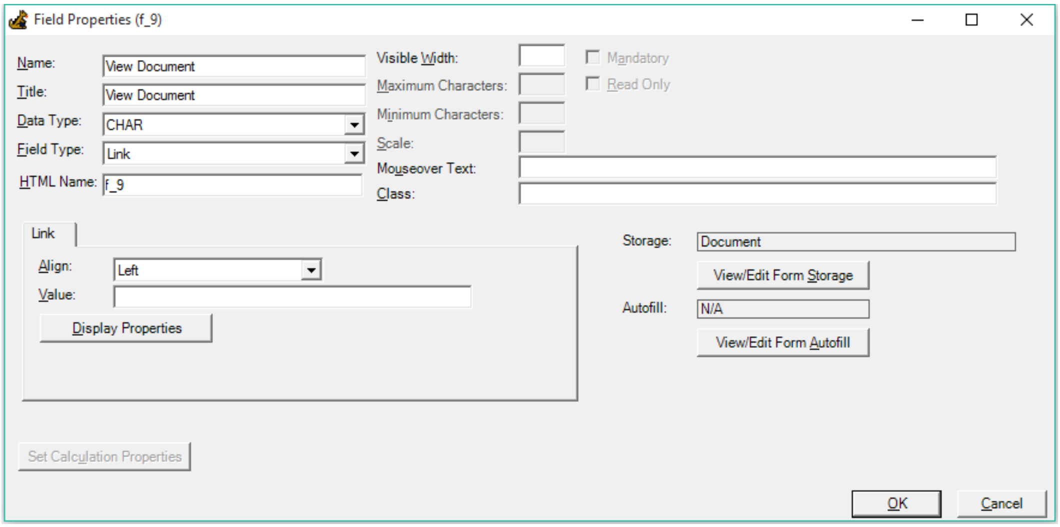
1. Click on the form where you want to add the link.

2. Click **Add Hyperlink** in the toolbar and select **Custom Hyperlink**. A custom hyperlink is added to the form.

   In the Designer, the custom hyperlink is represented by an icon Link. When the live form is viewed in a browser, the display text or image assigned in the field properties will display on the form as a hyperlink.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.


Field Properties (f_9)

Name: View Document

Title: View Document

Data Type: CHAR

Field Type: Link

HTML Name: f_9

Visible Width:

Maximum Characters:

Minimum Characters:

Scale:

Mouseover Text:

Class:

□ Mandatory

□ Read Only

Link

Align: Left

Value:

Storage: Document

Autofill: N/A

5. Click **Display Properties** to configure the custom hyperlink:

   - **Display Text:** Enter the text to display on the form as a hyperlink. Optionally **Add Field Tokens**.

   - Choose the type of URL you want to set:

     o **Plain URL:** Enter the **URL**; optionally **Add Field Tokens**. If you include FDD user name or login tokens in the URL, turn on **WebFDD or Forms iQ Link**. Turn on **Stay in Browser** to have the link open in the same browser window, otherwise it will open in a new browser window.

     Note: If you include FDD user name or login tokens in the URL for a **WebFDD or Forms iQ Link**, the form must require login. See Login Permissions for more information.

  - **OnClick event:** Enter the **OnClick** event; optionally **Add Field Tokens**. An OnClick event calls Custom JavaScript when the field is clicked. See Add Custom JavaScript for more information.

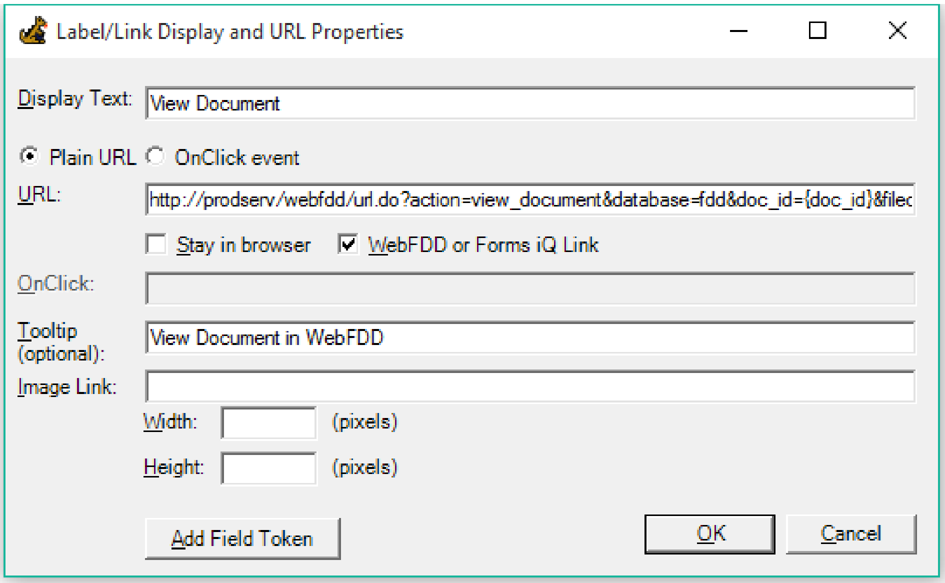
    Note: When using a token in a link, make sure to handle the value appropriately depending on the type of field the value is coming from. For example, you *need* quotes around the token for a textbox field receiving **John Smith** as input because the resulting value in the token would be **document.fddForm.f_1.value**. As another example, you *do not need* quotes around token from a label field already set to value **John Smith** because the resulting value in the token would be **John Smith**.

- **Tooltip:** Optionally enter text to display as a tooltip for the link.

- **Image Link:** Optionally enter link to an image you want to use as the object the user clicks to launch the link. You can set the image **Width** and **Height**.

- To add a field token to the **Display Text**, **URL**, or **OnClick**, place your cursor where you want to add the token then click **Add Field Token** and select the desired **Form Field** token or **WebFDD Token**. The token is added.

  Note: The **Display Text** can only accept certain tokens, such as read-only, hidden, and label fields.



Label/Link Display and URL Properties

Display Text: View Document

● Plain URL ○ OnClick event

URL: http://prodserv/webfdd/url.do?action=view_document&database=fdd&doc_id={doc_id}&filec

□ Stay in browser ☑ WebFDD or Forms iQ Link

OnClick:

Tooltip (optional): View Document in WebFDD

Image Link:

Width: (pixels)

Height: (pixels)

6. Click **OK**. The display properties are saved.

7. Click **OK**. The field properties are saved.

Logout Hyperlink

If your form requires login, a logout link can be added so the user can click it to log out. See Login Permissions for more information.

To add a logout hyperlink:

1. Click on the form where you want to add the link.
2. Click **Add Hyperlink** in the toolbar and select **Logout**. A logout hyperlink is added to the form.
3. To change the display text of the link, position the cursor within the text and edit the text as needed.

Log out of form

Image

Add an image to the form. The image may be a standard image (from a file on your computer) or a linked image.

Standard Image

To add a standard image:

1. Click on the form where you want to add the image.
2. Click **Add Image** button in the toolbar and select **Standard Image**. The **Open** dialog opens.
3. Browse to select a .jpg, .gif, .png or .bmp file, then click **Open**. The image is added to the form.
4. Optionally set alternate text for the image. Right-click the image and select **Set Image Alt Tag**. Enter the desired text and click **OK**. The alternate text is set.

To change the picture of a standard image:

1. Right-click a standard image and select **Image Properties**. The **Open** dialog opens.
2. Browse to select a .jpg, .gif, .png or .bmp file, then click **Open**. The image is added to the form.

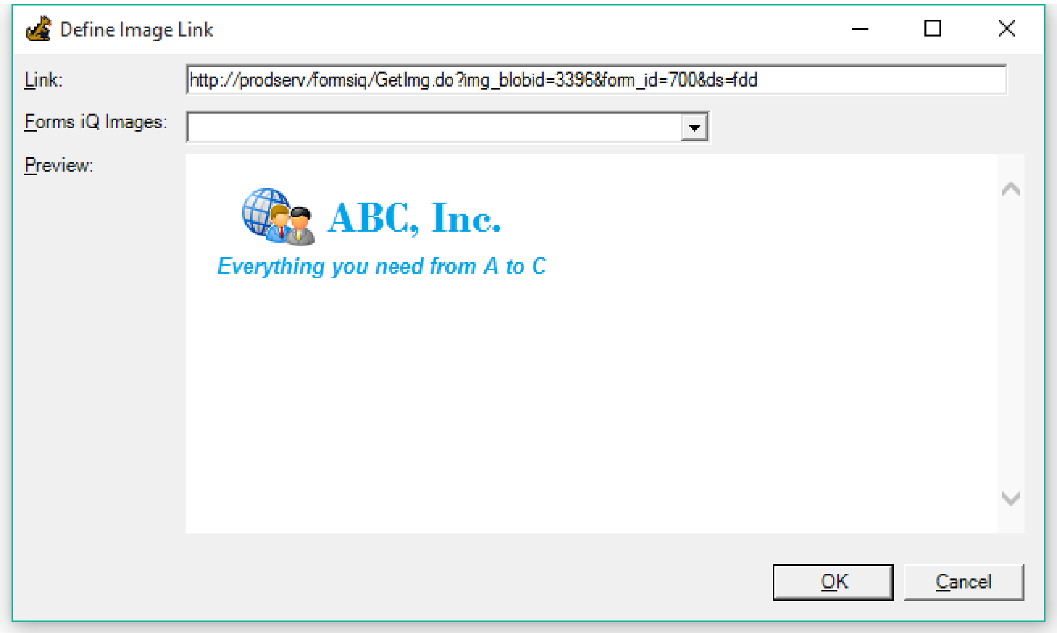
Linked Image

To add a linked image:

1. Click on the form where you want to add the image.
2. Click **Add Image** button in the toolbar and select **Link Image**. The **Define Image Link** dialog opens.
3. Enter the **Link** to the image, which should be a .jpg, .gif, .png or .bmp file.

   Any image you bring in as a **Standard Image** is automatically saved on your server to use as a linked image. Therefore, you may alternatively select an existing **Forms iQ Image** which will insert the **Link** for you.

4. A **Preview** of the image displays.



Define Image Link

Link: http://prodserv/formsiq/GetImg.do?img_blobid=3396&form_id=700&ds=fdd

Forms iQ Images:

Preview:

5. Click **OK**. The image is added to the form.

6. Optionally set alternate text for the image. Right-click the image and select **Set Image Alt Tag**. Enter the desired text and click **OK**. The alternate text is set.

To change the picture of a linked image:

1. Right-click a linked image and select **Image Properties**. The **Define Image Link** dialog opens.

2. Change the link as desired and click **OK**. The image is added to the form.

Lookup

Provide the user with a list of values. Two columns of data can be displayed in a lookup field.

To add a lookup field:

1. Click on the form where you want to add the field.

2. Click **Add Lookup Control** in the toolbar. A lookup field is added to the form.

3. Double-click the field to open the field properties.

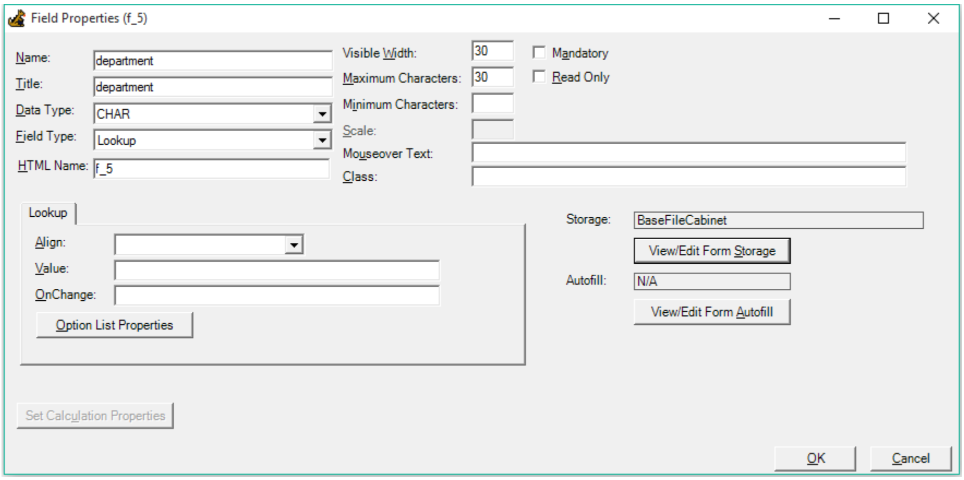
4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill** and **Storage**. See Common Field Properties for more information.

5. Set the properties specific to this field type:

    - **Align**: The alignment of text within the field. Options are **Left**, **Right** and **Center**.

    - **Value**: The optional default value of the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

    - **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.



Field Properties (f_5)

Name: department

Title: department

Data Type: CHAR

Field Type: Lookup

HTML Name: f_5

Visible Width: 30

Maximum Characters: 30

Minimum Characters:

Scale:

Mouseover Text:

Class:

□ Mandatory

□ Read Only

Lookup

Align:

Value:

OnChange:

Storage: BaseFileCabinet

Autofill: N/A

6. Click **Option List Properties** to configure the lookup field's list using the following options:

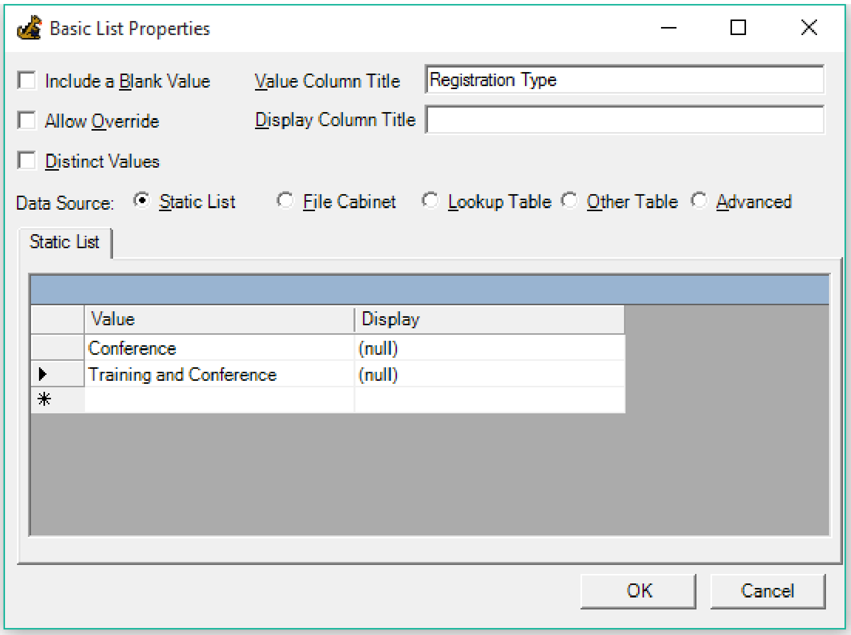
    - **Distinct Values**: Turn on and the option list will include only the distinct values from the selected value column in the data source.

      Note: This option is disabled if you choose the **Advanced** SQL **Data Source**.

    - **Value Column Title** and **Display Column Title**: Optionally enter friendly titles to appear as the column headers in the lookup field value list window when a user is filling out the form.

- **Data Source**: Select the data source for the list. Choices include:
  - **Static List:** Enter values for the list.

    Click in the row and enter the desired **Value**. Optionally enter a friendly **Display** value to show to the user next to the **Value**.



Basic List Properties

□ Include a Blank Value Value Column Title: Registration Type

□ Allow Override Display Column Title:

□ Distinct Values

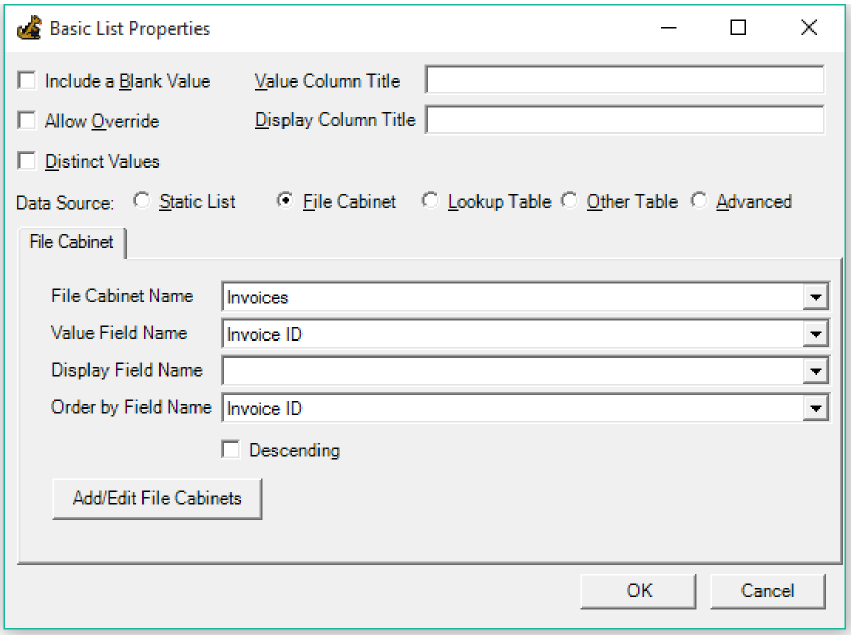
Data Source: ● Static List ○ File Cabinet ○ Lookup Table ○ Other Table ○ Advanced

Static List

Value	Display
Conference	(null)
Training and Conference	(null)

  - **File Cabinet:** Get values from a file cabinet.

    Select the **Field Cabinet Name** and **Value Field Name**. Optionally select a friendly **Display Field Name** to show to the user next to the value. Optionally select an **Order by Field Name** to order the values; the order is ascending by default but you can choose **Descending** instead.



Basic List Properties

□ Include a Blank Value Value Column Title:

□ Allow Override Display Column Title:

□ Distinct Values

Data Source: ○ Static List ● File Cabinet ○ Lookup Table ○ Other Table ○ Advanced

File Cabinet

File Cabinet Name: Invoices

Value Field Name: Invoice ID

Display Field Name:

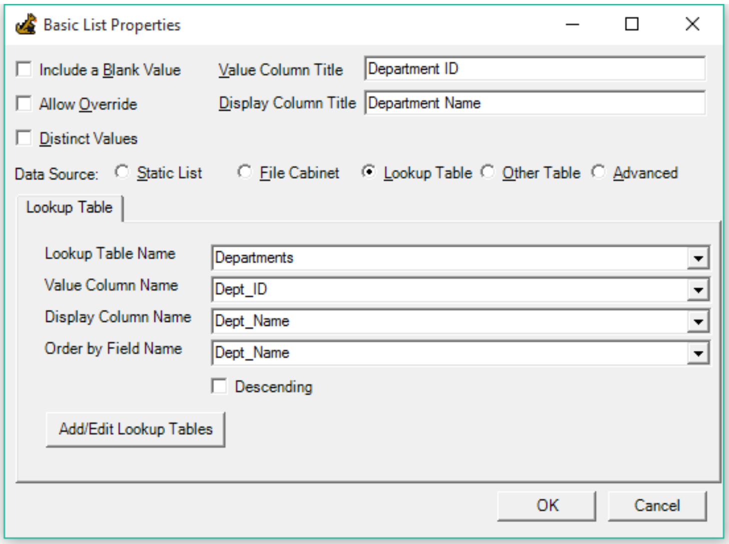
Order by Field Name: Invoice ID

□ Descending

- **Lookup Table:** Get values from a lookup table.

  Select the **Lookup Table Name** and **Value Column Name**. Optionally select a friendly **Display Column Name** to show to the user next to the value. Optionally select an **Order by Field Name** to order the values; the order is ascending by default but you can choose **Descending** instead.

  Alternatively, you can hand-type into these fields to use any table or view in your FDD database as the list. Make sure to type in the table/view name and column names correctly.



Basic List Properties

□ Include a Blank Value — Value Column Title: Department ID

□ Allow Override — Display Column Title: Department Name

□ Distinct Values

Data Source: ○ Static List ○ File Cabinet ● Lookup Table ○ Other Table ○ Advanced

Lookup Table

Lookup Table Name: Departments

Value Column Name: Dept_ID

Display Column Name: Dept_Name

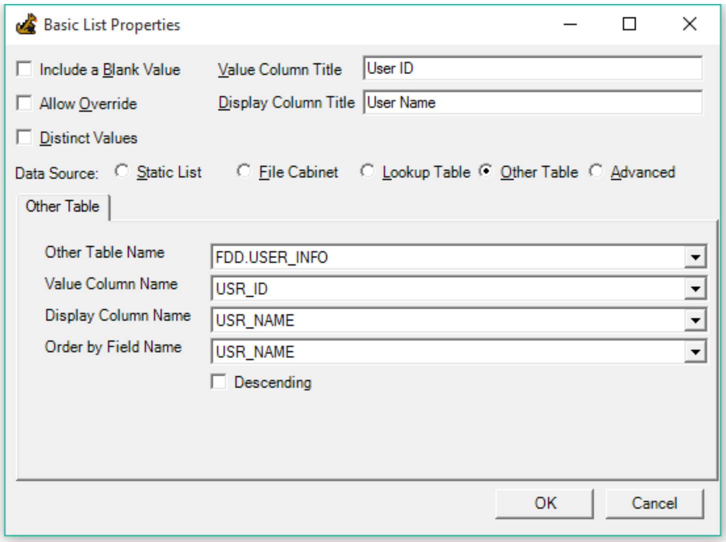
Order by Field Name: Dept_Name

□ Descending

- **Other Table:** Get values from another table.

  Select the **Other Table Name** and **Value Column Name**. Optionally select a friendly **Display Column Name** to show to the user next to the value. Optionally select an **Order by Field Name** to order the values; the order is ascending by default but you can choose **Descending** instead.

  Note: The table must be in your Other Tables List in order to select it here. See Maintain Other Tables List for more information.



Basic List Properties

□ Include a Blank Value — Value Column Title: User ID

□ Allow Override — Display Column Title: User Name

□ Distinct Values

Data Source: ○ Static List ○ File Cabinet ○ Lookup Table ● Other Table ○ Advanced

Other Table

Other Table Name: FDD.USER_INFO

Value Column Name: USR_ID

Display Column Name: USR_NAME

Order by Field Name: USR_NAME

□ Descending

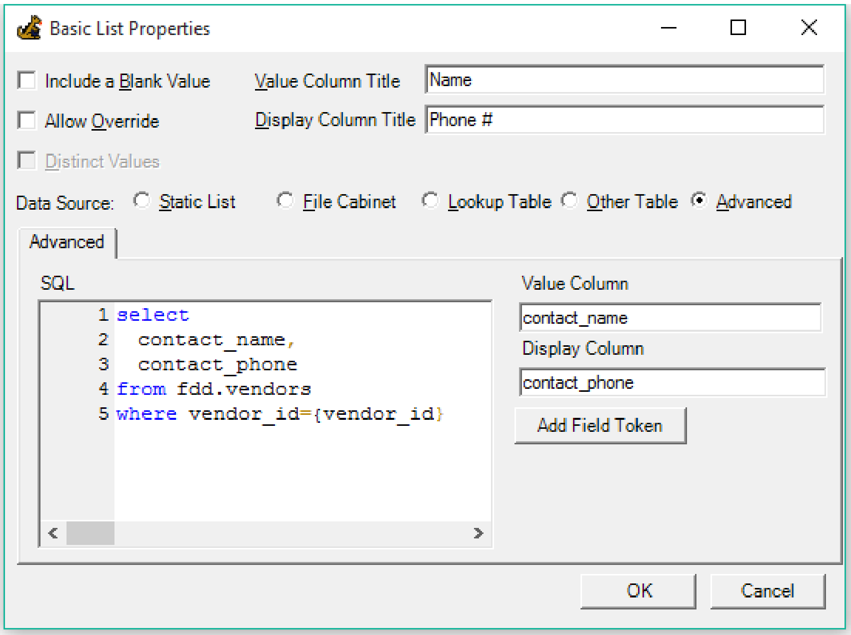
- **Advanced:** Get values from a SQL query.

  Enter the **SQL** query, in which you may **Add a Field Token**. Also enter the **Value Column**. Optionally enter a friendly **Display Column** to show to the user next to the value.

  Tip: You can format your text using **TAB** to indent and **SHIFT+TAB** to un-indent.

  Notes:

  - This option requires SQL knowledge and the **Edit Forms iQ SQL** task permission set in Feith Control Panel.
  - Make sure any field token you use is or is not enclosed in single quotes, as is appropriate for the type of data.
  - Using a token of a list field (e.g. STRING-LIST) is not supported.



Basic List Properties

□ Include a Blank Value

□ Allow Override

□ Distinct Values

Value Column Title: Name

Display Column Title: Phone #

Data Source: ○ Static List ○ File Cabinet ○ Lookup Table ○ Other Table ● Advanced

Advanced

SQL

```
select
  contact_name,
  contact_phone
from fdd.vendors
where vendor_id={vendor_id}
```

Value Column: contact_name

Display Column: contact_phone

- **Note:** When you set a **Display** in addition to a **Value**, both the Display and Value are shown to the user when they are filling out the form, but it is the Value that is actually stored.

7. Click **OK**. The option list properties are saved.

8. Click **OK**. The field properties are saved.

Label

Display text that cannot be edited. Since users will not be able to enter a value into a label field, the field should be configured with either a default value, calculation, or autofill.

To add a label:

1. Click on the form where you want to add the field.

2. Click **Add Label Control** in the toolbar. A label field is added to the form.

   In the Designer, the label is represented by an icon . The label displays as text when the live form is viewed in the browser.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, **Data Type**, **Autofill**, and **Storage**. See Common Field Properties for more information.

5. Set the properties specific to this field type:

   - **Align**: The alignment of text within the field. Options are **Left**, **Right** and **Center**.

   - **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

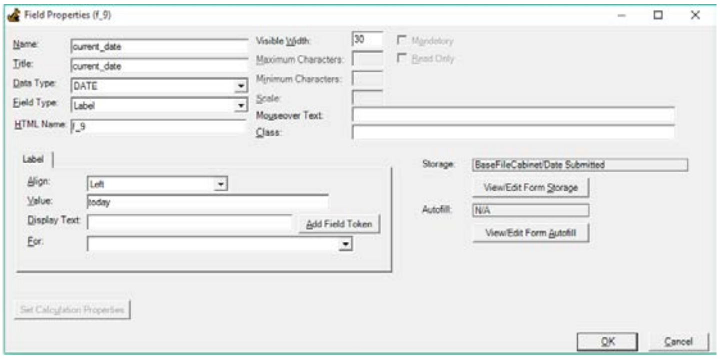
     Note: If you do not set a value, you need to set a calculation or autofill, because otherwise the label will be blank.

   - **Display Text:** Optionally enter text to display on the form. This text can differ from the actual value of the label field that is stored. If this is left blank, the value of the label will display instead.

     You can insert a token for another field in the Display Text by clicking **Add Field Token** and selecting a **Form Field**.

     Note: The Display Text can only accept certain tokens, such as read-only, hidden, and label fields.

   - **For:** Optionally define which field this label is for.

     Field Properties (f_9)

     Name: current_date
     Title: current_date
     Data Type: DATE
     Field Type: Label
     HTML Name: f_9
     Visible Width: 30
     Maximum Characters:
     Minimum Characters:
     Scale:
     Mouseover Text:
     Class:
     □ Mandatory
     □ Read Only

     Label
     Align: Left
     Value: today
     Display Text:
     For:

     Storage: BaseFileCabinet/Date Submitted
     Autofill: N/A

6. Click **OK**. The field properties are saved.

Dynamic List

Allow the user to enter multiple lines of detail pertaining to a single item. For example, a dynamic list might be used on an expense report form to capture multiple lines of expenses.

Tip: You can ask the Designer to automatically create columns in the dynamic list based on the columns in an autofill object's source, such as a lookup table. After selecting the autofill's **Source**, just click **Auto-create List**. See Autofill Dynamic List for more information.

To add a dynamic list:

1. Click on the form where you want to add the field.

2. Click **Add Dynamic List Control** in the toolbar. A dynamic list is added to the form.

   In the Designer, the dynamic list is represented by an icon . The dynamic list will display as rows of fields on the live form when viewed in a browser.

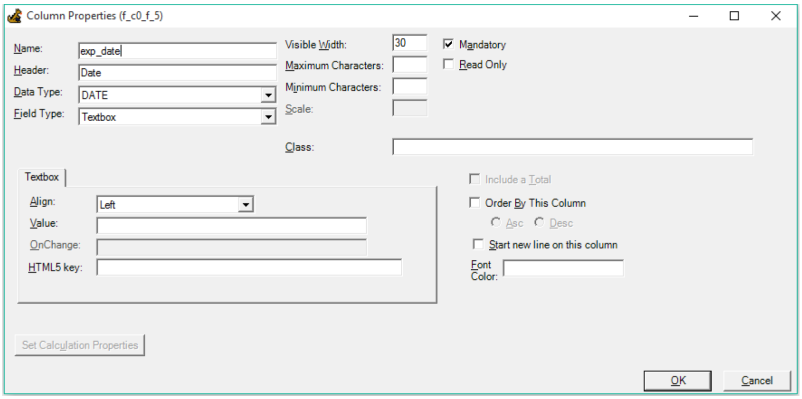
3. Double-click the field. The **Dynamic List** dialog opens.

4. Enter the **Name** of the dynamic list. This name is shown in the Designer's interface, such as when selecting field tokens for use in calculations.

5. Add columns to the dynamic list:

   a. To the right of the **List Columns** list, click **Add**. The **Column Properties** dialog opens.

   b. Enter the column properties. See Common Column Properties and Column Properties Specific to Field Type for more information.



Column Properties (f_c0_f_5)

Name: exp_date
Header: Date
Data Type: DATE
Field Type: Textbox
Visible Width: 30
Maximum Characters:
Minimum Characters:
Scale:
☑ Mandatory
☐ Read Only
Class:

Textbox
Align: Left
Value:
OnChange:
HTML5 key:

☐ Include a Total
☐ Order By This Column
○ Asc ○ Desc
☐ Start new line on this column
Font Color:

   c. Click **OK**. The column is added.

   You can **Edit**, reorder, and **Remove** columns as needed.

6. Click **View/Edit Form Storage** to choose where the dynamic list's data will store. You can store the data to the document or another table. See Store Dynamic List Data in Table for more information.

7. Click **View/Edit Form Autofill** to autofill the dynamic list with data. See Autofill Dynamic List for more information.

8. Set various dynamic list properties:

   - **Font**: Choose a font and size to apply to the text entered in the dynamic list.

- **Default Number of Rows**: Enter the number of rows to be added to the dynamic list by default.

- **Label for Total**: Enter the label for a total, if any, in the dynamic list. Defaults to **Total**. Only applicable if one or more dynamic list columns include a total.

- **Include Row Numbers**: Turn on and each row in the dynamic list will be numbered.

- **Disable Add/Remove Row**: Turn on and the user will not be able to add a row or remove a row in the dynamic list. This option is typically used in combination with autofill to create a report on the form.

- **Disable Reorder Rows**: Turn on and the user will not be able to reorder rows in the dynamic list.

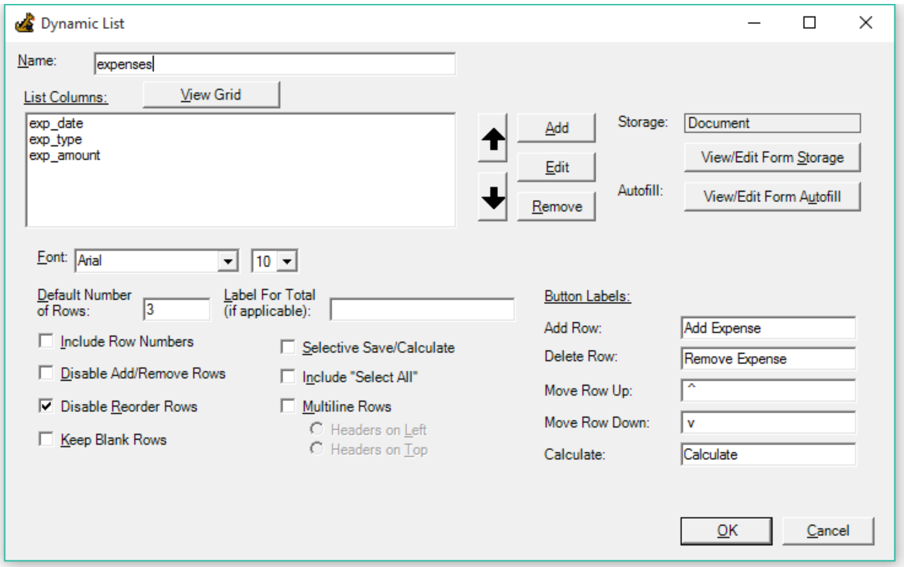
- **Keep Blank Rows**: Turn on and blank rows will be kept in the list when the form is submitted. If turned off, blank rows will be deleted from the list when the form is submitted.

- **Selective Save/Calculate**: Turn on and only checked rows will be saved when the form is submitted. In addition, if any columns have a total, only checked rows will be included in the total.

- **Include "Select All":** Turn on and a checkbox is added to the dynamic list's header. Click it to select all or deselect all the dynamic list rows.

- **Multi-line Rows**: Turn on and the dynamic list can display each row's fields on multiple lines. Choose either the **Headers on Left** or **Headers on Top** option. **Edit** the column you want to start on a new line and check **Start new line on this column** in its column properties. Each row can have as many lines as you want.

- **Button Labels:** Change the labels on the dynamic list buttons.



Dynamic List

Name: expenses

List Columns:
exp_date
exp_type
exp_amount

Storage: Document

Autofill:

Font: Arial 10

Default Number of Rows: 3

Label For Total (if applicable):

□ Include Row Numbers
□ Disable Add/Remove Rows
☑ Disable Reorder Rows
□ Keep Blank Rows
□ Selective Save/Calculate
□ Include "Select All"
□ Multiline Rows
○ Headers on Left
○ Headers on Top

Button Labels:
Add Row: Add Expense
Delete Row: Remove Expense
Move Row Up: ^
Move Row Down: v
Calculate: Calculate

9. Click **OK**. The dynamic list's properties are saved.

Common Column Properties

The following column properties are common to most dynamic list column types. If the property is unavailable for your column type, then it cannot be set for that column type.

See the following for related information:

- How to add a dynamic list
- Dynamic list column properties specific to the field type

COLUMN PROPERTY	DESCRIPTION
Name	The column name. The column name is shown in the Forms iQ Designer interface. For example, the column name is shown when selecting field tokens for use in calculations.
Header	The column's heading name that will display on the form.
Data Type	The data type of the field. The data type options are **CHAR, DATE, DATETIME, INTEGER, SMALL INT, DECIMAL, MONEY, STRING-LIST, DATE-LIST, and NUMBER-LIST.** **Note:** For some field types, such as file upload fields and smart icons, the **Data Type** setting does not apply and is automatically set to **N/A**.
Field Type	The field type. By default it is a textbox and you can change it as needed. **Note:** A field of the data type **STRING-LIST**, **DATE-LIST**, or **NUMBER-LIST** must be the field type **Text Area** or **Lookup**.
Visible Width	The width of the field. This number of characters displays in the field without scrolling.
Maximum Characters	The maximum number of characters accepted for input. A number will be suggested based on where the field's data will be stored. **Note:** For **DECIMAL** and **MONEY** fields, the maximum number of characters counts *all* characters in the field, including numbers before and after the decimal point as well as the decimal point itself.
Minimum Characters	The minimum number of characters accepted for input.
Scale	The number of digits to the right of the decimal point accepted for input. This setting applies to **DECIMAL** and **MONEY** fields only.
Class	The HTML class of the field.
Mandatory	Turn on and the field will require input. Users will not be able to submit the form without entering a value in the field.
Read Only	Turn on and the field value will be read-only. Users will not be able to edit the field value.
Calculation Properties	Populate a field with a calculated value. See Calculation Properties for more information.

Include a Total	Turn on and the values for each row in the column will be totaled. This option is only available if the column data type is integer, small integer, decimal or money.
Order by this Column	Turn on and the rows in the dynamic list will be ordered by this column. Choose either an **Asc**ending order or **Desc**ending order.
Start a New Line on this Column	If you turned on **Multi-line Rows** in the dynamic list's properties, turn this on to start a new line with this column. The dynamic list will display each row's fields on multiple lines, one of the lines starting with this column. Each row can have as many lines as you want.

Column Properties Specific to Field Type

The following column properties vary depending on the field type of the column.

See the following for related information:

- How to add a dynamic list
- Dynamic list column properties common to most field types

View the field-type-specific column properties for:

Textbox

- **Align**: The alignment of text within the field. Options are **Left**, **Right** and **Center**.
- **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.
- **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.
- **HTML5 key:** The HTML input type of the field (e.g. email).

Select

- **Height**: The height of the field.
- **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.
- **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.
- **Fit to Data**: Turn on and the width of the field will adjust to fit the longest value in the list.
- Click **Option List Properties** to configure the select field's list, including the source of the list. See Select for more information.

Check Box

- **Checked Value**: Enter the value of the check box if checked.
- **Unchecked Value**: Enter the value of the check box if unchecked.
- **OnClick**: Optionally define an OnClick event that calls Custom JavaScript when the field is clicked. See Add Custom JavaScript for more information.
- **Align:** Optionally choose the vertical alignment of the checkbox. Options are **Top**, **Middle**, **Bottom**, and **Baseline**.
- **Checked By Default**: Turn on and the check box will be checked by default.

Text Area

- **Height**: The height of the text area
- **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.
- **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

Hidden

Optionally set a default **Value** for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

Note: If you do not set a value, you need to set a calculation or autofill, because otherwise the hidden field will be blank.

Lookup

- **Align**: The alignment of text within the field. Options are **Left**, **Right** and **Center**.
- **Value**: The optional default value of the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.
- **OnChange**: Optionally define an OnChange event that calls Custom JavaScript when the field is changed. See Add Custom JavaScript for more information.
- Click **Option List Properties** to configure the lookup field's list, including the source of the list. See Lookup for more information.

Label

- **Align**: The alignment of text within the field. Options are **Left**, **Right** and **Center**.

- **Value**: Optionally set a default value for the field. You can type in a static value or use one of the provided keywords (e.g. date of **TODAY**). See Default Value for more information.

  Note: If you do not set a value, you need to set a calculation or autofill, because otherwise the label will be blank.

- **Display Text:** Optionally enter text to display on the form. This text can differ from the actual value of label field that is stored. If this is left blank, the value of the label will display instead.

  You can insert a token for another field in the Display Text by clicking **Add Field Token** and selecting a **Form Field**.

  Note: The Display Text can only accept certain tokens, such as read-only, hidden, and label fields.

- **For:** Optionally define which field this label is for.

Link (Custom Hyperlink)

Click **Display Properties** to configure the custom hyperlink, including the URL, display text, tooltip, and more. See Custom Hyperlink for more information.

File Upload

Click **File Upload Properties** to set up specific rules for the file that is uploaded, including the maximum file size, file type, and more. See File Upload for more information.

Smart Icon

Click **Smart Icon Properties** to set up the images and rules for the smart icon. See Smart Icon for more information.

File Upload

Allow the user to browse and select a file to upload with the form.

To add a file upload field:

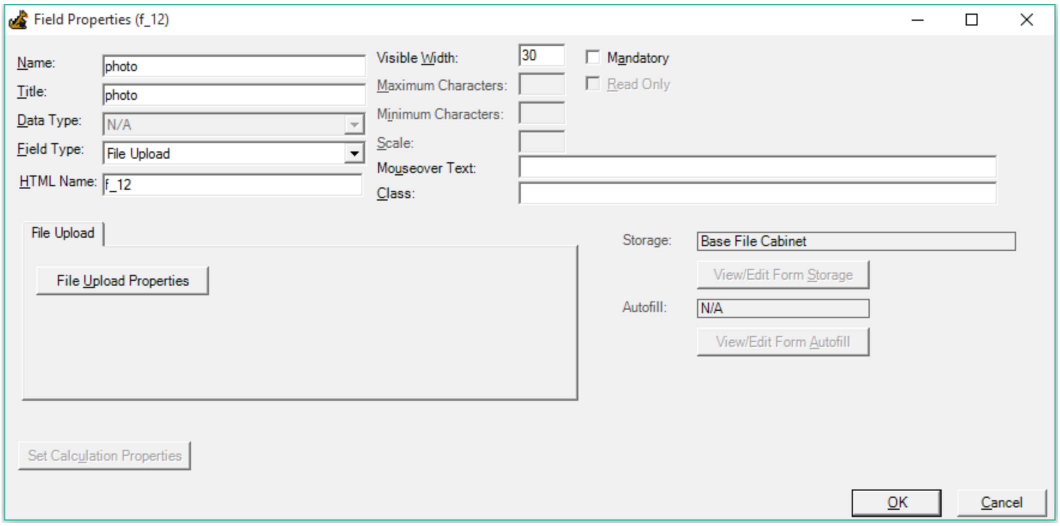
1. Click on the form where you want to add the field.

2. Click **Add File Upload Control** in the toolbar. A file upload field is added to the form.

   In the Designer, the file upload field is represented by an icon . The file upload field will display as a non-editable field with a browse button on the live form when viewed in a browser.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, and more. See Common Field Properties for more information.



Field Properties (f_12)

Name: photo
Title: photo
Data Type: N/A
Field Type: File Upload
HTML Name: f_12
Visible Width: 30
Maximum Characters:
Minimum Characters:
Scale:
Mouseover Text:
Class:
□ Mandatory
□ Read Only
Storage: Base File Cabinet
Autofill: N/A

5. Click **File Upload Properties** to set up specific rules for the file that is uploaded. Options are:

   - **Max File Size:** Optionally set a maximum file size. Enter a number and select a unit. Available units are **KB** (kilobyte) and **MB** (megabyte).

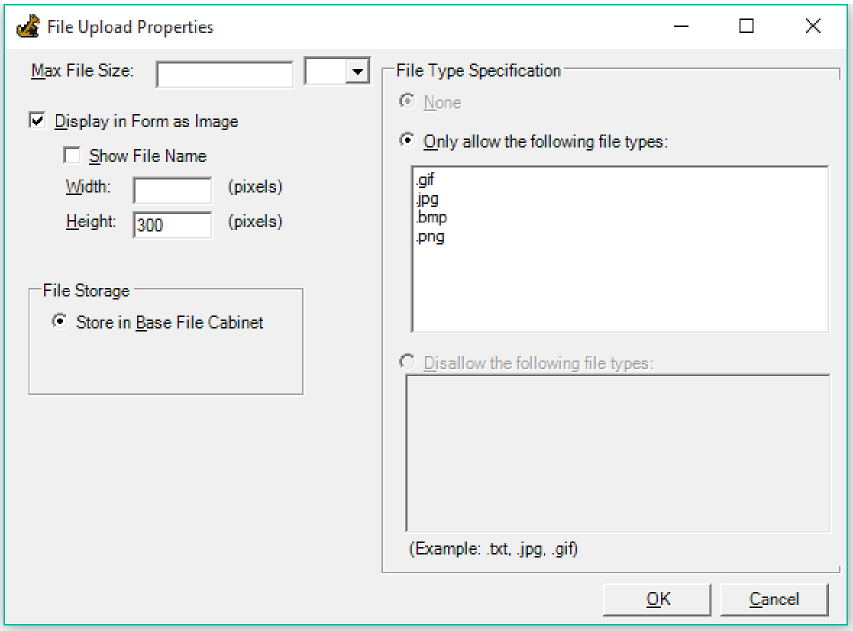
   - **Display in Form as Image:** Optionally display an uploaded image file in the form. Optionally turn on **Show File Name** to see the file name below the image. Set the **Width** and/or **Height** at which to display the image; the image will maintain its proportions if you only set one of the two.

     Note: This option only works for image files. Therefore, when you turn on **Display in Form as Image**, the **File Type Specification** setting changes to only allow image types that will be able to display in the form.

   - **File Storage:** When you submit the form, the uploaded file is **Stored in the Base File Cabinet** as another page with the form page in the resulting document.

     Note: If you created a file upload field in a previous version of Forms iQ, you may see a **Store in Form** option as well. This feature is no longer supported for new file upload fields, but if you have a file upload field created in a previous version of Forms iQ and chose the **Store in Form** option then it is still available to you.

- **File Type Specification:** Optionally restrict which file types are accepted for upload. Options are:
  - **None:** There is no restriction on what file types can be uploaded. All types are accepted.
  - **Only allow the following file types:** Only the files listed are accepted for upload. Enter file extensions for the files in the format **.ext** with a period and then the extension. When entering multiple file types, delimit them with a new line (hit **ENTER**), space, or tab.
  - **Disallow the following file types:** All files except those listed are accepted for upload. Enter file extensions for the files in the format **.ext** with a period and then the extension. When entering multiple file types, delimit them with a new line (hit **ENTER**), space, or tab.



Max File Size:

☑ Display in Form as Image

□ Show File Name

Width: (pixels)

Height: 300 (pixels)

File Storage

● Store in Base File Cabinet

File Type Specification

○ None

● Only allow the following file types:

.gif
.jpg
.bmp
.png

○ Disallow the following file types:

(Example: .txt, .jpg, .gif)

6. Click **OK**. The file upload properties are saved.

7. Click **OK**. The field properties are saved.

Smart Icon

Smart icons display one of many images based on a rule. Rules can be based on form fields or SQL queries. For example, you could use a smart icon to indicate whether a field's value was set correctly.

To add a smart icon:

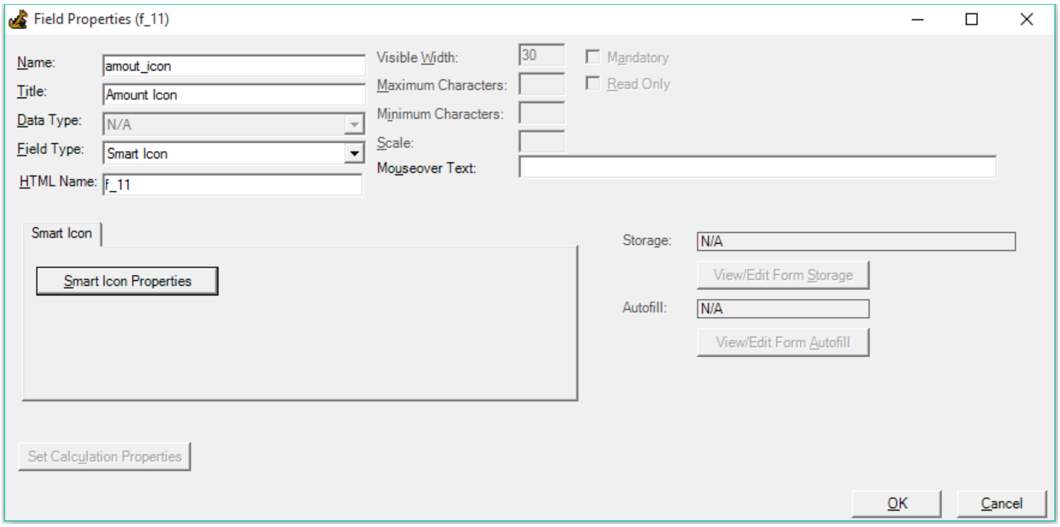
1. Click on the form where you want to add the field.

2. Click **Add Smart Icon Control** in the toolbar. A smart icon field is added to the form.

   In the Designer, the smart icon is represented by an icon 💡. The smart icon will display the configured images on the live form when viewed in a browser.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, and so on. See Common Field Properties for more information.



Field Properties (f_11)

Name: amout_icon

Title: Amount Icon

Data Type: N/A

Field Type: Smart Icon

HTML Name: f_11

Visible Width: 30

Maximum Characters:

Minimum Characters:

Scale:

Mouseover Text:

☐ Mandatory

☐ Read Only

Storage: N/A

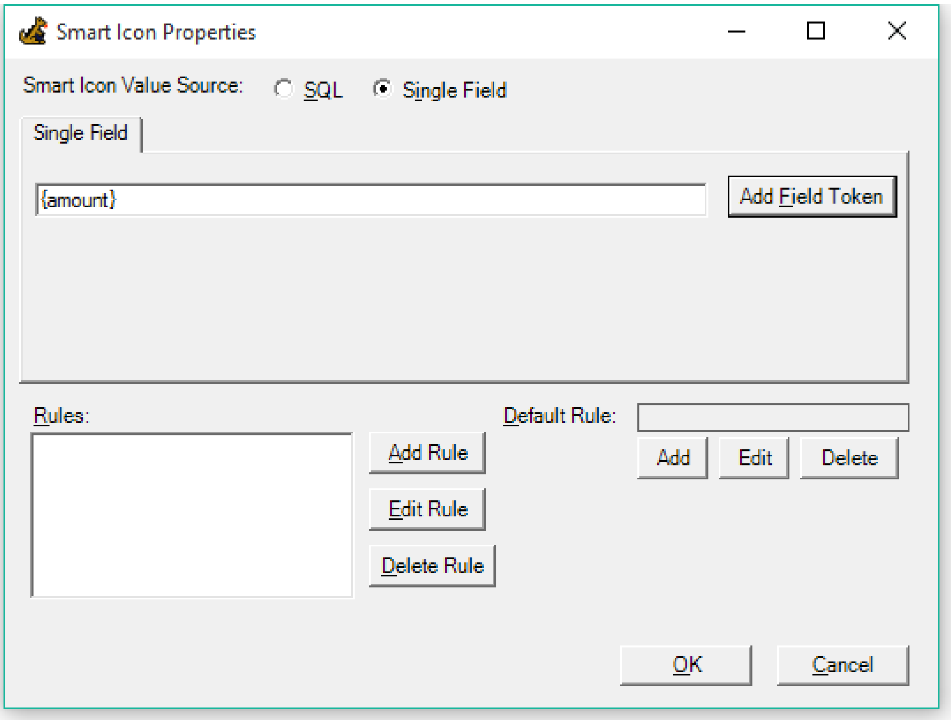
Autofill: N/A

5. Click **Smart Icon Properties** to set up the images and rules for the smart icon. The **Smart Icon Properties** dialog opens.

6. Select the **Smart Icon Value Source**. The value chosen here is compared to the rules you set up later. The result of that comparison determines which image is used for the icon. Options are:

   - **SQL:** Use a value returned from a SQL query. Enter the **SQL**, and optionally **Add a Field Token** in your SQL. Then type in the **Column Name** that contains the desired value. The value returned by this query will be compared to the rules you set up later.

   - **Single Field:** Use a value entered in a single form field. Click **Add Field Token** and select the desired field. The value in this field will be compared to the rules you set up later.

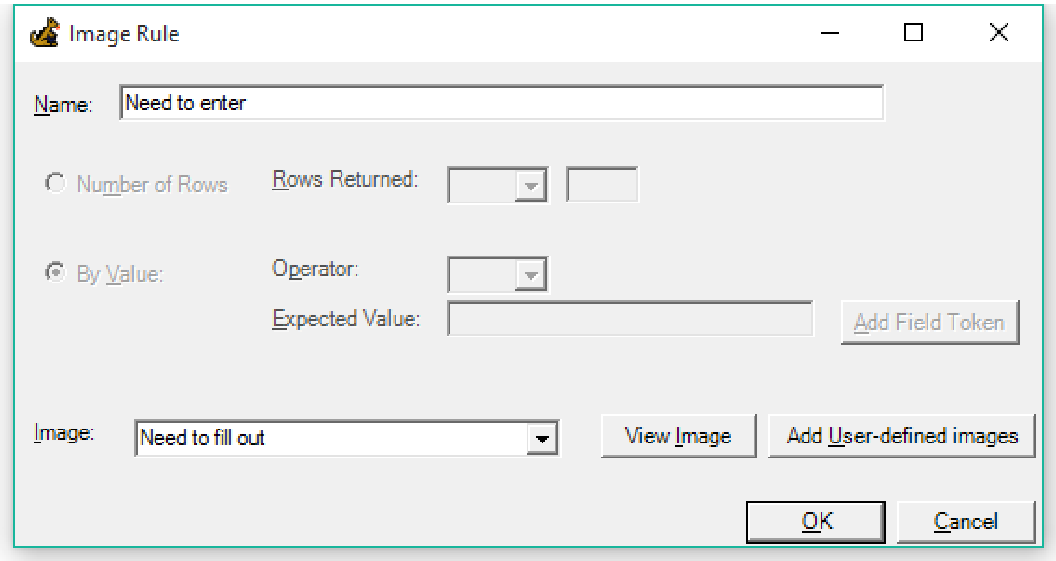


Smart Icon Value Source: ○ SQL ● Single Field

{amount}

7. Optionally add a **Default Rule** by clicking its **Add** button. In the **Image Rule** dialog, set the default rule's **Name** and select its **Image** or **Add a User-defined image**. If you need to **Add a User-defined image**, see Add Smart Icon Images for more information.

   Note: If a default rule is not set, a blank image will display on the form if none of the other rules' conditions are met.



Name: Need to enter

○ Number of Rows

● By Value:

Image: Need to fill out

8. Click **Add Rule** to set up rules that will be compared to the **Value Source** you selected previously. The result of that comparison determines which image is used for the icon. The **Image Rule** dialog opens.

9. Enter the rule's **Name**.

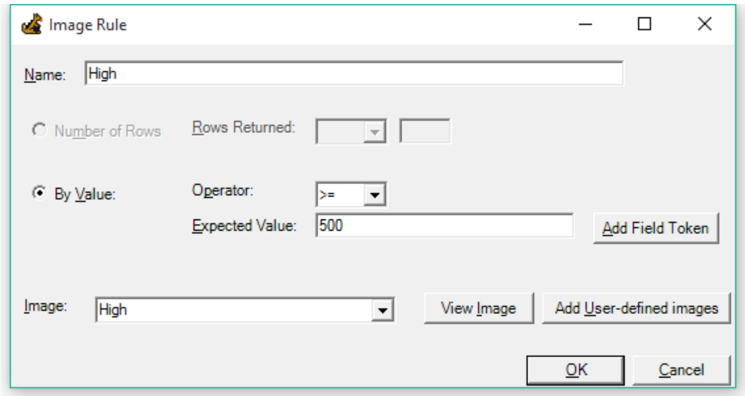
10. Choose the rule type and image. Options are:

    - **Number of Rows:** Select an operator and a number for the **Rows Returned**. Then select the **Image** or **Add a User-defined image** that will display when this rule is true. If the rows returned by your **SQL** in your **Smart Icon Value Source** meet this condition, the rule will take effect and its image will display.

      Note: This option is only available if you selected **SQL** as your **Smart Icon Value Source**.

    - **By Value:** Select an **Operator** and **Expected Value**. You may click **Add Field Token** to add a form field token to your **Expected Value**. Then select the **Image** or **Add a User-defined image** that will display when this rule is true. If the value returned by your **Smart Icon Value Source** meets this condition, the rule will take effect and its image will display.

    - If you need to **Add a User-defined image**, see Add Smart Icon Images for more information.



Name: High

○ Number of Rows   Rows Returned:

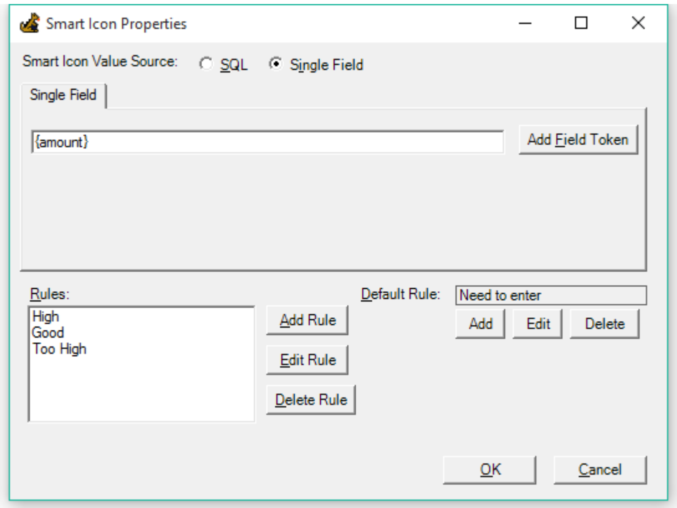
● By Value: Operator: >=

Expected Value: 500

Image: High

11. Click **OK**. The rule is added.

12. Continue to add rules as needed.



Smart Icon Value Source: ○ SQL ● Single Field

{amount}

Rules:
High
Good
Too High

Default Rule: Need to enter

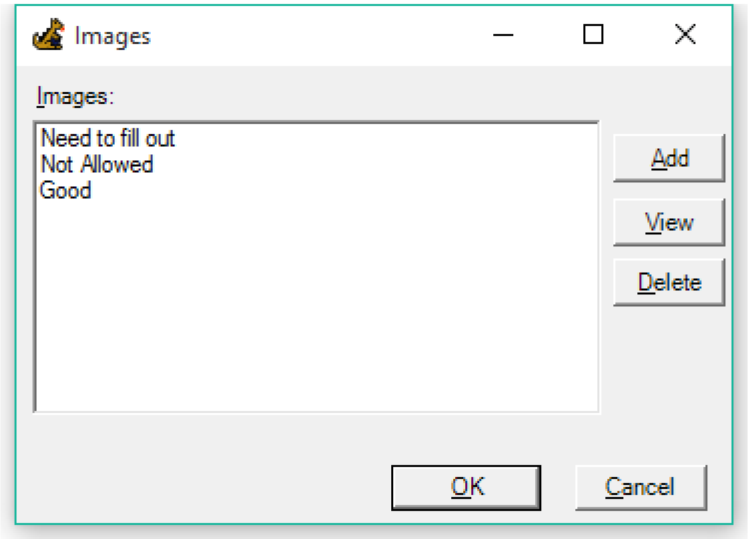
13. Click **OK**. The smart icon properties are saved.

14. Click **OK**. The field properties are saved.

Add Smart Icon Images

To add an image for a smart icon rule:

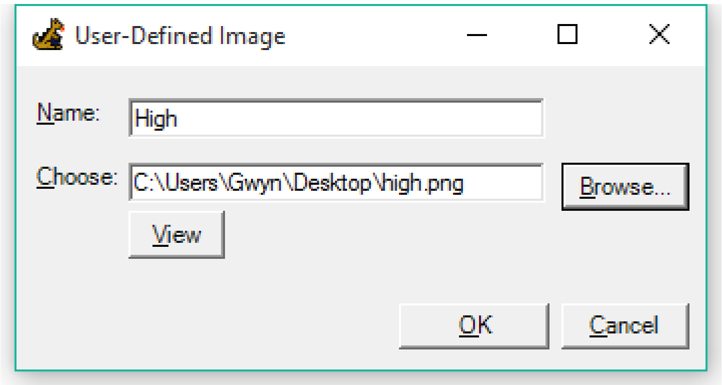
1. While configuring a smart icon rule, click **Add User-defined images**, or you can add images anytime under **Form>Images**. The **Images** dialog opens.



2. Click **Add**. The **User-Defined Image** dialog opens.

3. Enter the image **Name**.

4. Click **Browse** and select the desired image file.



Name: High

Choose: C:\Users\Gwyn\Desktop\high.png

5. Click **OK**. The image is added and can be selected for a smart icon rule.

Barcode

Display a barcode on the form. The barcode value can be the document ID, the page ID, a selected field value, or a constant value.

For example, put a barcode on a form so users can print the submitted form and us it as a cover page when mailing or faxing supporting documentation for the form. The barcode would be used to identify the supporting documentation when it is added to the FDD system using barcode recognition.

To add a barcode:

1. Click on the form where you want to add the field.

2. Click **Add Barcode Control** in the toolbar. A barcode is added to the form.

   In the Designer, the barcode is represented by an icon . The actual barcode will display on the form after the form is submitted or when viewing the submitted form (depending on your configuration).

3. Double-click the barcode field. The **Update Barcode** dialog opens.

4. Enter the barcode's **Name**.

5. Select the **Barcode Data** that will be the barcode's value. Options are:

   - **Doc ID:** The unique document ID of the submitted form document.

   - **Page ID:** The unique page ID of the submitted form page.

   - **Field:** The value of the specified form field. Choose the desired form field.

   - **Constant:** A constant value. Enter the constant value.

6. Select the **Symbology** to use for the barcode. Options are **Code128** and **Code39**.

7. If you chose **Code39** as your **Symbology**, optionally turn on **Check Sum** and a checksum will be included in the barcode.

   Tip: A checksum is a mathematical calculation that verifies the integrity of the barcode.

8. Optionally enter a **Prefix** value to include in the barcode data.

9. Optionally turn on **Draw Text** to display the value of the barcode below it in text format. Optionally specify the **Font Name**, **Font Style**, and **Font Size** of the text.

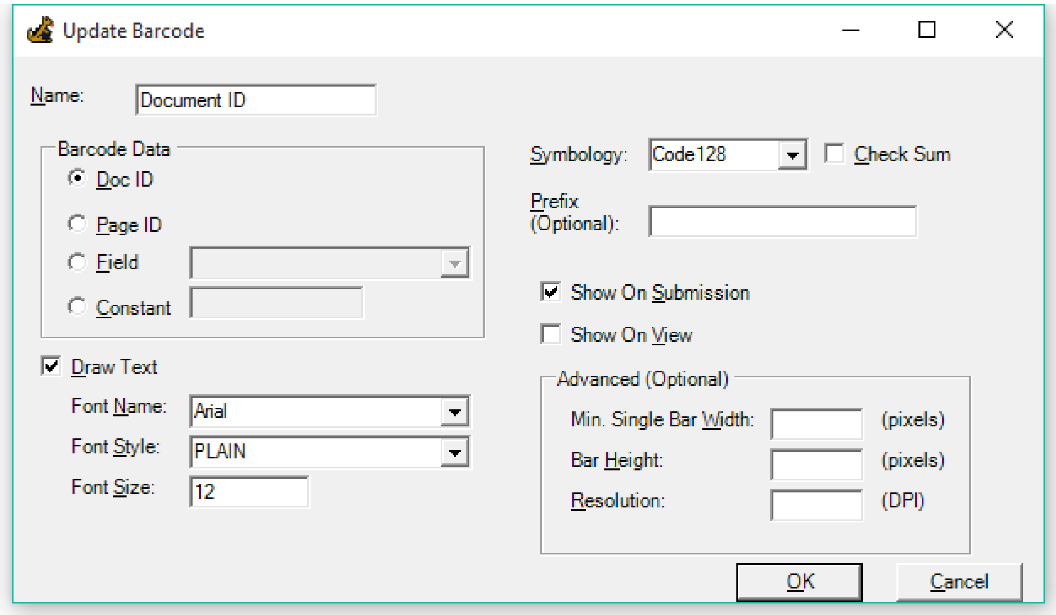
10. Turn on **Show On Submission** to display the barcode after the form is submitted on the success page.

11. Turn on **Show On View** to display the barcode when the form is viewed in the clients (FDD or WebFDD).

12. In the **Min. Single Bar Width**, optionally enter the desired width, in pixels, for the thinnest bar in the barcode. Other bars will adjust accordingly. The default is 2 pixels.

13. Optionally specify the desired **Bar Height**, in pixels, for the bars in the barcode.

14. Optionally enter the **Resolution**, in DPI (dots per inch). The default is 72 DPI.



Update Barcode

Name: Document ID

Barcode Data
● Doc ID
○ Page ID
○ Field
○ Constant

☑ Draw Text
Font Name: Arial
Font Style: PLAIN
Font Size: 12

Symbology: Code128 □ Check Sum
Prefix (Optional):

☑ Show On Submission
□ Show On View

Advanced (Optional)
Min. Single Bar Width: (pixels)
Bar Height: (pixels)
Resolution: (DPI)

15. Click **OK**. The barcode's properties are saved.

Signature

Allow the user to sign the form. The signature is based on standard PKI (Public Key Infrastructure).

Notes:

- A signature requires X509 certificates. Implementing this feature requires access to a certificate authority.
- The first time the end user opens a form with a signature, they will be prompted to install the Microsoft Capicom control, which is required for using signatures.

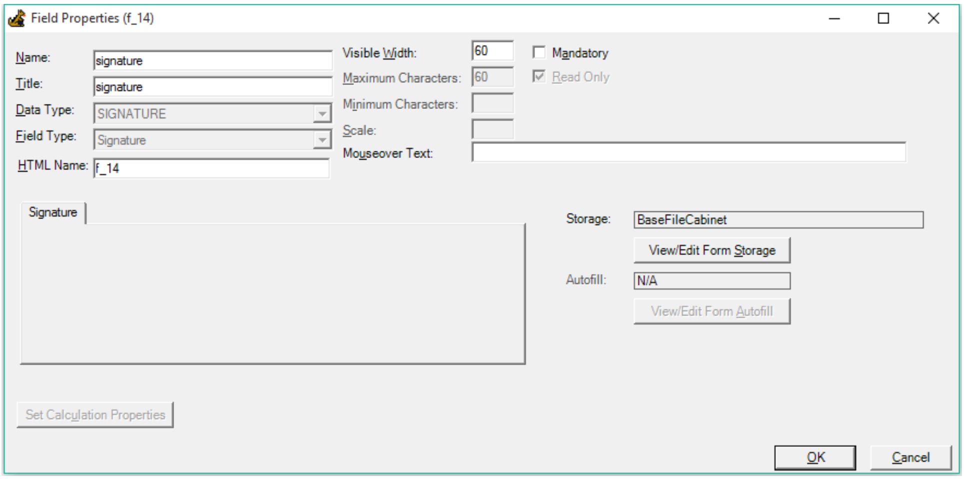
To add a signature:

1. Click on the form where you want to add the field.
2. Click **Add Signature Control** in the toolbar. A signature is added to the form.

   In the Designer, the signature is represented by an icon . The signature will display as a textbox with a "sign" button on the live form when viewed in a browser.
3. Double-click the field to open the field properties.
4. Set the common field properties, including **Name**, **Title**, and more. See Common Field Properties for more information.

   Notes:

   - A signature field is always read-only.
   - A signature field must have a data type of **SIGNATURE** and field type of **Signature**.
   - A signature field can only be stored to the file cabinet with a file cabinet field type of **Signature**.



Field Properties (f_14)

Name: signature

Title: signature

Data Type: SIGNATURE

Field Type: Signature

HTML Name: f_14

Visible Width: 60

Maximum Characters: 60

Minimum Characters:

Scale:

Mouseover Text:

☐ Mandatory

☑ Read Only

Storage: BaseFileCabinet

Autofill: N/A

5. Click **OK**. The field properties are saved.

Grid

Display data in report format. Grid provides several helpful features such as the ability to reorder columns, print, and export to Microsoft Excel.

To add a grid:

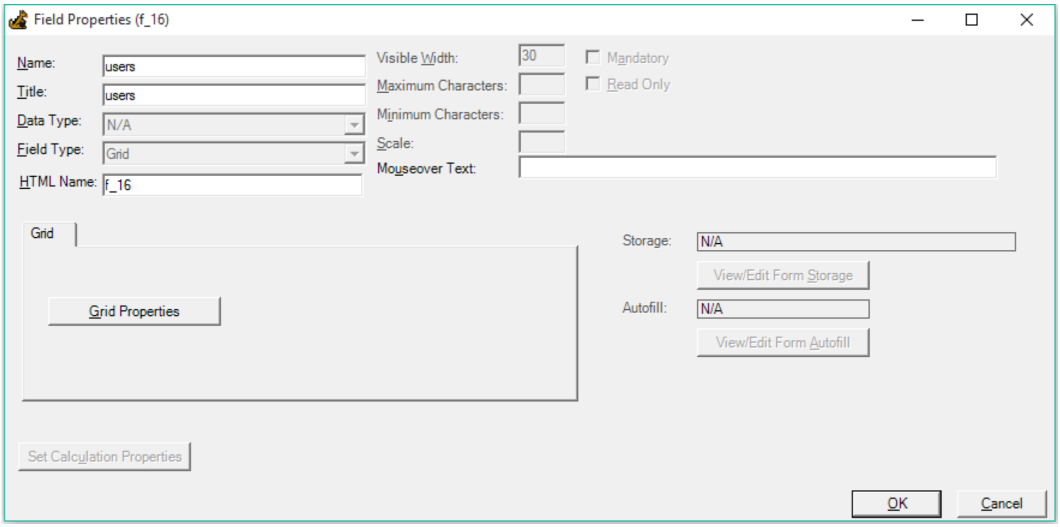
1. Click on the form where you want to add the field.

2. Click **Add Grid Control** in the toolbar. A grid is added to the form.

   In the Designer, the grid is represented by an icon ▦. The grid will display as a table full of data on the live form when viewed in a browser.

3. Double-click the field to open the field properties.

4. Set the common field properties, including **Name**, **Title**, and so on. See Common Field Properties for more information.



Field Properties (f_16)

Name: users

Title: users

Data Type: N/A

Field Type: Grid

HTML Name: f_16

Visible Width: 30

Maximum Characters:

Minimum Characters:

Scale:

Mouseover Text:

□ Mandatory

□ Read Only

Storage: N/A

Autofill: N/A

5. Click **Grid Properties** to configure the grid's data. The **Grid Properties** dialog opens.

6. Select the **View / Table Name** that contains the data you want to display in the grid.

7. Select the **Columns** to display:

   a. Double-click in the blank row. The **Tree Grid Column Properties** dialog opens.

   b. Select the desired **Column**.

   c. Optionally select the desired **Format**. Options include **NUMBER**, **MONEY**, and **DATE**. You can also have a sum at the bottom of the column by selecting **SUM**.

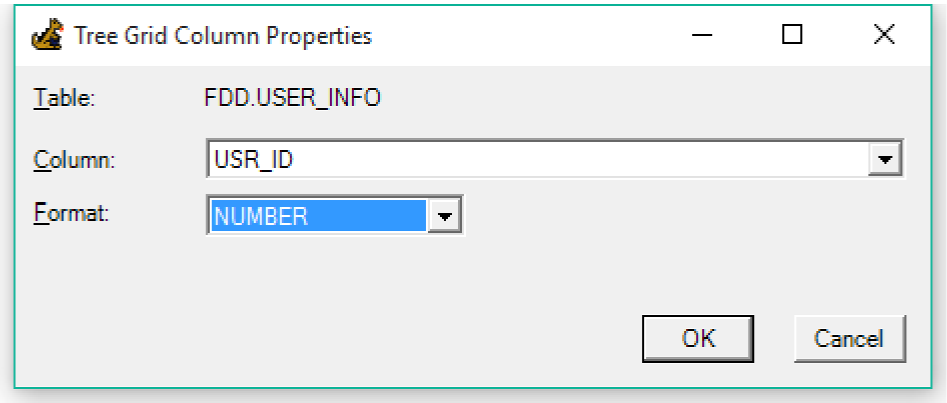


Table: FDD.USER_INFO

Column: USR_ID

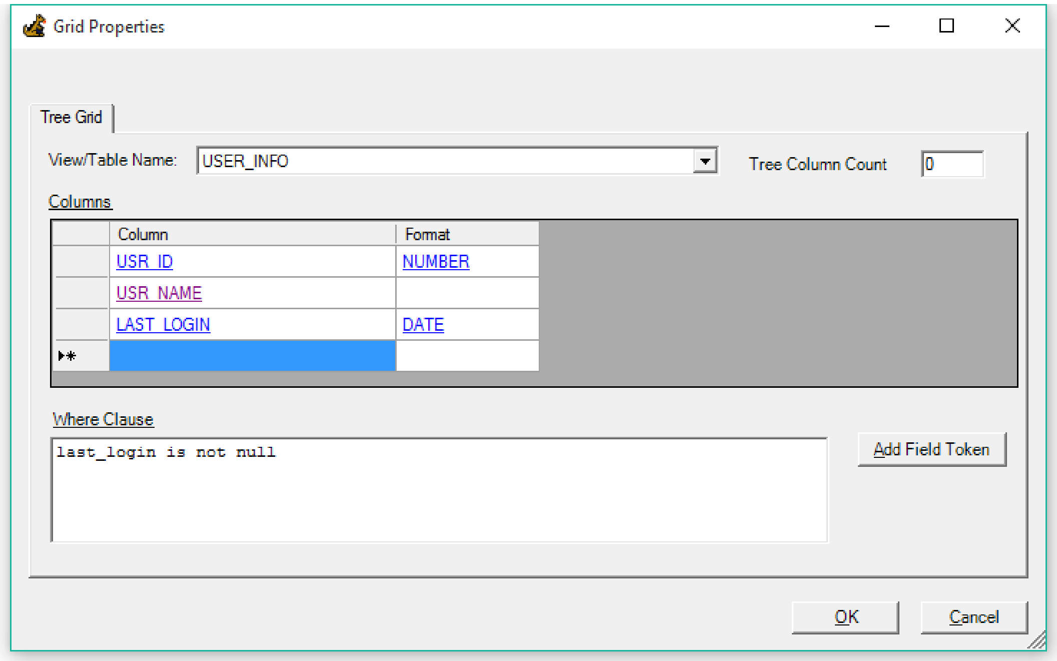
Format: NUMBER

d. Click **OK**. The column is added.

e. Continue to add columns as desired.

8. Optionally format some of your columns in a "tree" that you can collapse and expand. In **Tree Column Count**, enter how many columns you want in the tree, starting with the first column. For example, you could enter "2" and the first two columns "Country" and "City" would be in a tree format; expanding a Country would list all the cities therein.

9. Optionally enter a **Where Clause** to restrict which rows are brought back from the table. You may **Add a Field Token** for a form field in your where clause.



View/Table Name: USER_INFO

Tree Column Count: 0

	Column	Format
	USR_ID	NUMBER
	USR_NAME	
	LAST_LOGIN	DATE
▸*		

```
last_login is not null
```

10. Click **OK**. The grid properties are saved.

11. Click **OK**. The field properties are saved.

Autofill

Autofill

Automatically fill fields with data as the user works on the form. For example, a user might enter a name and then the address and phone number is filled in for them automatically. Autofill helps the user fill out the form more quickly and also assists in ensuring the form is filled out correctly.

TYPE OF AUTOFILL	HOW IS IT CONFIGURED?
Default Value The field is populated with the default value.	Enter a default value in the form field properties.
Autofill from Sign-In The field is autofilled with a value from the sign-in page. This option can be used to store values from the sign-in page, since the sign-in page does not save any information. This option is only available if your form includes a sign-in page. See Sign-In Page for more information.	Select a sign-in field in the form field properties.
Autofill from File Cabinet (FC) The field is autofilled with a value from a file cabinet field.	Add an autofill object in the Autofill editor.
Autofill from Lookup Table (LU) The field is autofilled with a value from a lookup table column.	Add an autofill object in the Autofill editor.
Autofill from Other Table The field is autofilled with a value from an “other” table column.	Add an autofill object in the Autofill editor.
Autofill from SQL The field is autofilled with a value from the results of a SQL query.	Add an autofill object in the Autofill editor.

Notes on autofilling certain field types:

- To autofill a URL value into a custom hyperlink, configure the autofill normally, filling in the custom hyperlink field, then set the custom hyperlink **URL** to the field token for the custom hyperlink field (e.g. **{vendor_site}**).
- To configure a check box or radio button to be selected by default, choose the **Checked By Default** option in the field properties.
- If autofill is configured for a check box or radio button, the check box/radio button will be selected if the autofill value matches the field value.
- The grid field type cannot be autofilled. Its data is set in the **Grid Properties** accessed in its field properties.

Default Value

The field is populated with the default value.

To set a default value:

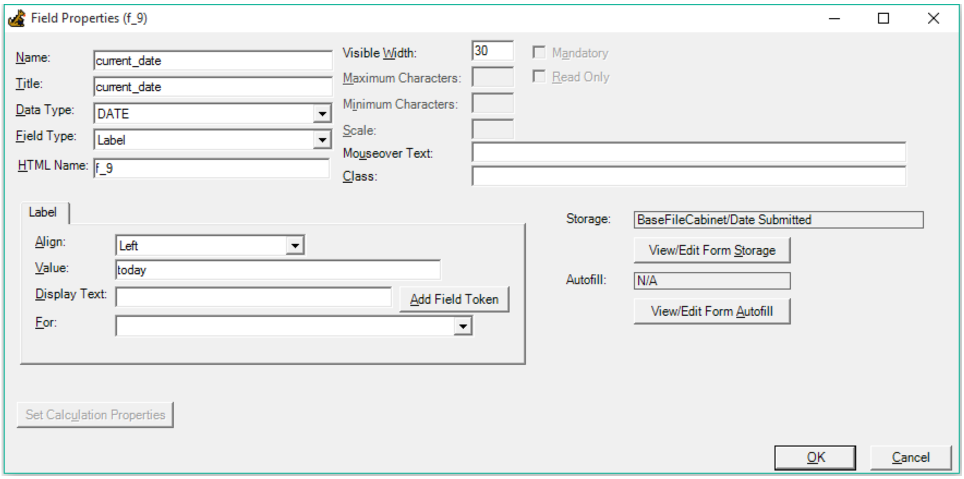
1. Double-click the field to open its field properties.
2. Set the default value based on the field type:

   - Textbox, select, text area, rich text, hidden, lookup, and label fields: Enter the default value in the **Value** field.

     The following keywords are supported as default values:

     - **SEQUENCE**: Defaults to an incrementing sequence number (per form set). Numeric fields only.
     - **TODAY**: Defaults to the current date. Date and datetime fields only.
     - **NOW**: Defaults to the current date and time. Datetime fields only.

   - Check box or radio button: Choose the **Checked By Default** option.



Field Properties (f_9)

Name: current_date
Title: current_date
Data Type: DATE
Field Type: Label
HTML Name: f_9
Visible Width: 30
Maximum Characters:
Minimum Characters:
Scale:
Mouseover Text:
Class:
□ Mandatory
□ Read Only

Label
Align: Left
Value: today
Display Text:
For:

Storage: BaseFileCabinet/Date Submitted
Autofill: N/A

3. Click **OK**. The field properties are saved.

Autofill from Sign-In

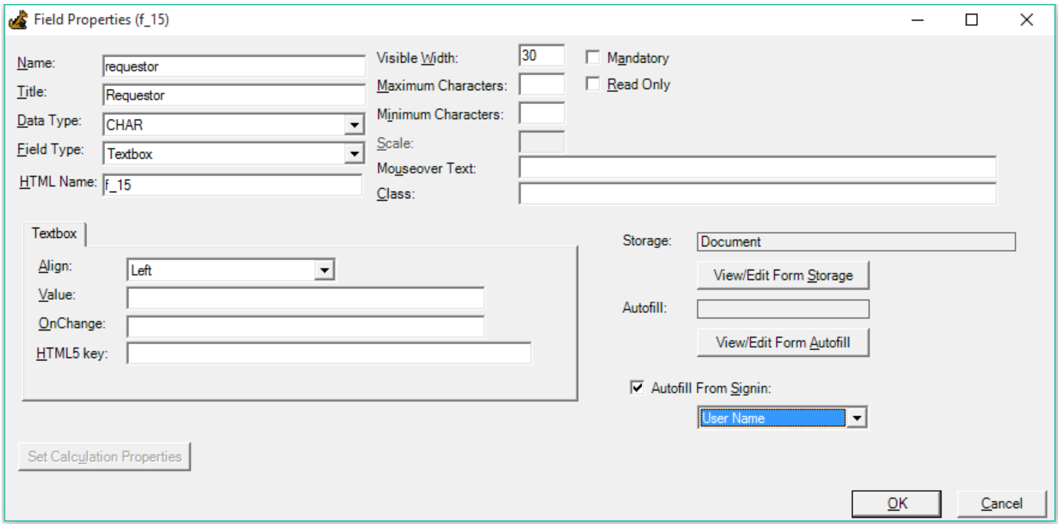
The field is autofilled with a value from the sign-in page. This option can be used to store values from the sign-in page, since the sign-in page does not save any information. This option is only available if your form includes a sign-in page. See Sign-In Page for more information.

To autofill from sign-in:

1. Double-click the field to open its field properties.

2. Turn on **Autofill From Signin** and select a sign-in field.

   Note: For check boxes and radio buttons, the field will be selected if the autofill value matches the field value.



Field Properties (f_15)

Name: requestor

Title: Requestor

Data Type: CHAR

Field Type: Textbox

HTML Name: f_15

Visible Width: 30

Maximum Characters:

Minimum Characters:

Scale:

Mouseover Text:

Class:

□ Mandatory

□ Read Only

Textbox

Align: Left

Value:

OnChange:

HTML5 key:

Storage: Document

Autofill:

☑ Autofill From Signin: User Name

3. Click **OK**. The field properties are saved.

Autofill from File Cabinet, Lookup Table or Other Table

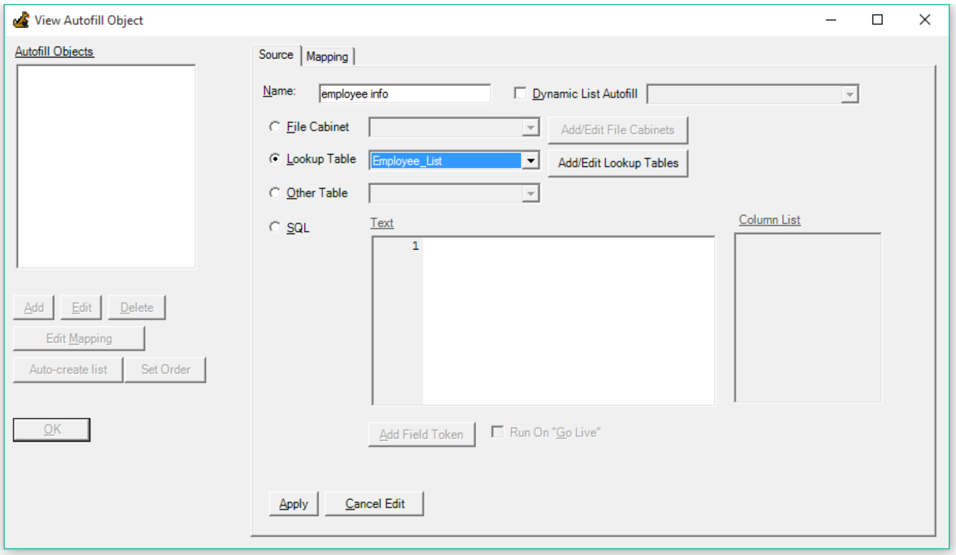
The field is autofilled with a value from a file cabinet field, lookup table column, or another table's column. You can set up the autofill to fill multiple form fields at once.

Note: If you want to autofill a dynamic list, see Autofill Dynamic List for instructions.

To autofill from a file cabinet, lookup table, or other table:

1. Select **Form>Autofill**. The **Autofill** dialog opens.
2. Click **Add**. The **Source** tab displays on the right.
3. Enter the autofill object's **Name**.
4. Select the autofill's source. Options are:
    - **File Cabinet:** Autofill data from file cabinet.
    - **Lookup Table:** Autofill data from lookup table
    - **Other Table:** Autofill data from another table.

    Note: The table must be in your Other Tables List in order to select it here. See Maintain Other Tables List for more information.



Name: employee info

☐ Dynamic List Autofill

○ File Cabinet

● Lookup Table Employee_List

○ Other Table

○ SQL

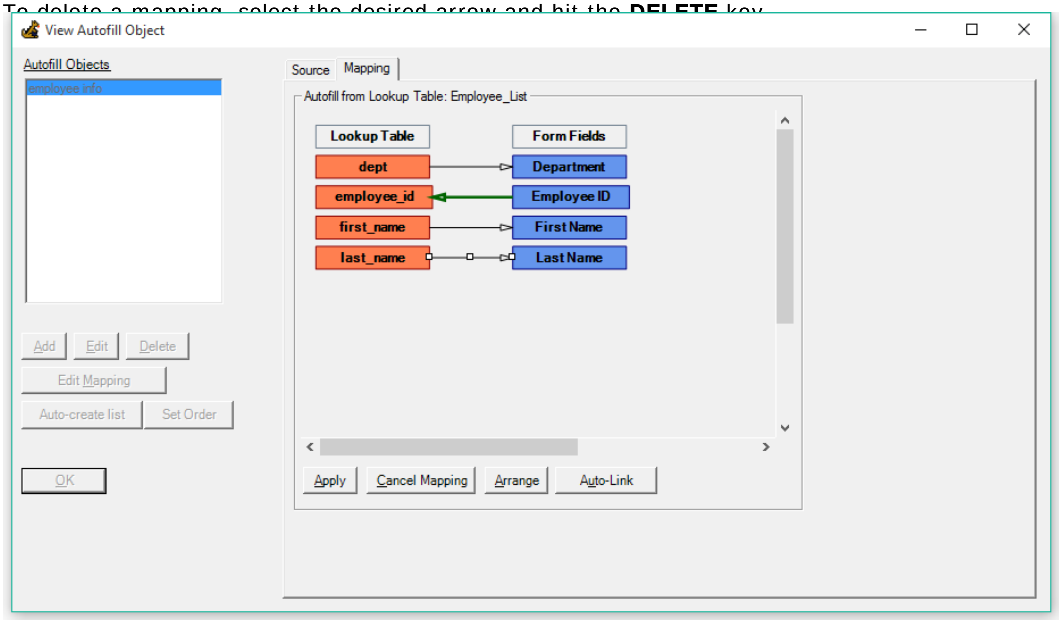
☐ Run On "Go Live"

5. Go to the **Mapping** tab. The columns from your source display on the left and the form fields display on the right.
6. Set the form fields that will be autofilled. Click on a source column on the left and drag to a form field on the right. An arrow is drawn from left to right, indicating that the value from that source column will be filled into that form field. Map as many fills as needed.

    To delete a mapping, select the desired arrow and hit the **DELETE** key.

7. Set the join condition that will determine which values are grabbed from the source's columns. Click on a form field on the right and drag to a source column on the left. A green arrow is drawn from right to left, indicating that the value in that form field will be used to retrieve the specific, desired values that need to be autofilled into the other form fields. Map as many joins as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.



8. Click **Apply**. The autofill's mapping is saved.

Note: Optionally set the order of file cabinet autofill objects, lookup table autofill objects, or other table autofill objects - defining the order in which the autofill objects (of a given type) run - by clicking **Set Order**.

Autofill from SQL

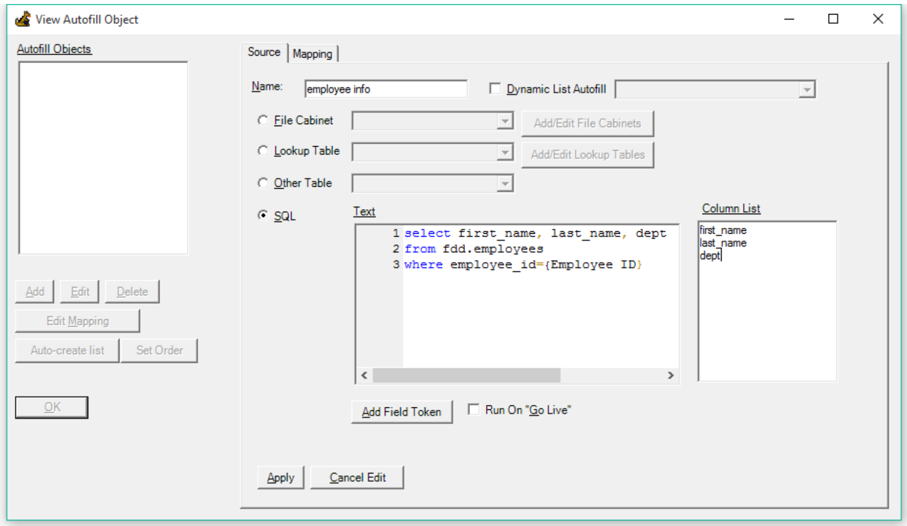
The field is autofilled with a value from the results of a SQL query. You can set up the autofill to fill multiple form fields at once.

Note: If you want to autofill a dynamic list, see Autofill Dynamic List for instructions.

To autofill from a SQL query:

1. Select **Form>Autofill**. The **Autofill** dialog opens.
2. Click **Add**. The **Source** tab displays on the right.
3. Enter the autofill object's **Name**.
4. Select **SQL** as the source.

   Note: This option is disabled if you do not have the **Edit Forms iQ SQL** task permission.
5. Enter the SQL query **Text**. Follow these guidelines when writing the SQL:
   - Only the desired autofill values should be returned, i.e. make sure you have the correct join condition in your SQL.
   - Click **Add Field Token** to include form field tokens in your SQL. Make sure the token is or is not enclosed in single quotes, as is appropriate for the type of data.
   - You can format your text using **TAB** to indent and **SHIFT+TAB** to un-indent.
   - Do not end the query with a semicolon.
   - Using a token from a list field (e.g. STRING-LIST) is not supported. You could write SQL that can retrieve values from a list field (which is an XML column) if necessary (e.g. on Oracle: select (emails).getStringVal() as email_list from fdd.employees where employee_id='{employee_id}'
6. In the **Columns List**, enter the name of the columns from which you want to autofill. Each column should be on a separate line.
7. Optionally choose to make the autofill **Run on "Go Live"**, which will make the autofill run when a user "goes live" on a submitted form.



Name: employee info

□ Dynamic List Autifill

○ File Cabinet

○ Lookup Table

○ Other Table

● SQL

Text

```
select first_name, last_name, dept
from fdd.employees
where employee_id={Employee ID}
```

Column List

```
first_name
last_name
dept
```

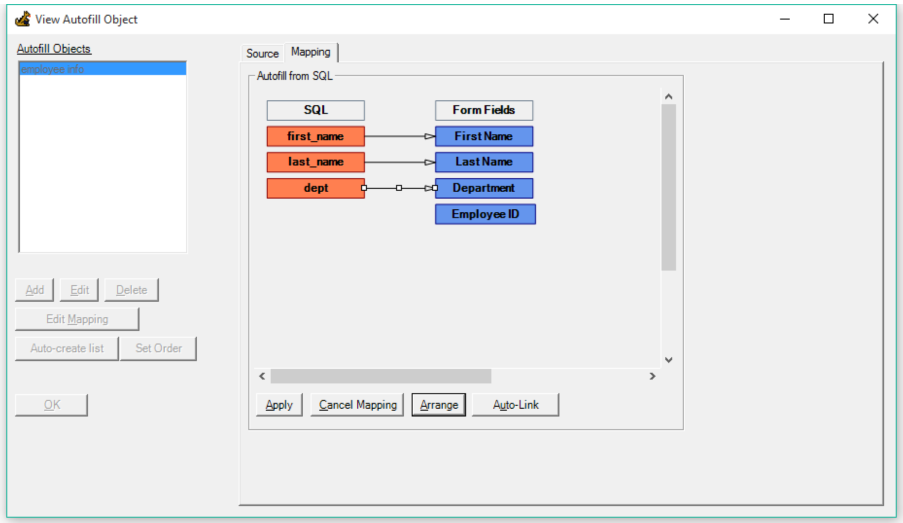
□ Run On "Go Live"

8. Go to the **Mapping** tab. The columns from your source SQL display on the left and the form fields display on the right.

9. Set the form fields that will be autofilled. Click on a source column on the left and drag to a form field on the right. An arrow is drawn from left to right, indicating that the value from that source column will be filled into that form field. Map as many fills as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.

   Note: No join condition needs to be set when mapping, as that was done in the SQL query itself on the **Source** tab.



Autofill Objects: employee info

Autofill from SQL

SQL	Form Fields
first_name	First Name
last_name	Last Name
dept	Department
	Employee ID

10. Click **Apply**. The autofill's mapping is saved.

Note: Optionally set the order of SQL autofill objects, defining the order in which they run, by clicking **Set Order**.

Autofill Dynamic List

When autofilling a dynamic list, you have a couple options:

- Autofill one column based on the value in another column in the dynamic list.
- Autofill many rows into the dynamic list based on a value in another field on the form.

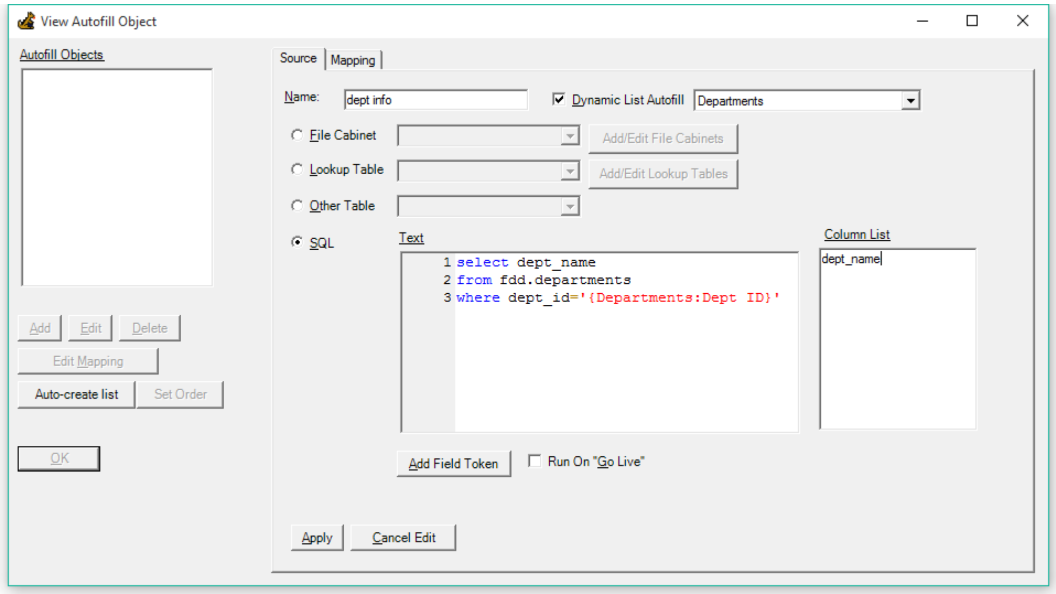
  Tip: This option can be used to create a report on data in your system, like documents in a file cabinet. When creating a report, you probably want to make the fields read-only or labels so they aren't editable and disable other editing options (e.g. **Disable Add/Remove Row**).

Autofill a Column Based on Another Column

To autofill one column based on the value in another column in the dynamic list:

1. Select **Form>Autofill**. The **Autofill** dialog opens.
2. Click **Add**. The **Source** tab displays on the right.
3. Enter the autofill object's **Name**.
4. Turn on **Dynamic List Autofill** and select the dynamic list.
5. Select **SQL** as the source.

   Note: This option is disabled if you do not have the **Edit Forms iQ SQL** task permission.
6. Enter the SQL query **Text**. See Autofill from SQL for more information.
7. Fill out the **Columns List** and optionally turn on **Run on "Go Live"**. See Autofill from SQL for more information.



Name: dept info

☑ Dynamic List Autofill: Departments

○ File Cabinet
○ Lookup Table
○ Other Table
● SQL

Text

```
select dept_name
from fdd.departments
where dept_id='{Departments:Dept ID}'
```

Column List: dept_name

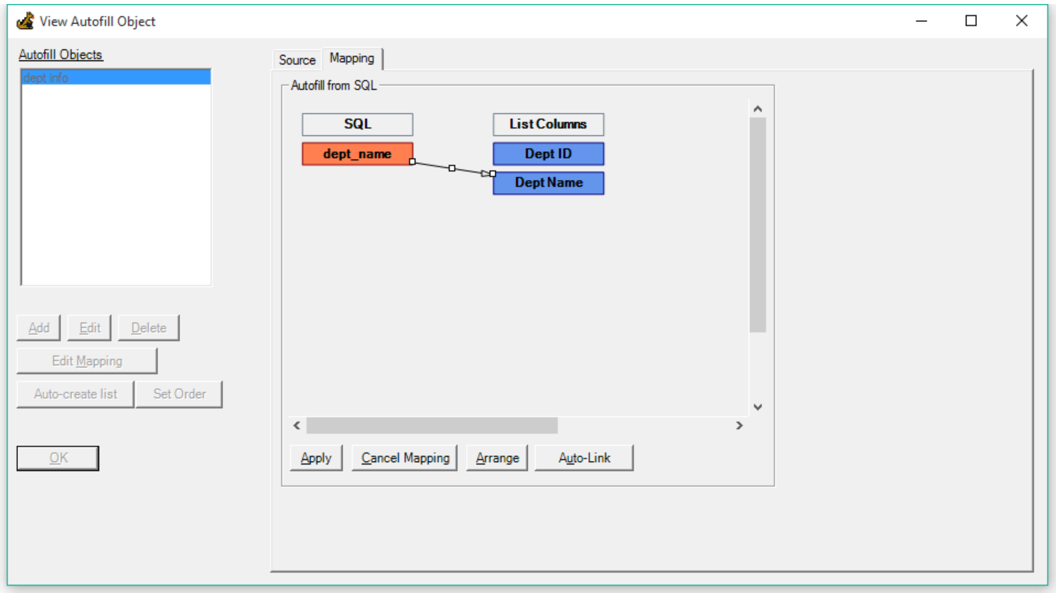
□ Run On "Go Live"

8. Go to the **Mapping** tab. The columns from your source SQL display on the left and the dynamic list columns display on the right.

9. Set the dynamic list columns that will be autofilled. Click on a source column on the left and drag to a dynamic list column on the right. An arrow is drawn from left to right, indicating that the value from that source column will be filled into that dynamic list column. Map as many fills as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.

   Note: No join condition needs to be set when mapping, as that was done in the SQL query itself on the **Source** tab.



10. Click **Apply**. The autofill's mapping is saved.

Autofill Many Rows Based on Another Field

Tips:

- You can ask the Designer to automatically create columns in the dynamic list based on the columns in an autofill object's source, such as a lookup table. After selecting the autofill's **Source**, just click **Auto-create List**. See below for more information.

- This option can be used to create a report on data in your system, like documents in a file cabinet. When creating a report, you probably want to make the fields read-only or labels so they aren't editable and disable other editing options (e.g. **Disable Add / Remove Row**).

To autofill many rows into the dynamic list based on a value in another field on the form:

1. Select **Form > Autofill**. The **Autofill** dialog opens.

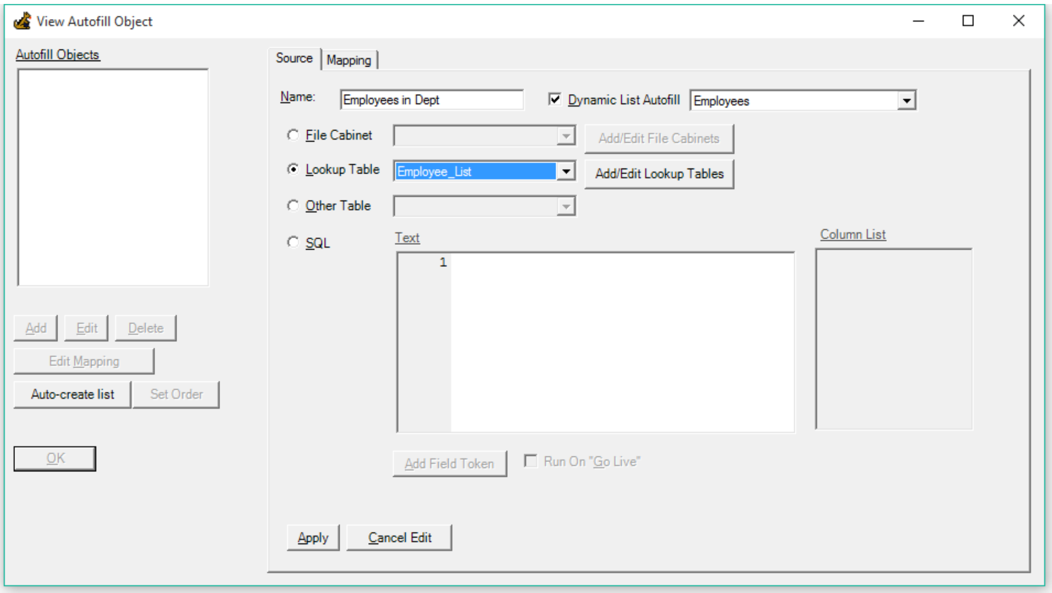
2. Click **Add**. The **Source** tab displays on the right.

3. Enter the autofill object's **Name**.

4. Turn on **Dynamic List Autofill** and select the dynamic list.

5. Select the autofill's source. Options are **File Cabinet**, **Lookup Table**, **Other Table**, and **SQL**. See Autofill from FC, LU, or Table or Autofill from SQL for more information.

6. If using a **SQL** source, fill out the **Columns List** and optionally turn on **Run on "Go Live"**. See Autofill from SQL for more information.



Name: Employees in Dept

☑ Dynamic List Autofill: Employees

○ File Cabinet

● Lookup Table: Employee_List

○ Other Table

○ SQL

□ Run On "Go Live"

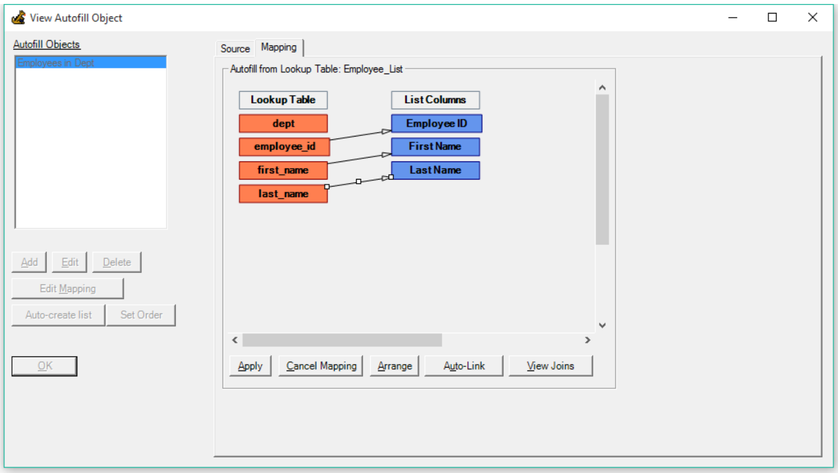
7. Optionally ask the Designer to automatically create columns in the dynamic list based on the columns in an autofill object's source, such as a lookup table. After selecting the autofill's **Source**, just click **Auto-create List**. Columns are added to the dynamic list to match the columns in the autofill source, and the autofill mapping is displayed.

   Note: Any columns that were already in the dynamic list, before you clicked Auto-create list, are removed.

8. Go to the **Mapping** tab. The columns from your source display on the left and the dynamic list columns display on the right.

9. Set the dynamic list columns that will be autofilled. Click on a source column on the left and drag to a dynamic list column on the right. An arrow is drawn from left to right, indicating that the value from that source column will be filled into that dynamic list column. Map as many fills as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.

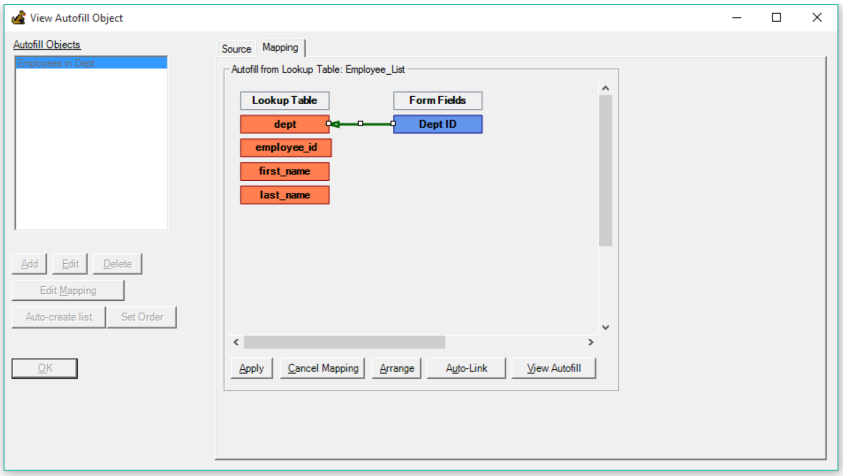


10. Optionally click **View Joins** to set the join condition that will determine which values are grabbed from the source's columns; if none is set then the dynamic list will be autofilled with all rows from the source.

    Click on a form field on the right and drag to a source column on the left. A green arrow is drawn from right to left, indicating that the value in that form field will be used to retrieve the specific, desired values that need to be autofilled into the other form fields. Map as many joins as needed.

    To delete a mapping, select the desired arrow and hit the **DELETE** key.

    Note: When using a **SQL** query as your **Source**, no join condition needs to be set when mapping, as that was done in the SQL query itself.



11. Click **Apply**. The autofill's mapping is saved.

Data Storage

Data Storage

Store form field data to the desired location.

Tip: Generate form fields from columns in an "other" table to which you want to store using **Form>Add Fields From Table**. See Add Form Fields From Table for more information.

TYPE OF DATA STORAGE	HOW IS IT CONFIGURED?
Base File Cabinet The form field data is stored in a file cabinet field. When you submit a form, it is indexed as a document into the FDD file cabinet you selected on creation of the form. The form field data is used as the index values, or metadata, for the document and users can search on this information to find the form. **Notes:** • At least one field on the form *must* store to a file cabinet field. • Dynamic list columns cannot store data in a file cabinet. They must be stored in the Document or another Table.	When you generate a default form on creation, a form field is created for each file cabinet field and stores to each file cabinet field. Modify the **Base File Cabinet** storage object's mapping in the Storage Editor.
Document The form field data is stored in the form document that is created upon submit. Data stored in the document is not searchable.	By default, all new fields you add to the form are stored to the document. The **Document** storage object cannot be modified. Instead, data is stored here automatically if it is not stored anywhere else (i.e. a file cabinet or another table). You can view which form fields store to the **Document** in the Storage Editor.
Table The form field data is stored in another table's column.	Add a storage object in the Storage Editor.

Store Data in File Cabinet

The form field data is stored in a file cabinet field. When you generate a default form on creation, a form field is created for each file cabinet field and stores to each file cabinet field. When you submit a form, it is indexed as a document into the FDD file cabinet you selected on creation of the form. The form field data is used as the index values, or metadata, for the document and users can search on this information to find the form.

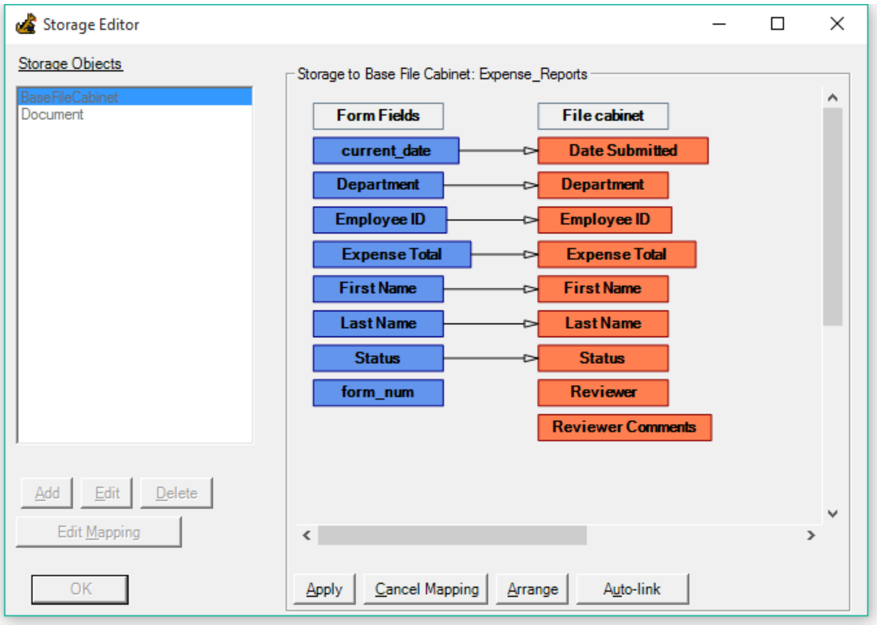
Notes:

- At least one field on the form *must* store to a file cabinet field.
- When a file cabinet field is modified, such as by a user in FDD Client, the form field storing to that file cabinet field will reflect the change.
- Dynamic list columns cannot store data in a file cabinet. They must be stored in the Document or another Table.

To view or modify the **Base File Cabinet** storage object's mapping:

1. Select **Form>Data Storage**. The **Storage Editor** opens.
2. Select the **BaseFileCabinet** storage object. The mapping displays, showing form fields on the left and file cabinet fields to which they store on the right.
3. Click **Edit Mapping** to modify the mapping. The mapping interface on the right is enabled.
4. Set the form fields that will store to the base file cabinet. Click on a form field on the left and drag to a file cabinet field on the right. An arrow is drawn from left to right, indicating that the value from that form field will be stored in that file cabinet field. Map as many fields as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.



Form Fields	File cabinet
current_date	Date Submitted
Department	Department
Employee ID	Employee ID
Expense Total	Expense Total
First Name	First Name
Last Name	Last Name
Status	Status
form_num	Reviewer
	Reviewer Comments

5. Click **Apply**. The storage object's mapping is saved.

Store Data in Document

The form field data is stored in the form document that is created upon submit. By default, all new fields you add to the form are stored to the document. For example, you might want a text area form field that accepts a large value to store to the document.

Notes:

- Data stored in the document is not searchable. When deciding whether to store form field data in the document, consider whether users will need to search on the value to retrieve the submitted form.

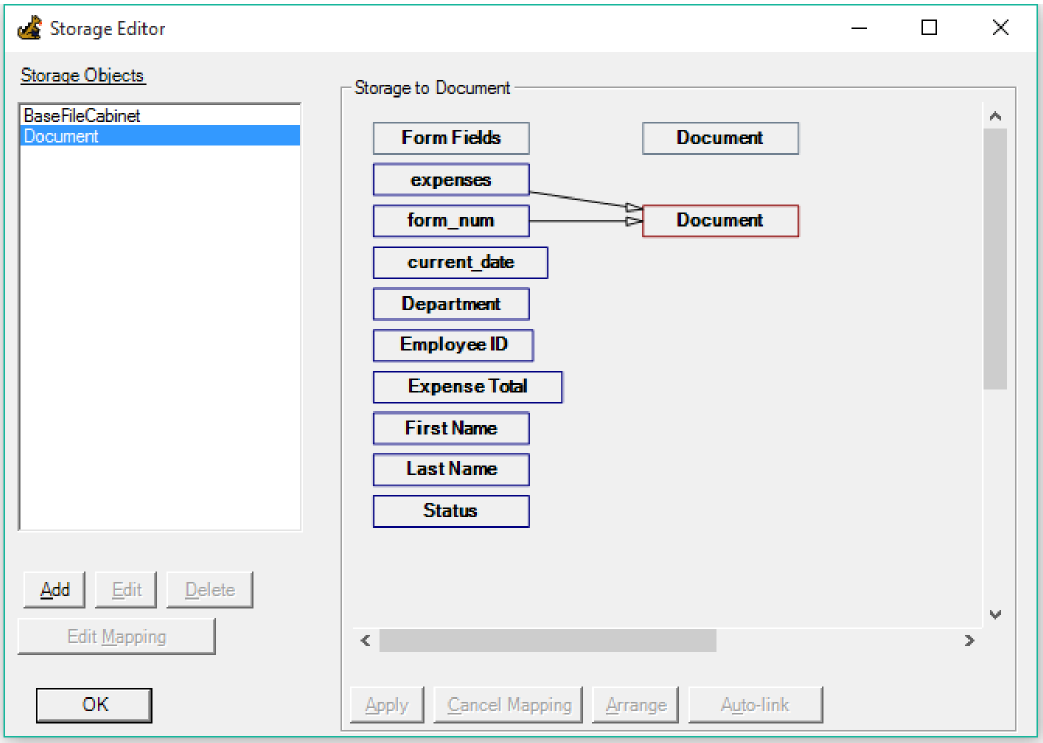
- A field of data type **Signature** cannot be stored to the document.

To view the **Document** storage object's mapping:

1. Select **Form>Data Storage**. The **Storage Editor** opens.

2. Select the **Document** storage object. The mapping displays, showing form fields on the left and the document on the right. An arrow drawn from left to right indicates that the value from that form field will be stored in the document.

   The **Document** storage object cannot be modified. Instead, data is stored here automatically if it is not stored anywhere else (i.e. a file cabinet or another table).



Store Data in Table

The form field data is stored in another table's column.

Tip: Generate form fields from columns in an "other" table to which you want to store using **Form>Add Fields From Table**. See Add Form Fields From Table for more information.

Note: A field of data type **Signature** cannot be stored to a table.

To add a storage object:

1. Select **Form>Data Storage**. The **Storage Editor** opens.

2. Click **Add**. The **Add Storage Object** dialog opens.

3. Enter the storage object's **Name**.

4. Select the **Table** to which you want to store.

   Note: The table must be in your Other Tables List in order to select it here. See Maintain Other Tables List for more information.

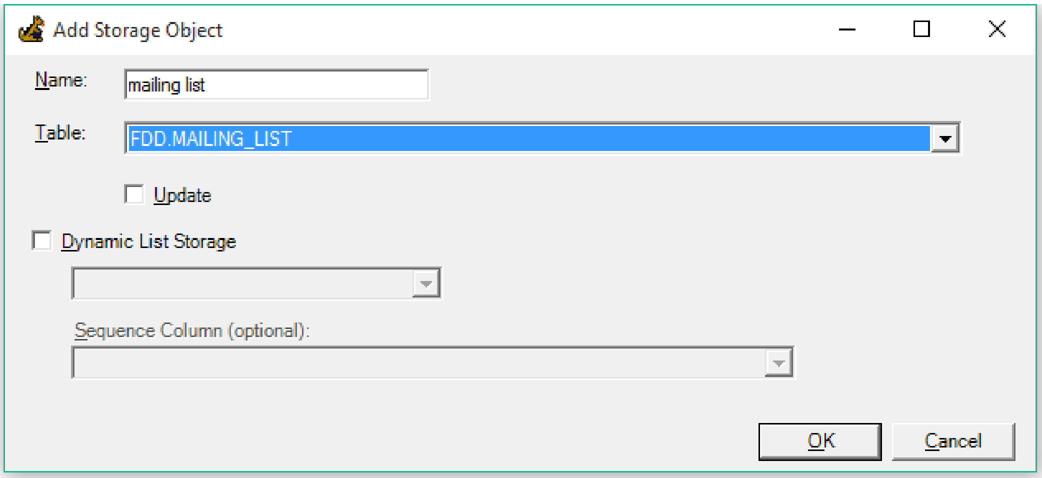
5. Optionally turn on **Update** to update existing data in the table instead of inserting new data. For example, an address change form might be designed to autofill with a vendor's address information (based on an entered value such as Vendor ID), then update the table with modified values when the form is submitted.

   Notes:

   - This option is not available when storing dynamic list data to a table.

   - Later on, make sure you set up your join condition so that the correct rows are updated. See below for more information.

6. Optionally turn on **Dynamic List Storage** to store dynamic list column data to another table. See Store Dynamic List Data in Table for further instructions.



Name: mailing list

Table: FDD.MAILING_LIST

☐ Update

☐ Dynamic List Storage

Sequence Column (optional):

7. Click **OK**. The storage object is added and its mapping displays, showing form and join fields on the left and the table's columns on the right.

8. Set the form fields that will store to the table. Click on a form field on the left and drag to a table column on the right. An arrow is drawn from left to right, indicating that the value from that form field will be stored in that table column. Map as many fields as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.

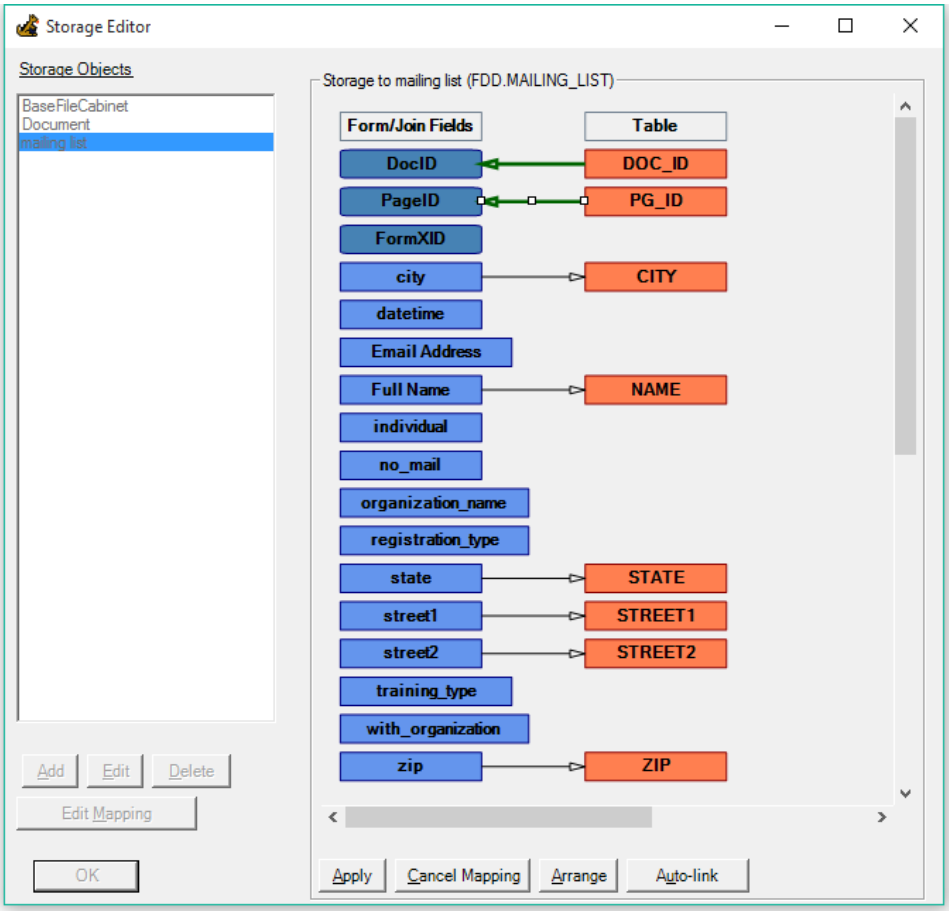
9. Set the join condition that will enable the form to retrieve and display the data stored to the table. Click on a table column on the right and drag to a form or join field on the left. Besides form fields, internal IDs associated with the form are also listed, including **DocID**, **PageID**, and **FormXID**. A green arrow is drawn from right to left, indicating that the value in that table column will be used to retrieve the specific values that were filled out in the form. Map as many joins as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.

   Notes:

   - If you chose the **Update** option previously, the join *cannot* be based on **DocID** or **PageID**. This join condition is used to indicate which rows in the other table need to be updated, in addition to enabling the form to retrieve and display the data stored to the table.

   - See below for notes and guidelines on creating join conditions.



10. Click **Apply**. The storage object's mapping is saved.

Notes on join conditions:

1. If the form allows duplicate indexing, at least one join must be to **PageID** or to a field stored in the document. This is required to ensure that the join is unique to the submitted form (i.e. if duplicate indexing is allowed, **DocID** is not guaranteed to be unique to the submitted form).
2. When a form is submitted and form data is written to a table, the form's join value (form field, **DocID**, **PageID**, or **FormXID**) is written to its mapped "join column" in the table. If a form field is used as a join, the form field value is written both to the field's configured storage location and to the "join column."
3. If you choose to use a form field as a join (instead of **DocID**, **PageID**, or **FormXID**), use the following guidelines:

a. The form field should store its data either in the file cabinet or in the document, not in a table.

b. The mapping needs to be configured so that the relationship between the submitted form and the data in the other table will be unique. So, either the form field should be configured so that its value is unique (e.g. store to a file cabinet field that does not allow duplicate values), or you should configure multiple join conditions for the storage object.

Store Dynamic List Data in Table

The dynamic list column data is stored in another table's column. If you do not store a dynamic list to a table, it will be stored to the document.

Notes:

- Not all columns in the dynamic list have to be stored in the table, although you need a Sequence Column in order to do this. Any dynamic list columns not stored to the table will simply not be stored at all.

- If the **Selective Save** option is selected in the dynamic list properties, only checked rows will be saved when the form is submitted. See Dynamic List for more information.

To add a storage object for a dynamic list:

1. Select **Form>Data Storage**. The **Storage Editor** opens.

2. Click **Add**. The **Add Storage Object** dialog opens.

3. Enter the storage object's **Name**.

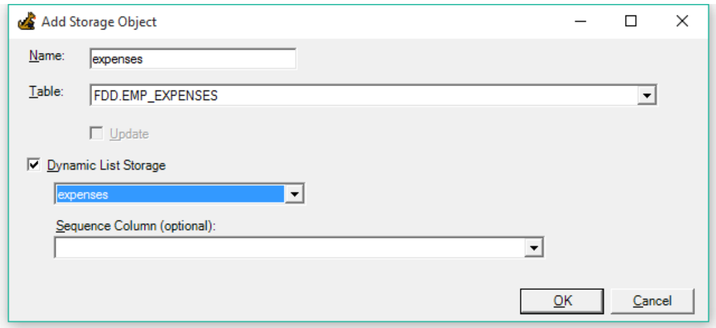
4. Select the **Table** to which you want to store.

   Note: The table must be in your Other Tables List in order to select it here. See Maintain Other Tables List for more information.

5. Turn on **Dynamic List Storage** and select the dynamic list you want to store to this table.

   Note: The **Update** option is not available when storing dynamic list data to a table.

6. Optionally select a **Sequence Column** if you want to ensure that dynamic list's rows are retrieved in the same order as they were when the form was submitted. Select a column to which the sequence number will be stored. Forms iQ will write an incrementing number, starting at 0, to the selected column for each row in the dynamic list.

   Add Storage Object

   Name: expenses

   Table: FDD.EMP_EXPENSES

   □ Update

   ☑ Dynamic List Storage

   expenses

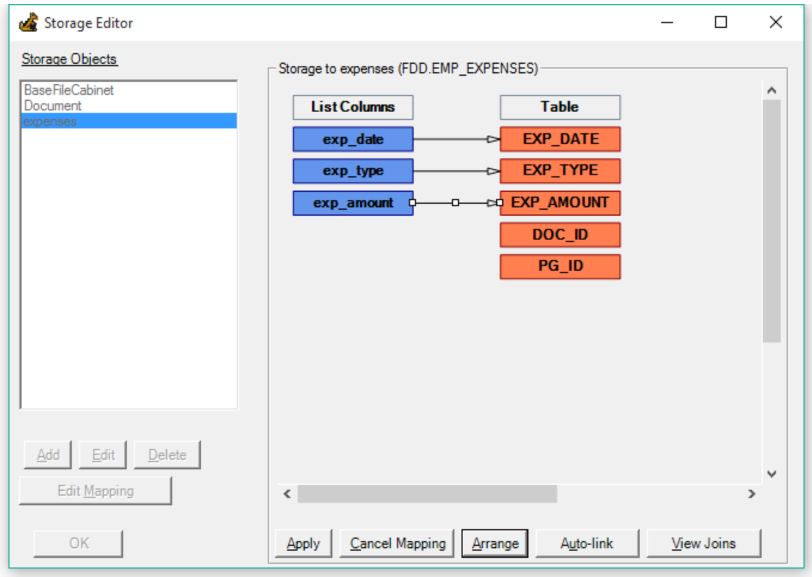
   Sequence Column (optional):

7. Click **OK**. The storage object is added and its mapping displays, showing dynamic list columns on the left and table's columns on the right.

8. Set the dynamic list columns that will store to the table. Click on a dynamic list column on the left and drag to a table column on the right. An arrow is drawn from left to right, indicating that the value from that dynamic list column will be stored in that table column. Map as many columns as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.

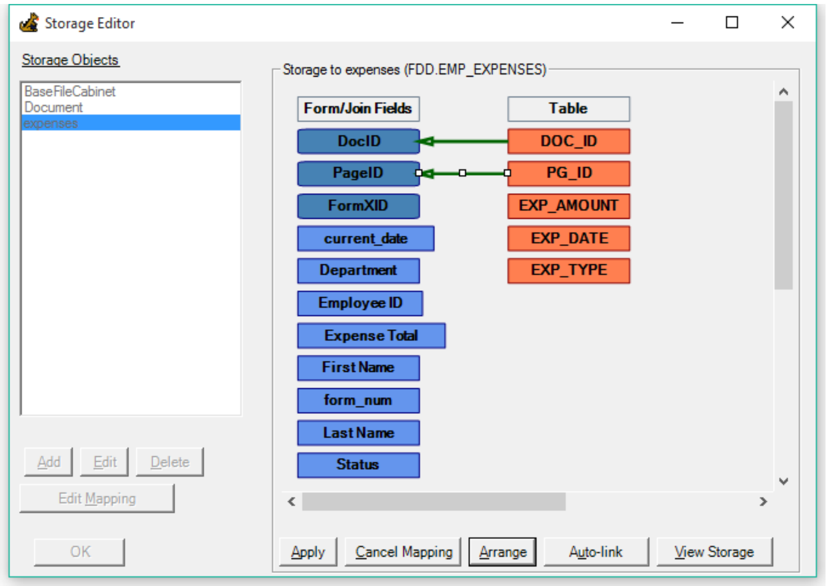
Note: Not all columns in the dynamic list have to be stored in the table, although you need a Sequence Column in order to do this. Any dynamic list column not mapped to a table column will simply not be stored at all.



9. Click **View Joins** to set the join condition that will enable the form to retrieve and display the data stored to the table. Click on a table column on the right and drag to a form or join field on the left. Besides form fields, internal IDs associated with the form are also listed, including **DocID**, **PageID**, and **FormXID**. A green arrow is drawn from right to left, indicating that the value in that table column will be used to retrieve the specific values that were filled out in the dynamic list. Map as many joins as needed.

   To delete a mapping, select the desired arrow and hit the **DELETE** key.

   Note: A dynamic list column cannot be used as a join.



10. Click **Apply**. The storage object's mapping is saved.

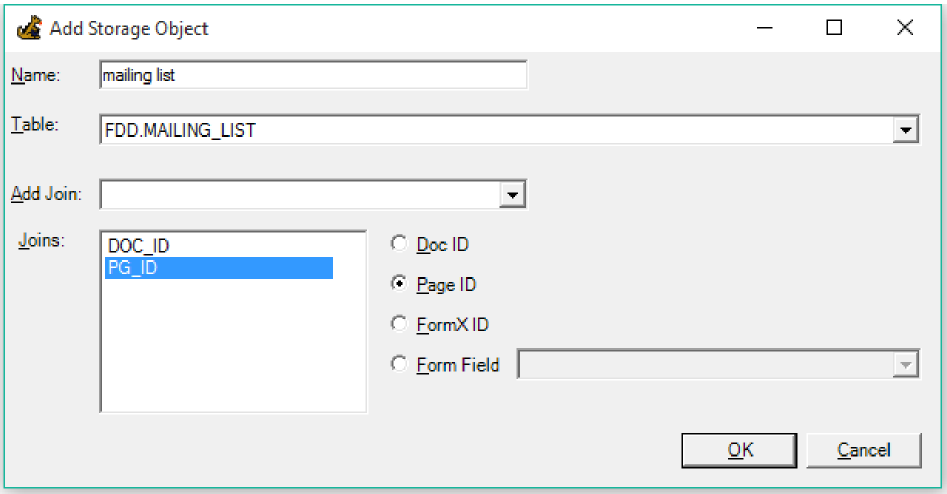
Add Form Fields From Table

Generate form fields from an external table to which you want the new form fields to store.

To add form fields from an external table:

1. Select **Form>Add Fields from Table**. The **Add Storage Object** dialog opens.
2. Enter a **Name** for the storage object.
3. Select the table to use for data storage.

   Note: The table must be in your Other Tables List in order to select it here. See Maintain Other Tables List for more information.
4. Set the join condition that will enable the form to retrieve and display the data stored to the table:
   a. Select a table column from the **Add Join** list.
   b. Select that table column you just added in the **Joins** list.
   c. Choose the form data to which the table column will join. You can join to form fields or internal IDs associated with the form, including **Doc ID**, **Page ID**, and **FormX ID**.
   d. Add as many joins as needed.



Name: mailing list

Table: FDD.MAILING_LIST

Add Join:

Joins: DOC_ID, PG_ID

○ Doc ID
● Page ID
○ FormX ID
○ Form Field

5. Click **OK**. The form fields are generated at the bottom of the form as textboxes and the storage object for the fields is created.

Advanced Features

Validations

Create and Manage Validations

Validate data entered in form fields to make sure the data is correct before the form is submitted. If the validation check passes, the form submits successfully. If the validation check fails, the form does not submit and the user gets an error message.

See the following for instructions:

- Create Validation
- Clone Validation
- Modify Validation
- Delete Validation

Create Validation

To create a validation:

1. Select **Form>Validations**. The **Define Validation Expressions** dialog by opens.
2. Click **Add**. The **Validation Expression** dialog opens.
3. Enter the validation's **Name**.
4. Optionally turn on **In-row validation list** to validate each row in a dynamic list. If the validation check fails for any row in the dynamic list, an error will be returned for each row with a failure. Turning on this option lets you add field tokens for dynamic list columns in your expression or failure message, in addition to the form fields.
5. Enter the **Failure Message** given to the user when the validation check fails upon submitting the form. Optionally **Add a Field Token** to the message.
6. Choose the type of **Expression** and set its properties:
   - Math Expression
   - SQL Expression
   - Regular Expression
7. Click **OK**.   The validation is created.

Clone Validation

To clone a validation:

1. Select **Form>Validations**. The **Define Validation Expressions** dialog by opens.
2. Select the validation you want to clone and click the **Clone** button. A clone of the validation is created with the same name plus a number on the end.

Modify Validation

To modify a validation:

1. Select **Form>Validations**. The **Define Validation Expressions** dialog by opens.
2. Select the desired validation and click **Edit**. The **Validation Expression** dialog opens.
3. Modify the validation as desired. See the following for instructions:
   - Create Validation
   - Math Expression
   - SQL Expression
   - Regular Expression
4. Click **OK**. The validation is modified.

Delete Validation

To delete a validation:

1. Select **Form>Validations**. The **Define Validation Expressions** dialog by opens.
2. Select the validation you want to delete and click **Remove**. The validation is deleted.

Math Validation

A math expression must evaluate to a Boolean value. The form will submit successfully when the condition is true.

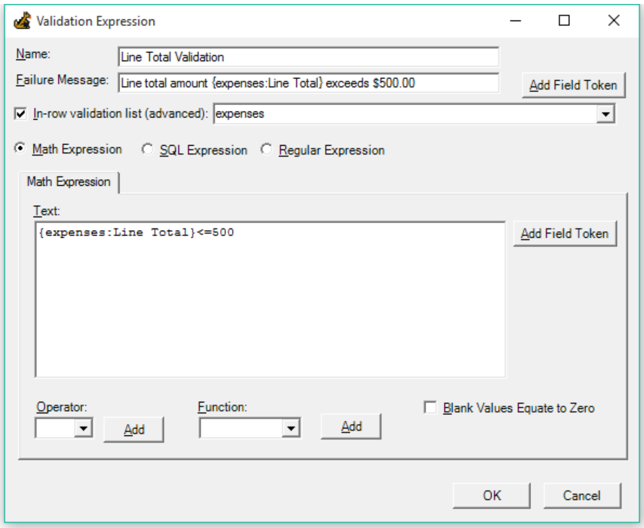
To create a math expression validation:

1. Begin creating a validation. See Create and Manage Validations for more information.
2. Select **Math Expression**. The math expression options display.
3. Enter the expression in the **Text** field. Use the following tools and notes to help you write the expression:

    - Click **Add Field Token** and select a field token. The field token will be replaced with the field value when the form is opened in the browser.
    - Select an **Operator** and click **Add**.
    - Select a **Function** and click **Add**.
    - See Expression Syntax for more information.

4. Optionally turn on **Blank Values Equate to Zero**. If this option is on, and if any of the fields included in the expression contain a blank value, the blank value will equate to zero. If this option is off, and if any of the fields included in the expression contain a blank value, the validation check will not be performed when the form is submitted.

    Note: If you want to require the forms user to enter a value in a form field, it is recommended that you make the field mandatory.



Validation Expression

Name: Line Total Validation

Failure Message: Line total amount {expenses:Line Total} exceeds $500.00

☑ In-row validation list (advanced): expenses

● Math Expression ○ SQL Expression ○ Regular Expression

Math Expression

Text:

```
{expenses:Line Total}<=500
```

Operator:

Function:

☐ Blank Values Equate to Zero

5. Finish creating the validation as usual. See Create and Manage Validations for more information.

Note: If the result of a calculation is out of range it is converted to largest possible integer (either 2147483647 or -2147483647, depending on the sign of the result).

Expression Syntax

OPERATOR/ FUNCTION	COMPARISON/CALCULATION	EXAMPLE
==	Equal to	{field1}=={field2}

<	Less than	{field1}<{field2}
>	Greater than	{field1}>{field2}
+	Adds two numbers.	{field1}+{field2}
-	Subtracts one number from another.	{field1}-{field2}
*	Multiplies two numbers.	{field1}*{field2}
/	Divides one number by another.	{field1}/{field2}
^	Raises a number to the power of an exponent.	{field1}^{field2}
!	Boolean NOT	{field1}!=={field2}
%	Modulus. Returns the remainder after one number is divided by another number.	{field1}%{field2}
<=	Less than or equal to	{field1}<={field2}
>=	Greater than or equal to	{field1}>={field2}
!=	Not equal to	{field1}!={field2}
&&	Boolean AND	({field1}>={field2})&&({field1}<={field3})
||	Boolean OR	({field1}>={field2})||({field1}<={field3})
sum()	Adds all numbers in a set.	sum({field1},{field2})
abs()	Returns the absolute value of a number (the number without a sign).	abs({field1})
rand()	Returns a random number between 0 and 1. To return an integer, multiply by a number.	rand()*{field1}
mod()	Modulus. Returns the remainder after one number is divided by another number.	mod({field1}, {field2})
sqrt()	Returns the square root of a number.	sqrt({field1})

dateDiff()	Calculates the difference, in days or months, between two date fields.	dateDiff(field1, field2, "days")
		dateDiff(field1, field2, "months")

date()	Converts a date field to an integer value for use in a comparison between two or more date fields. For example, if the "startDate" field value must be less than the "endDate" field value, use the following expression:
	date({startDate})<date({endDate})
	If a date literal is to be used, it must be enclosed with double quotes:
	date({startDate})<date("01/01/2005")
		date({field1})<date({field2})

SQL Validation

A SQL query expression validates against the result set returned from a SQL query. Validation can be done against the number of rows returned or against values returned in the result set. The values to check against can contain field tokens. The form will submit successfully when the condition is true.

To create a SQL query expression validation:

1. Begin creating a validation. See Create and Manage Validations for more information.

2. Select **SQL Expression**. The SQL expression options display.

3. Enter the expression in the **Text** field. Use the following tools and notes to help you write the expression:

    - Click **Add Field Token** and select a field token. The field token will be replaced with the field value when the form is opened in the browser. Make sure the token is or is not enclosed in single quotes, as is appropriate for the type of data.
    - You can format your text using **TAB** to indent and **SHIFT+TAB** to un-indent.
    - Do not end the query with a semicolon.

4. Select the constraints to apply to the SQL query results:

    - **Number of Rows:** Specify the number of rows that need to be returned in the result set in order for the validation to pass. To do this, select an operator from the **Rows Returned** drop-down list and enter a number in the text field. See Operator List for more information on operators.

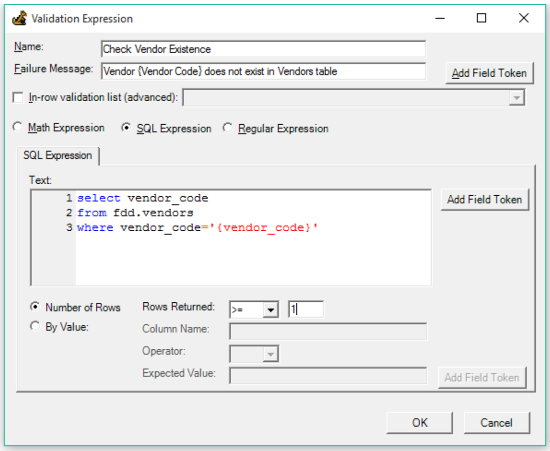
    - **By Value:** Specify the condition to be run against the result set:

        a. Enter the **Column Name**.

        b. Select an **Operator**. See Operator List for more information.

        c. Enter the **Expected value**. You can **Add a Field Token** to the expected value.

        Note: The SQL query should be written so that it returns only one row. If the query returns more than one row, the **By Value** condition will be run against the first row returned.



Name: Check Vendor Existence

Failure Message: Vendor {Vendor Code} does not exist in Vendors table

☐ In-row validation list (advanced):

○ Math Expression ● SQL Expression ○ Regular Expression

Text:

```
select vendor_code
from fdd.vendors
where vendor_code='{vendor_code}'
```

● Number of Rows Rows Returned: >= 1

○ By Value: Column Name: Operator: Expected Value:

5. Finish creating the validation as usual. See Create and Manage Validations for more information.

Operator List

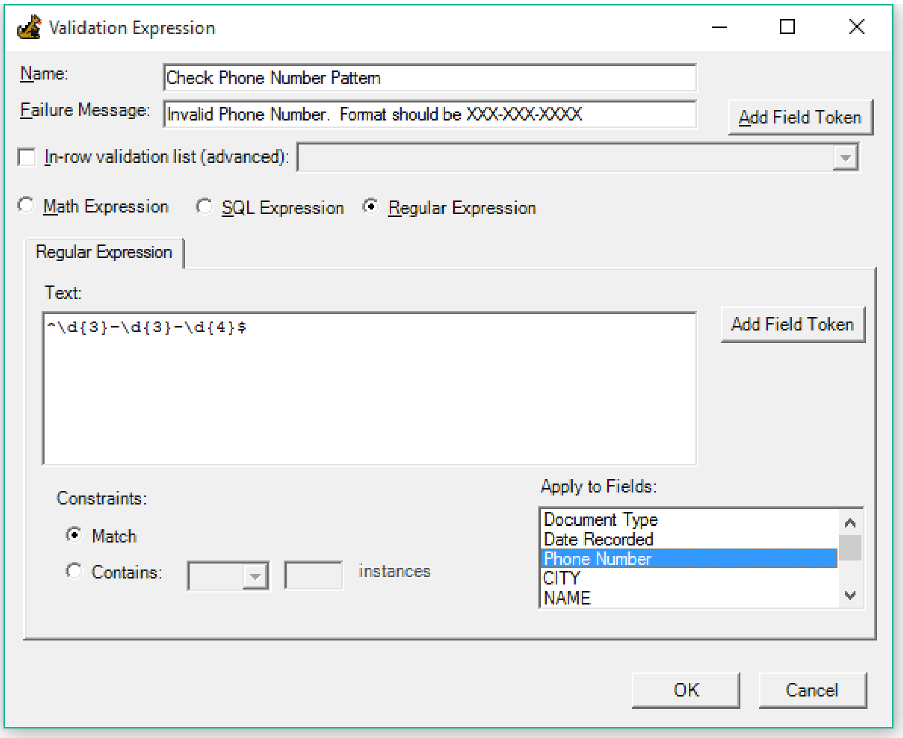
OPERATOR	COMPARISON
<	Less than
>	Greater than
==	Equal to
<=	Less than or equal to
>=	Greater than or equal to
!=	Not equal to

Regular Expression Validation

Regular expressions provide a way to define a string pattern. Since field values frequently follow a pattern or format, a regular expression can be created to test input data for proper format. The form will submit successfully when the condition is true.

To create a regular expression validation:

1. Begin creating a validation. See Create and Manage Validations for more information.
2. Select **Regular Expression**. The regular expression options display.
3. Enter the expression in the **Text** field. Use the following tools and notes to help you write the expression:
   - Click **Add Field Token** and select a field token. The field token will be replaced with the field value when the form is opened in the browser.
   - See Regular Expression Syntax for more information on regular expressions and example patterns.
4. Select the **Constraint** to **Apply to Fields**:
   - **Match:** The value in the selected field must match the regular expression pattern.
   - **Contains:** The value in the selected field must contain the specified number of **instances** of the regular expression pattern.



Name: Check Phone Number Pattern

Failure Message: Invalid Phone Number. Format should be XXX-XXX-XXXX

□ In-row validation list (advanced):

○ Math Expression ○ SQL Expression ● Regular Expression

Text:

```
^\d{3}-\d{3}-\d{4}$
```

Constraints:

● Match

○ Contains: instances

Apply to Fields:

Document Type
Date Recorded
Phone Number
CITY
NAME

5. Finish creating the validation as usual. See Create and Manage Validations for more information.

Regular Expression Syntax

Regular expressions provide a way to define a string pattern. Since field values frequently follow a pattern or format, a regular expression validation can be created to check input data for proper format. See Regular Expression Validation for more information.

For example, check a phone number using the regular expression pattern **^\d{3}-\d{3}-\d{4}$**:

^	The beginning of the pattern.
\d{3}	The first 3 digits.
-	The first hyphen.
\d{3}	The middle 3 digits.
-	The second hyphen.
\d{4}	The last 4 digits.
$	The end of the pattern.

Regular Expression Syntax

[\^$.|?*()	These are the special characters of regular expressions. When these characters exist in the pattern, then they can be escaped with a '\' character.
^	The caret specifies the beginning of a pattern.
$	The dollar sign specifies the end of a pattern.
[ ]	This specifies a set of characters. For example, [abc] specifies 'a' or 'b' or 'c'. [a-zA-Z] specifies the alphabet.
[^]	The specifies anything but the set of characters. For example, [^0-9] specifies anything but the digits 0 through 9.
\d	Specifies a digit.
\w	A word character. The same as [a-zA-Z0-9]
|	The pipe character performs an OR. For example, abc|xyz specifies "abc" or "xyz".
{n}	Specifies a sequence of characters. For example, z{2} specifies "zz". \d{3} specifies 3 digits.
{n,m}	Specifies a sequence of characters of at least n and less than equal to m. For example, \d{2,3} specifies 2 or 3 digits.
?	Specifies that the previous item optional. For example, Bills? Specifies "Bill" or "Bills"
*	Specifies that the previous item will be repeat zero or more times. The period specifies any single character.
+	Specifies that the preceding character(s) will repeat 1 or more times.
()	The parenthesis supports precedence.

Example Regular Expression Patterns

DESCRIPTION	REGULAR EXPRESSION PATTERN
Phone Number	^\d{3}-\d{3}-\d{4}$
Phone Number With Extension	^\d{3}-\d{3}-\d{4} Ext \d{3,4}$
Social Security Number	^\d{3}-\d{2}-\d{4}$
Zip Code	^\d{5}$
IP Address	^\d{3}.\d{3}.\d{3}.\d{3}$
Email Address	^[\w\\.-]+@[\w\.-]+\.[a-zA-Z]+$

Permissions

Login Permissions

By default, your form is public and anyone can access the form. You can restrict access to the form so only FDD users can login and submit the form. A user who goes to the form will be prompted for their FDD user name and password. They enter their login information, and then they can fill out the form as usual. If they enter incorrect login information, they are denied access to the form.

Tip: If you think login permissions are too restrictive, but you don't want the form to be completely open, consider adding a sign-in page to your form. See Sign-In Page for more information on sign-in.

To set login permissions:

1. Select **Permissions>Login**. The **Form Login Permissions** dialog opens.

   By default it is a **Public Form** and no login is required.

2. Select **Login Permission**.

3. Click **Add**. The **Users and Groups** dialog opens.

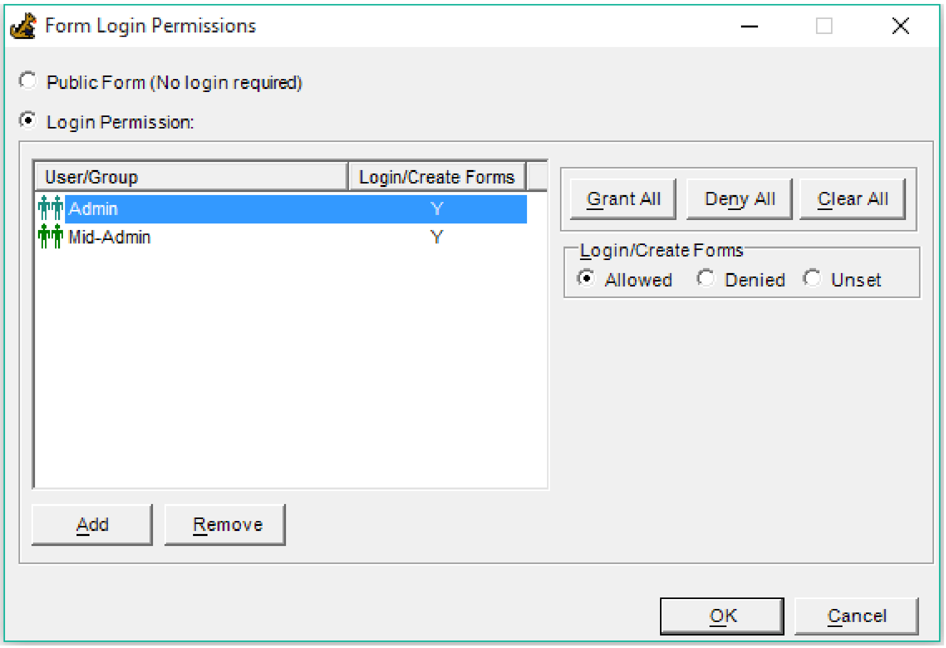
4. Select one or more users and groups to assign login permission. Select multiple users and groups using **CTRL+click** or **SHIFT+click**.

5. Click **OK**. The users and/or groups are added to the **User/Group** list with the **Login/Create Forms** permission set to **Allowed**.

   To remove a user and/or group from the list, just select them and click **Remove**.

6. Optionally change the permission for each user and/or group. Select the desired user/group and change the **Login/Create Forms** permission to **Allowed**, **Denied**, or **Unset**.

   User permissions override group permissions. If a user is granted in one group, they can login. Unset permissions default to denied.



Form Login Permissions

○ Public Form (No login required)

● Login Permission:

User/Group	Login/Create Forms
Admin	Y
Mid-Admin	Y

Login/Create Forms: ● Allowed ○ Denied ○ Unset

7. Click **OK**. Login permissions are set.

Tips:

- Add a **Save** button to a form with login permissions, which lets users partially fill out the main form and save it with their login credentials so they can finish filling it out later. See Add Buttons for more information.

- You can add a logout link to the form for users who login. See Logout Hyperlink for more information.

Go Live Permissions

By default, any user who has access to the submitted form can click the **Go Live** button and update the form then re-submit it. You can restrict access to go live to certain users.

To set go-live permissions:

1. Select **Permissions>Go Live**. The **Go Live Permissions** dialog opens.

   By default everyone is **Allowed** and no one is **Denied**.

2. Choose whether the **Default Access** should be **Allowed** or **Denied**.

   Next you will choose the users and/or groups that are the exceptions to the **Default Access**.

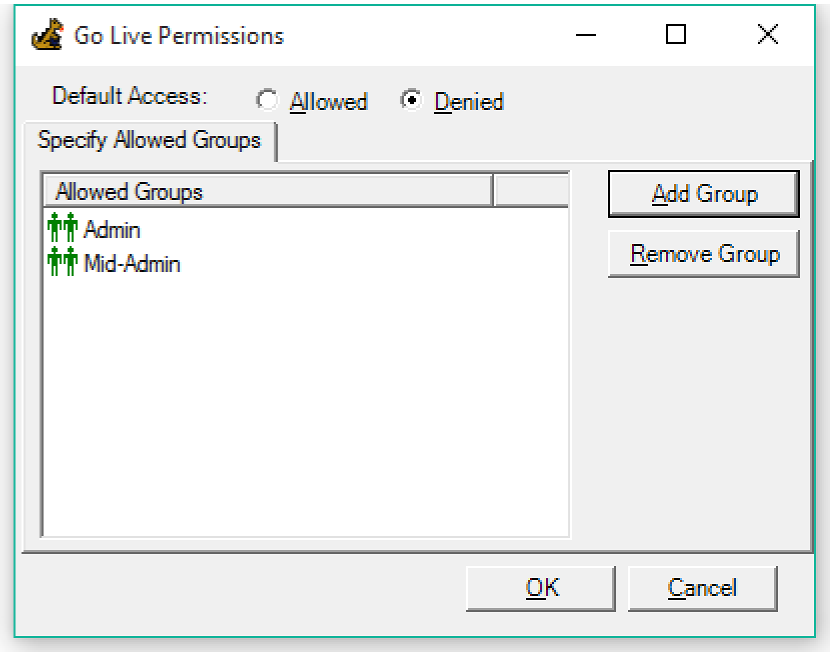
3. Click **Add Group**. The **Users and Groups** dialog opens.

4. Select one or more users and groups to assign. Select multiple users and groups using **CTRL+click** or **SHIFT+click**.

5. Click **OK**. The users and/or groups are added to the **Groups** list. Depending on your **Default Access**, the groups list is either:

   - **Denied Groups:** Most users can go live, but these few users and/or groups are denied go-live permission.

   - **Allowed Groups:** Most users cannot go live, but these few users and/or groups are allowed go-live permission.



Default Access: ○ Allowed ● Denied

Allowed Groups

- Admin
- Mid-Admin

6. Click **OK**. The go-live permissions are set.

Show History Permissions

By default, any user who has access to the submitted form can click the **Show History** link and view the history of changes to the form. You can restrict access to show history to certain users.

To set show-history permissions:

1. Select **Permissions>History**. The **History Permissions** dialog opens.

   By default everyone is **Allowed** and no one is **Denied**.

2. Choose whether the **Default Access** should be **Allowed** or **Denied**.

   Next you will choose the users and/or groups that are the exceptions to the **Default Access**.

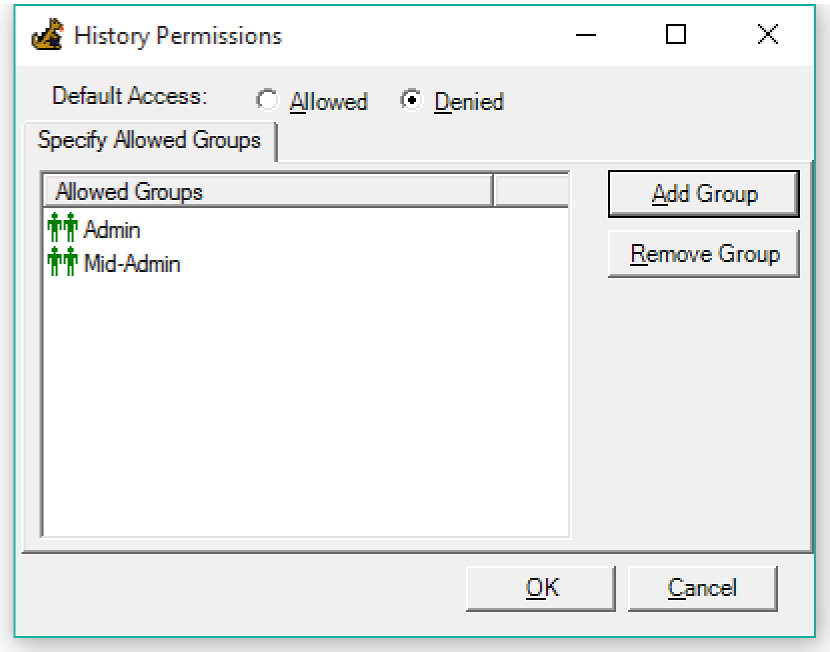
3. Click **Add Group**. The **Users and Groups** dialog opens.

4. Select one or more users and groups to assign. Select multiple users and groups using **CTRL+click** or **SHIFT+click**.

5. Click **OK**. The users and/or groups are added to the **Groups** list. Depending on your **Default Access**, the groups list is either:

   - **Denied Groups:** Most users can view the history, but these few users and/or groups are denied access to the history.

   - **Allowed Groups:** Most users cannot view the history, but these few users and/or groups are allowed access to the history.



History Permissions

Default Access: ○ Allowed ● Denied

Specify Allowed Groups

Allowed Groups
Admin
Mid-Admin

6. Click **OK**. The show-history permissions are set.

Field Set Permissions

By default, all users with access to the form have full read and write permission to all of the form's fields. You can restrict access to some fields so that only certain users and groups can read or write to them.

To create a field set with permissions:

1. Select **Permissions>Field Set**. The **Define Field Set Permissions** dialog opens.

2. Click **Add**. The **Field Set Permission** dialog opens.

3. Enter the **Field Set Name**.

4. Add the fields to which you want to restrict access. Click **Add Field** and **Choose a field**. The field is added to your **Fields** list.

   Remove a field by selecting it in the **Fields** list and clicking **Remove Field**.

   Note: A signature field cannot be added to a field set.

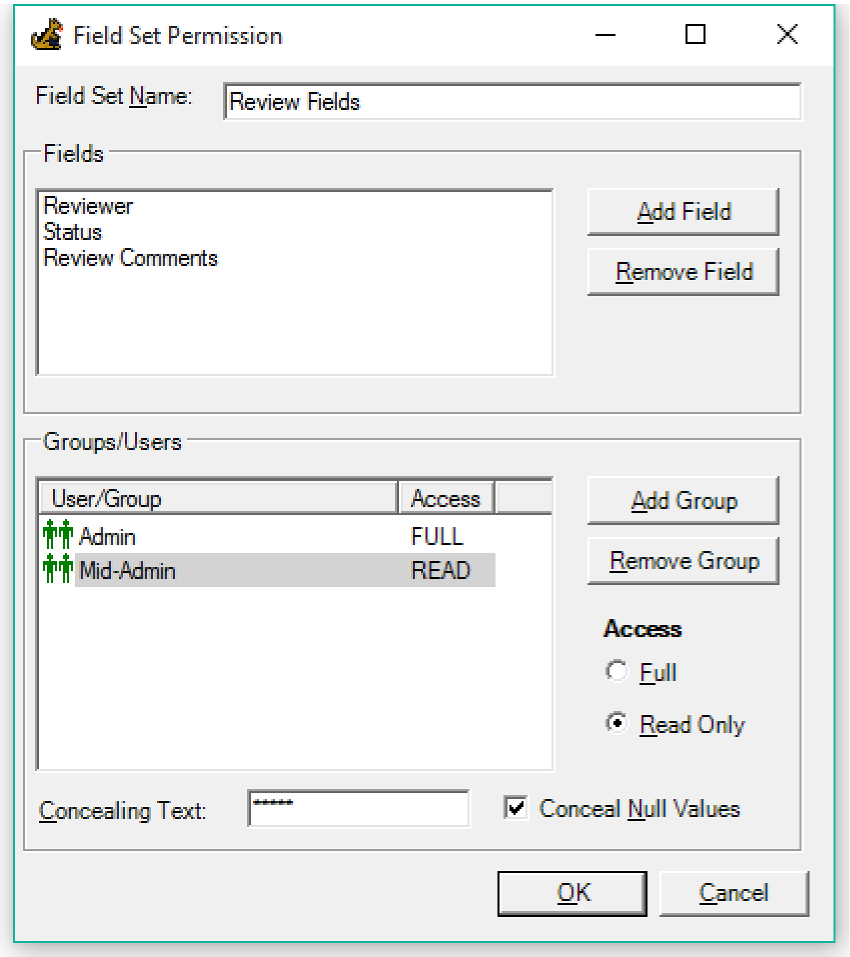
5. Add the users and groups that will have permission to this set of fields. Click **Add Group**. The **Users and Groups** dialog opens.

6. Select one or more users and groups to assign. Select multiple users and groups using **CTRL+click** or **SHIFT+click**.

7. Click **OK**. The users and/or groups are added to the **Groups/Users** list with the **Access** permission set to **Full**.

8. Optionally change the permission for each user and/or group. Select the desired user/group and change the **Access** permission to **Full** read and write or **Read Only**.

9. Optionally enter **Concealing Text** that will display in the fields to users who do not have access to this field set. Optionally turn on **Conceal Null Values** to display **Concealing Text** for blank fields as well.



Field Set Permission

Field Set Name: Review Fields

Fields

Reviewer
Status
Review Comments

Groups/Users

User/Group	Access
Admin	FULL
Mid-Admin	READ

Access

○ Full

● Read Only

Concealing Text: ******

☑ Conceal Null Values

10. Click **OK**. The field set is added.

Note: For public forms, if you want the user who initially fills out and submits the form to have access to the fields in a field set, assign the **formsiq** user to the field set. The **formsiq** user is used as the login for public forms submission.

Calculation Properties

Populate a field with a calculated value.

Note: Calculation Properties are not supported for list fields (e.g. STRING-list).

To set calculation properties:

1. Double-click the field in the form to open the field properties, or double-click the dynamic list column in the **Dynamic List** columns list. The **Field Properties** or **Column Properties** open.

2. Click **Set Calculation Properties**. The **Set Calculate Properties** dialog opens.

3. Enter the calculation in the **Expression** field. Use the following tools and notes to help you write the expression:

    - Click **Add Field Token** to select a form field or dynamic list column token. The field token will be replaced with the field value when the form is opened in the browser.

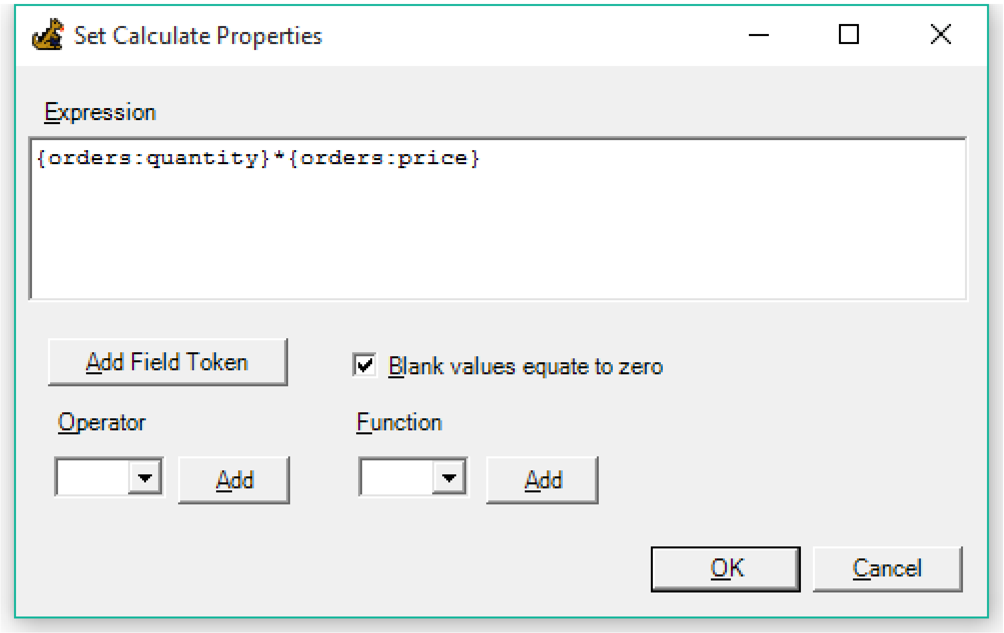
      Note: Dynamic list column tokens can only be used in the calculation properties of another column in the same dynamic list. However, if a dynamic list column has a total, that total can be used in a calculation in a standard form field outside the dynamic list.

    - Select an **Operator** and click **Add**.

    - Select a **Function** and click **Add**.

    - See Expression Syntax for more information.

4. Optionally turn on **Blank values equate to zero** and any blank fields in the expression will equate to zero in this calculation. If you leave this option off and there are blank fields in the expression, the calculation will not be performed.



Expression

```
{orders:quantity}*{orders:price}
```

☑ Blank values equate to zero

5. Click **OK**. The calculation properties are set.

6. Click **OK** in **Field Properties** or **Column Properties** and your changes are saved.

Expression Syntax

OPERATOR/FUNCTION	CALCULATION	EXAMPLE
+	Adds two numbers.	{field1}+{field2}
-	Subtracts one number from another.	{field1}-{field2}
*	Multiplies two numbers.	{field1}*{field2}
/	Divides one number by another.	{field1}/{field2}
^	Raises a number to the power of an exponent.	{field1}^{field2}
sum()	Adds all numbers in a set.	sum({field1},{field2})
abs()	Returns the absolute value of a number (the number without a sign).	abs({field1})
rand()	Returns a random number between 0 and 1. To return an integer, multiply by a number.	rand()*{field1}
mod()	Returns the remainder after one number is divided by another number.	mod({field1}, {field2})
sqrt()	Returns the square root of a number.	sqrt({field1})

dateDiff()	Calculates the difference, in days or months, between two date fields.	dateDiff(field1, field2, "days")
		dateDiff(field1, field2, "months")

Custom JavaScript

Add Custom JavaScript

Customize your form further with your own JavaScript. The **check errors** JavaScript is included by default to get you started on error-checking in JavaScript. There are also some examples of hiding and showing form fields using JavaScript in Hide and Show Examples.

Note: This option requires JavaScript knowledge.

See the following for instructions:

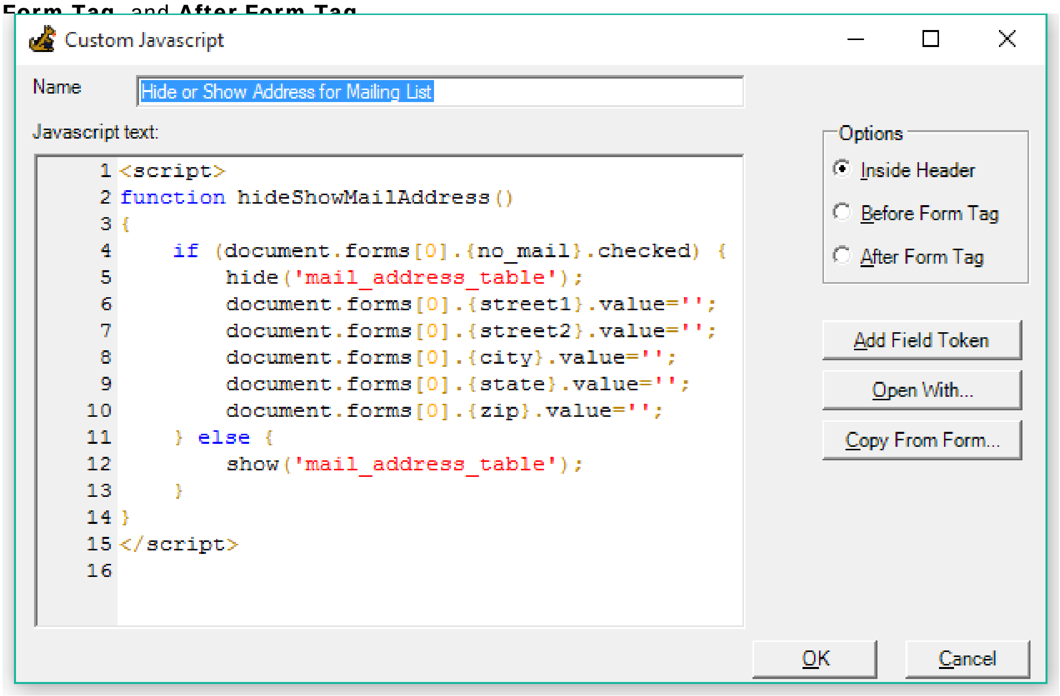
- Add Custom JavaScript
- Assign JavaScript Event to Field

Add Custom JavaScript

To add custom JavaScript:

1. Select **Tools>User-Defined Javascript**. The **User-Defined Javascript** dialog opens.
2. Click **Add**. The **Custom Javascript** dialog opens.
3. Enter the custom JavaScript's **Name**.
4. Enter your **Javascript text**. Use the following tools and notes to help you write your JavaScript:
   - Copy existing JavaScript from another form. Click **Copy From Form**, select the form and click **Open**, select the desired JavaScript and click **OK**. The text from the selected JavaScript is inserted into your new JavaScript.
   - You can include form field values in your logic by clicking **Add Field Token**. The field token will be replaced with the field value when the form is opened in the browser.
   - You can format your text using **TAB** to indent and **SHIFT+TAB** to un-indent.
   - Open the JavaScript in your preferred editor. Click **Open With**, select an editor, and click **OK**. Edit the JavaScript in the other editor, then save and close the editor, and your JavaScript in the Designer is updated. See Maintain Open With Apps for more information on adding your preferred editor.
   - See examples of hiding and showing form fields using JavaScript in Hide and Show Examples.

5. Choose where you want this JavaScript in the form's HTML. Options are **Inside Header**, **Before Form Tag**, and **After Form Tag**.



Name: Hide or Show Address for Mailing List

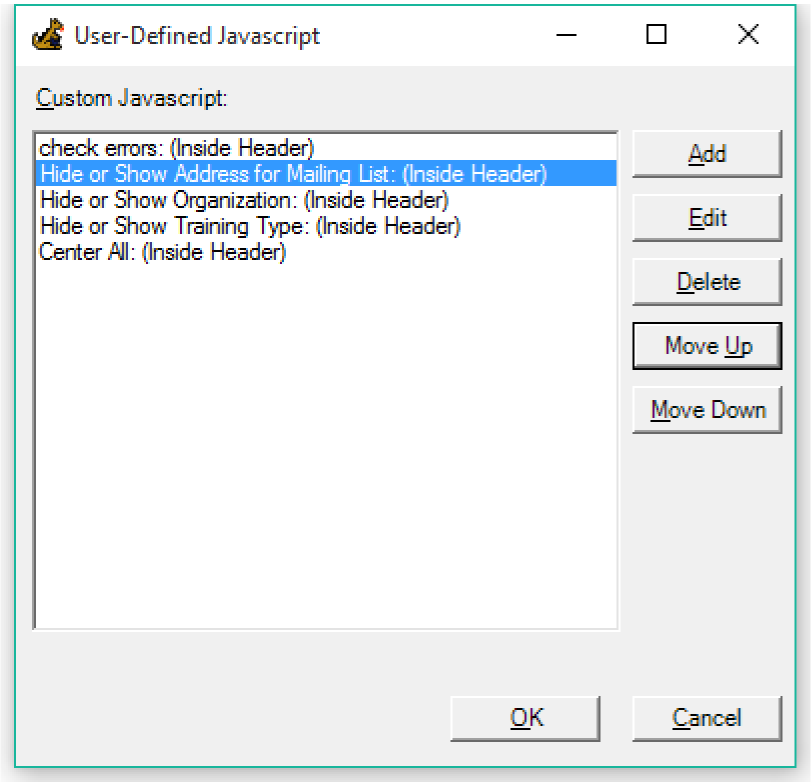
Javascript text:

```
<script>
function hideShowMailAddress()
{
    if (document.forms[0].{no_mail}.checked) {
        hide('mail_address_table');
        document.forms[0].{street1}.value='';
        document.forms[0].{street2}.value='';
        document.forms[0].{city}.value='';
        document.forms[0].{state}.value='';
        document.forms[0].{zip}.value='';
    } else {
        show('mail_address_table');
    }
}
</script>
```

Options
- ● Inside Header
- ○ Before Form Tag
- ○ After Form Tag

6. Click **OK**. The custom JavaScript is added.

7. Continue to add more custom JavaScript is needed. You may choose their order in the form's **HTML** by using the **Move Up** and **Move Down** buttons in the **User-Defined Javascript** dialog.



Custom Javascript:

- check errors: (Inside Header)
- Hide or Show Address for Mailing List: (Inside Header)
- Hide or Show Organization: (Inside Header)
- Hide or Show Training Type: (Inside Header)
- Center All: (Inside Header)

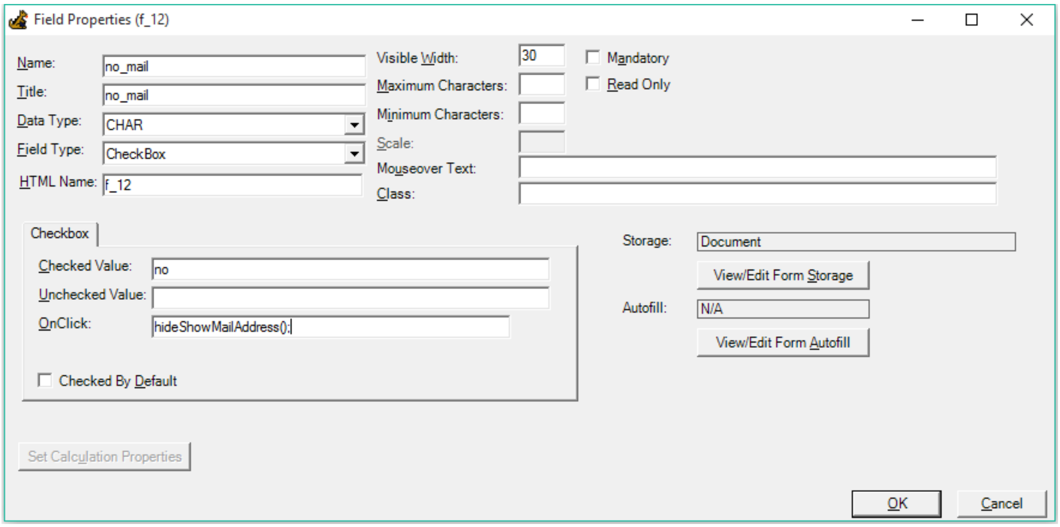
Assign JavaScript Event to Field

To assign a JavaScript OnChange or OnClick event to a form field:

1. Double-click the field to open the field properties.

2. Enter the JavaScript in the **OnChange** or **OnClick** field. For example, you may want to call a custom JavaScript function you added under **Tools>User-Defined Javascript**.

   OnChange can be set for a textbox, select field, text area, password, or lookup. **OnClick** can be set for a check box, radio button, or custom hyperlink.



Field Properties (f_12)

Name: no_mail
Title: no_mail
Data Type: CHAR
Field Type: CheckBox
HTML Name: f_12
Visible Width: 30
Maximum Characters:
Minimum Characters:
Scale:
Mouseover Text:
Class:
□ Mandatory
□ Read Only

Checkbox
Checked Value: no
Unchecked Value:
OnClick: hideShowMailAddress();
□ Checked By Default

Storage: Document
Autofill: N/A

3. Click **OK**. The field properties are saved.

Note on referencing form elements:

When referencing a form field in JavaScript, use the syntax **document.forms[0].element**, inserting a form field token for the element name. For example: **document.forms[0].{Vendor_Name}**. The form field token will be replaced with the field name when the form is opened in the browser.

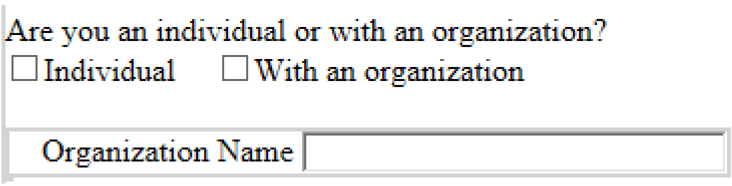
Hide and Show Examples

One way you may want to use Custom JavaScript on a form is to hide and show fields based on what the user enters.

Below are a few examples on how to hide and show fields using JavaScript. The general technique is to put fields into a table that is hidden or shown based on the selection in another field. See Add Custom Javascript and Add Tables for more information.

- Example 1: Show Field When Checkbox Checked
- Example 2: Hide and Clear Fields When Checkbox Checked
- Example 3: Show Field When Another Field is a Specific Value

Example 1: Show Field When Checkbox Checked



Are you an individual or with an organization?
□ Individual □ With an organization

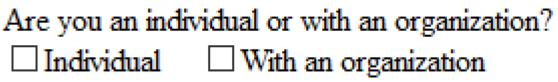
Organization Name

Appearance in Designer

First checkbox's **Name**: individual
Second checkbox's **Name**: with_organization
Table's **Table ID**: organization_table
Table's **Display**: none

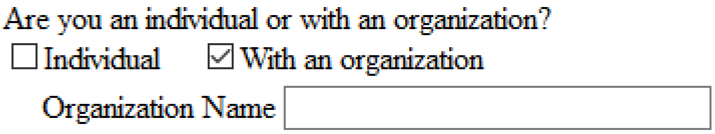
```
<script>
function hideShowOrganization()
{
      if
      (document.forms[0].{with_organization}.checked){
            show('organization_table');
      } else {
            hide('organization_table');
      }
}
</script>
```

Custom JavaScript



Are you an individual or with an organization?
□ Individual □ With an organization

Appearance in browser when form loads

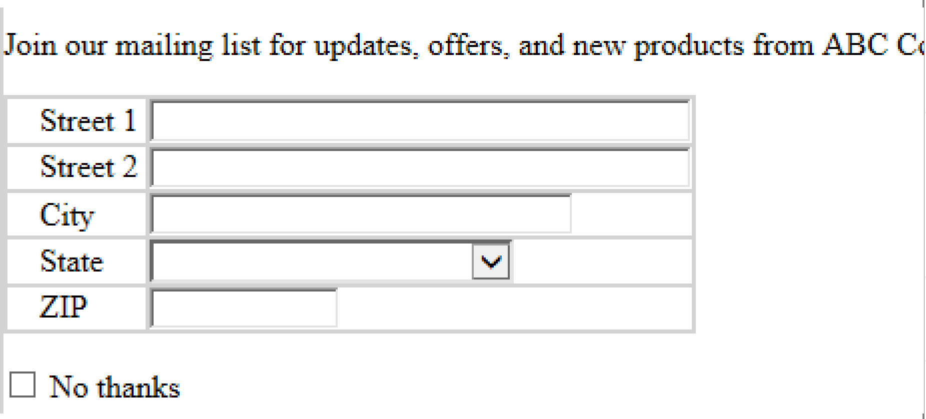


Are you an individual or with an organization?
□ Individual ☑ With an organization

Organization Name

Appearance in browser after checking checkbox

Example 2: Hide and Clear Fields When Checkbox Checked



Join our mailing list for updates, offers, and new products from ABC Co

Street 1	
Street 2	
City	
State	
ZIP	

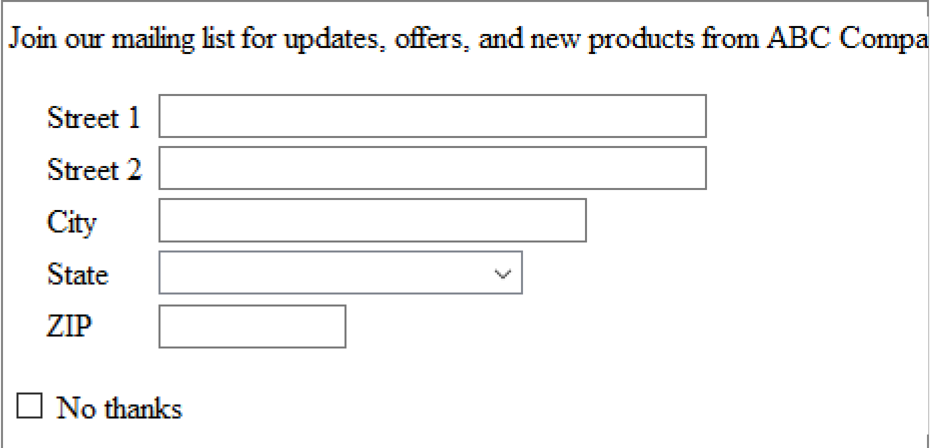
□ No thanks

Appearance in Designer

First textbox's **Name**: street1
Second textbox's **Name**: street2
Third textbox's **Name**: city
Fourth textbox's **Name**: state
Fifth textbox's **Name**: zip
Table's **Table ID**: mail_address_table
Table's **Display**: blank i.e. shown
Checkbox's **Name**: no_mail

```
<script>
function hideShowMailAddress()
{
     if (document.forms[0].{no_mail}.checked) {
          hide('mail_address_table');
          document.forms[0].{street1}.value='';
          document.forms[0].{street2}.value='';
          document.forms[0].{city}.value='';
          document.forms[0].{state}.value='';
          document.forms[0].{zip}.value='';
     } else {
          show('mail_address_table');
     }
}
</script>
```

Custom JavaScript

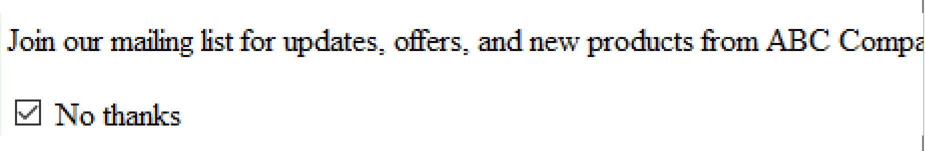


Join our mailing list for updates, offers, and new products from ABC Compa

Street 1	
Street 2	
City	
State	
ZIP	

□ No thanks

Appearance in browser when form loads

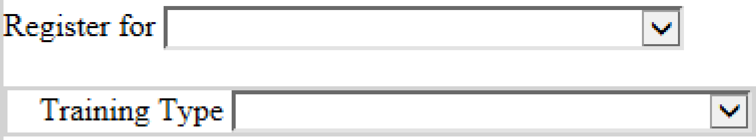
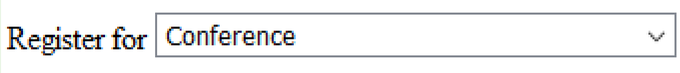
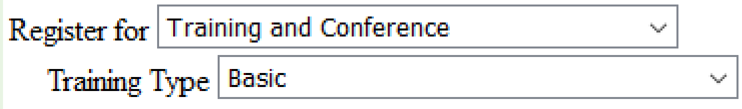


Join our mailing list for updates, offers, and new products from ABC Compa

☑ No thanks

Appearance in browser after checking checkbox

Example 3: Show Field When Another Field is a Specific Value

<table>
<tr><td>
Register for
Training Type</td><td>Appearance in Designer
First select field's <b>Name</b>: registration_type
Second select field's <b>Name</b>: training_type
Table's <b>Table ID</b>: training_type_table
Table's <b>Display</b>: none</td></tr>
<tr><td><pre>
<script>
function hideShowTrainingType()
{
      if(document.forms[0].{registration_type}.value=='Training and Conference') {
            show('training_type_table');
      } else {
            hide('training_type_table');
      }
}
</script>
</pre></td><td>Custom JavaScript</td></tr>
<tr><td>
Register for: Conference</td><td>Appearance in browser when form loads</td></tr>
<tr><td>
Register for: Training and Conference
Training Type: Basic</td><td>Appearance in browser after selecting specific value</td></tr>
</table>

Data Cloning

Let users clone data from an existing form to a new form. This is helpful when submitting forms that often have the same information and only a couple values change. A user can view a submitted form and click a **Clone** button to start a new form with the cloned data.

Configure the fields you want to allow users to clone:

1. Select **Form>Data Cloning Setup**. The **Form Data Cloning** dialog opens.

   By default, all fields are set to **Don't Clone**.

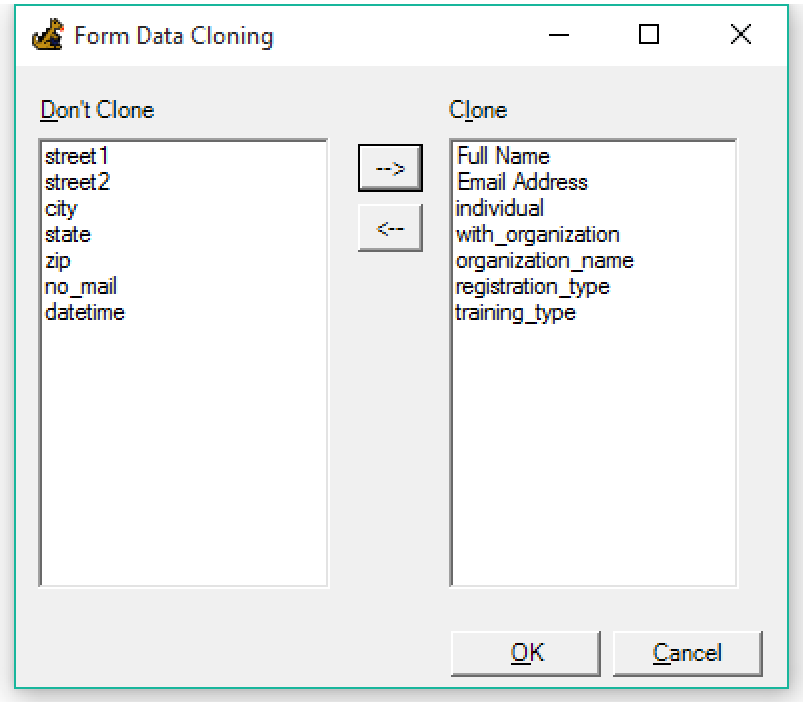
2. Select one or more fields in the **Don't Clone** list and click the right arrow --> button. Select multiple fields using **CTRL+click** or **SHIFT+click**. The fields are moved to the **Clone** list.

   To disallow cloning on a field, just move it back to the **Don't Clone** list by clicking the left arrow button.

   Notes:

   - Numeric fields (i.e. **DECIMAL**, **MONEY**, **SMALLINT**, and **INTEGER**) set to a default keyword value (e.g. **SEQUENCE**) cannot be cloned and therefore do not display in the **Form Data Cloning** window.

   - Signature fields cannot be cloned and therefore do not display in the **Form Data Cloning** window.



Don't Clone	Clone
street1	Full Name
street2	Email Address
city	individual
state	with_organization
zip	organization_name
no_mail	registration_type
datetime	training_type

3. Click **OK**. The selected fields can be cloned by end users. A user can view a submitted form and click a **Clone** button to start a new form with the cloned data.

Manage Forms

Open Form

Open a form to view its design, make changes, or do something else.

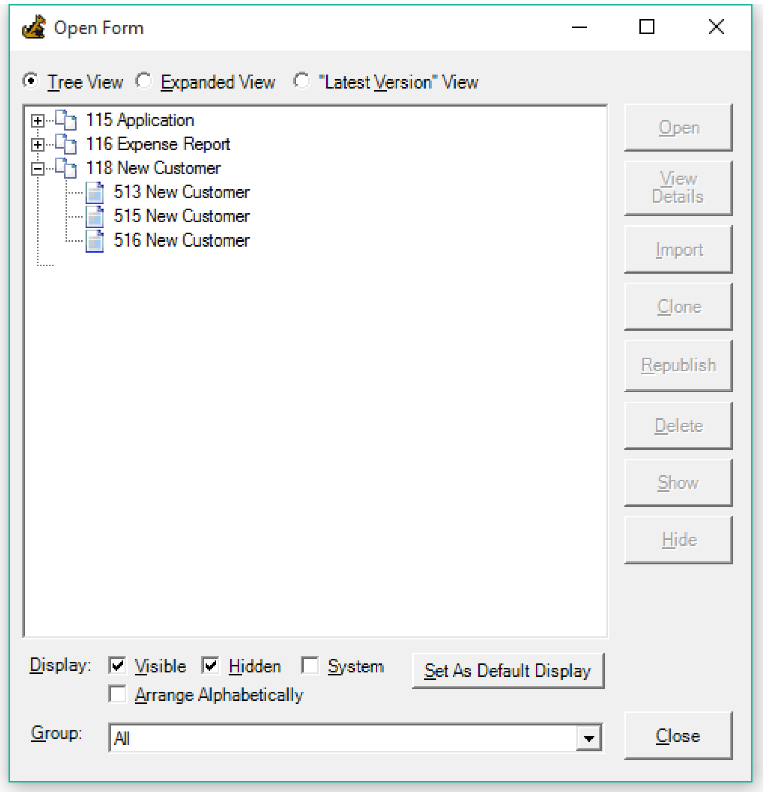
To open an existing form:

1. Select **File**>**Open**. The **Open Form** dialog opens and all form sets you are permitted to see are listed.

   Notes:

   - You can only modify forms built on file cabinets to which you have resource permission. Resource permissions are set in Feith Control Panel. See Feith Control Panel User Guide for more information.
   - If you receive a message that your form references Feith Developer objects that are not available, then you can click **Yes** to open the form anyway and lose the references to those objects, or you can click **No** to cancel. See Set Form Properties for more information.

2. Optionally change the view options to find the form you need. See Open Form View Options for more information.



● Tree View ○ Expanded View ○ "Latest Version" View

- 115 Application
- 116 Expense Report
- 118 New Customer
  - 513 New Customer
  - 515 New Customer
  - 516 New Customer

Display: ☑ Visible ☑ Hidden □ System □ Arrange Alphabetically

Group: All

3. Select the form you want to open and click **Open**. The form is displayed in the Designer.

   After opening the form, you may want to:

   - View or modify the form design
   - View form details
   - View form design report
   - Export the form for migration

Open Form View Options

There are various options to find your form in the list of the Open Form dialog:

- **Tree View:** View form sets and their versions in a "tree" list. The form sets are listed and you can expand any form set to view its versions.
- **Expanded View:** All form versions are listed (and are not sub-divided by form set).
- **Latest Version View:** The latest version of each form set is listed.
- **Visible:** Display forms marked as visible that end users can access.
- **Hidden:** Display forms marked as hidden that end users cannot access.
- **System:** Display system forms.
- **Arrange Alphabetically:** Order forms alphabetically by name instead of ordering by internal ID.
- **Set As Default Display:** Click to remember the currently-selected view options as the desired defaults.
- **Group:** Display forms in the selected group. Select **All** to display all forms from all groups. See Form Groups for more information.

See Form Sets and Versions for more information on form sets and form versions.

Note: You can only edit forms built on file cabinets to which you have resource permission. Resource permissions are set in Feith Control Panel. See Feith Control Panel User Guide for more information.

Modify Form

Modify an existing form design, improving its functionality and adapting it to the changing requirements of your business process.

To modify a form:

1. Open the form you want to modify. See Open Form for more information.
2. Modify the form as needed. You may want to:
   - Add form fields
   - Format the form, add tables, and add buttons
   - Modify the form's HTML directly in the Designer or your preferred editor
   - Add another form page or a Sign-in page
   - Autofill fields with information from your data
   - Store fields' data to various locations
   - Validate information entered in fields
   - Choose who can do what on your form with permissions
   - Calculate the value for a field
   - Customize your form further with your own JavaScript
3. Save your changes by selecting **File>Save/Publish**. See Save Form for more information.

Save Form

Save your form and optionally publish it for end users to access. If your form is under heavy development, you can also do a quick save without further prompts.

See the following for more information:

- Save and Publish
- Quick Save

Save and Publish

To save, and optionally publish, a form:

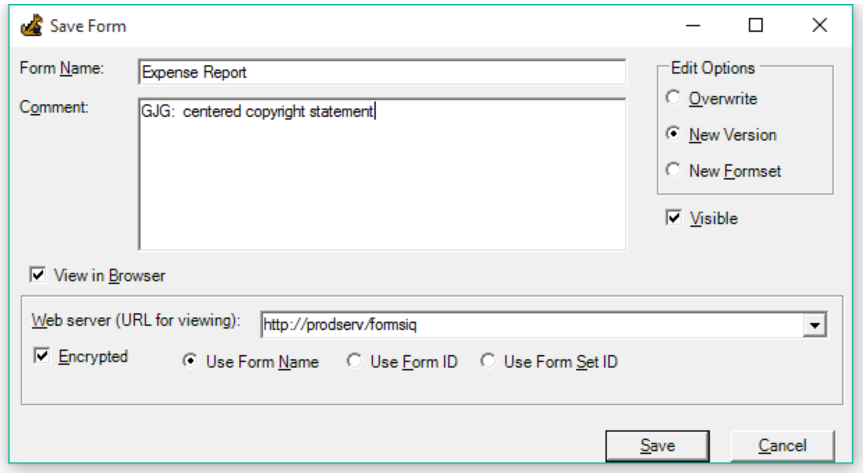
1. Select **File>Save/Publish**. The **Save Form** dialog opens.
2. Optionally change the **Form Name**.
3. Enter a comment for the form version you are saving.

   Tip: We recommend entering a comment listing the changes you made, especially for a form in production, and perhaps along with your initials.
4. If you are saving changes to a form that already existed, choose from the **Edit Options:**
   - **Overwrite:** Overwrite the existing form version you modified.

     Note: If documents or drafts are saved with the current version of the form, the overwrite option is not available.
   - **New Version:** Create a new form version in the form set.
   - **New Formset:** Create an entirely new, separate form set.
   - See Forms Sets and Versions for more information.
5. Optionally check on **Visible** to make this form version available to end users. If you leave Visible unchecked, this form version will not be available to end users. See Hide or Show Form for more information.
6. Optionally check on **View in Browser** to view the form in the browser after saving. Choose the following view options:
   - **Web server:** Select the Forms iQ Server you want to use to view the form. These server entries are created in Feith Control Panel's **Servers** module. See Feith Control Panel User Guide for more information.
   - **Encrypted:** Optionally encrypt the URL instead of seeing it in plain text.

- Choose how you want to identify the form in the URL:
    - **Use Form Name:** The name of the form set. The highest, visible form version in the form set will be viewed.
    - **Use Form ID:** The ID of the specific form version. The specific form version will be viewed.

      Tip: If you unchecked **Visible** to hide the form from users, but you want to view it yourself in the browser, choose **Use Form ID** so you can see the hidden form.
    - **Use Form Set ID:** The ID of the whole form set. The highest, visible form version in the form set will be viewed.
    - See Forms Sets and Versions for more information.



Form Name: Expense Report

Comment: GJG: centered copyright statement

Edit Options
○ Overwrite
● New Version
○ New Formset

☑ Visible

☑ View in Browser

Web server (URL for viewing): http://prodserv/formsiq

☑ Encrypted ● Use Form Name ○ Use Form ID ○ Use Form Set ID

7. Click **Save**. The form is saved. If the **View in Browser** option was selected, the form is opened in a new browser window.

Quick Save

Save the form without any further prompts. We recommend using Quick Save on a form in heavy development. We recommend using the standard **File>Save/Publish** when making small changes to a form in production, so that you have the opportunity to make comments and choose the appropriate save options.

To "quick save" a form:

- Select **File>Quick Save**. The form is saved and you are not prompted to answer any questions.

  Whether Quick Save overwrites an existing version or creates a new version depends on your settings in Global Defaults. If **Default to overwrite** is on, the form version will be overwritten. If **Default to overwrite** is off, a new form version will be created. See Global Defaults for more information.

Submit Form

Fill out and submit a form, then view the submitted form document in the FDD database.

See the following for instructions:

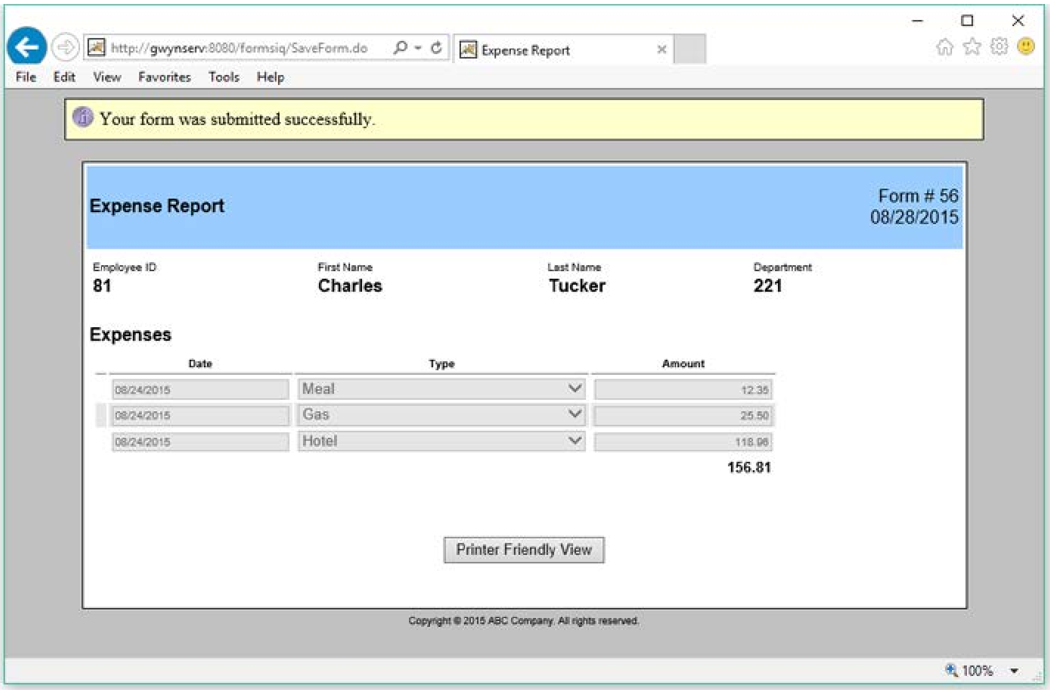
- How to Submit a Form
- What Happens When a Form is Submitted?
- How to View Submitted Forms

How to Submit a Form

To submit a form:

1. View the form in a browser. See View Form in Browser for more information.
2. Enter form field data.
3. Click the **Submit** button. The form is submitted.

Tip: Testing the successful submit of a form is a recommended step of the form design process.



Your form was submitted successfully.

Expense Report

Form # 56
08/28/2015

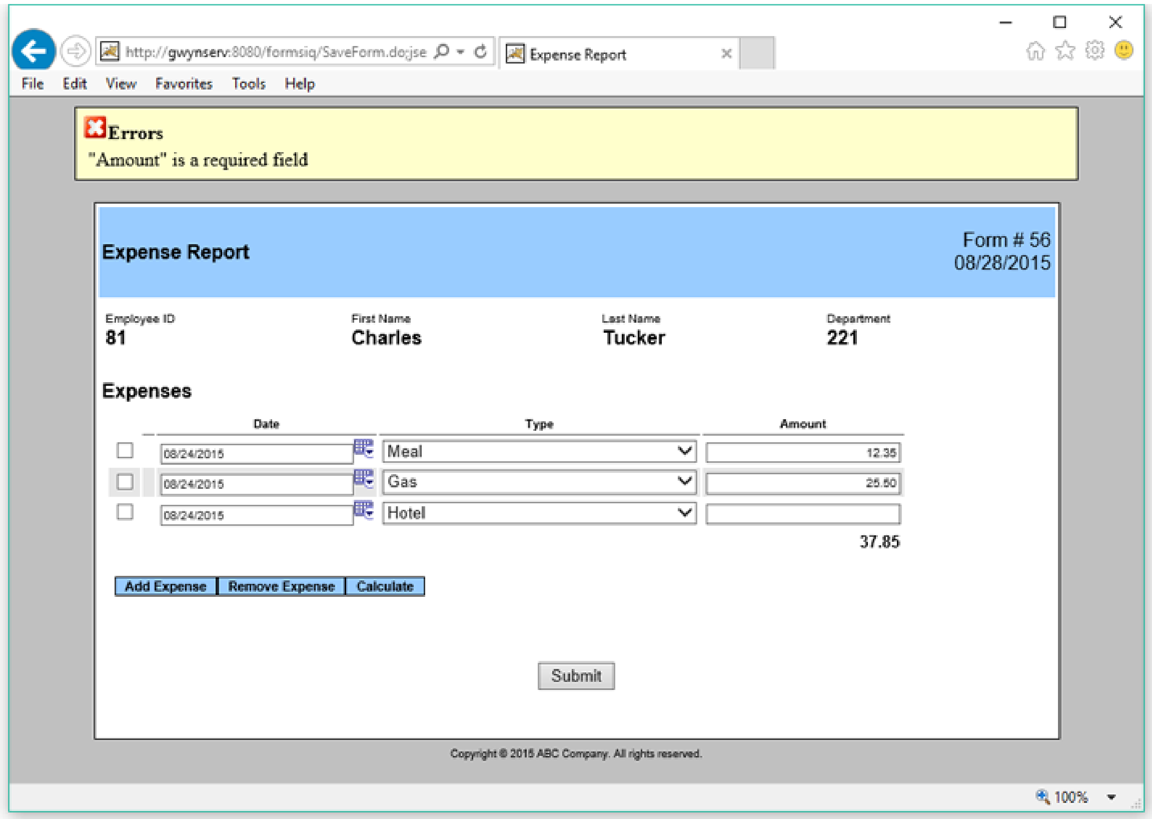
Employee ID	First Name	Last Name	Department
81	Charles	Tucker	221

Expenses

Date	Type	Amount
08/24/2015	Meal	12.35
08/24/2015	Gas	25.50
08/24/2015	Hotel	118.96
		156.81

Copyright © 2015 ABC Company. All rights reserved.

If any form field validation checks fail, such as if a mandatory field is left null, the form will not be submitted and an error message will be shown at the top of the form.



Errors
"Amount" is a required field

Expense Report

Form # 56
08/28/2015

Employee ID	First Name	Last Name	Department
81	Charles	Tucker	221

Expenses

	Date	Type	Amount
□	08/24/2015	Meal	12.35
□	08/24/2015	Gas	25.50
□	08/24/2015	Hotel	
			37.85

Copyright © 2015 ABC Company. All rights reserved.

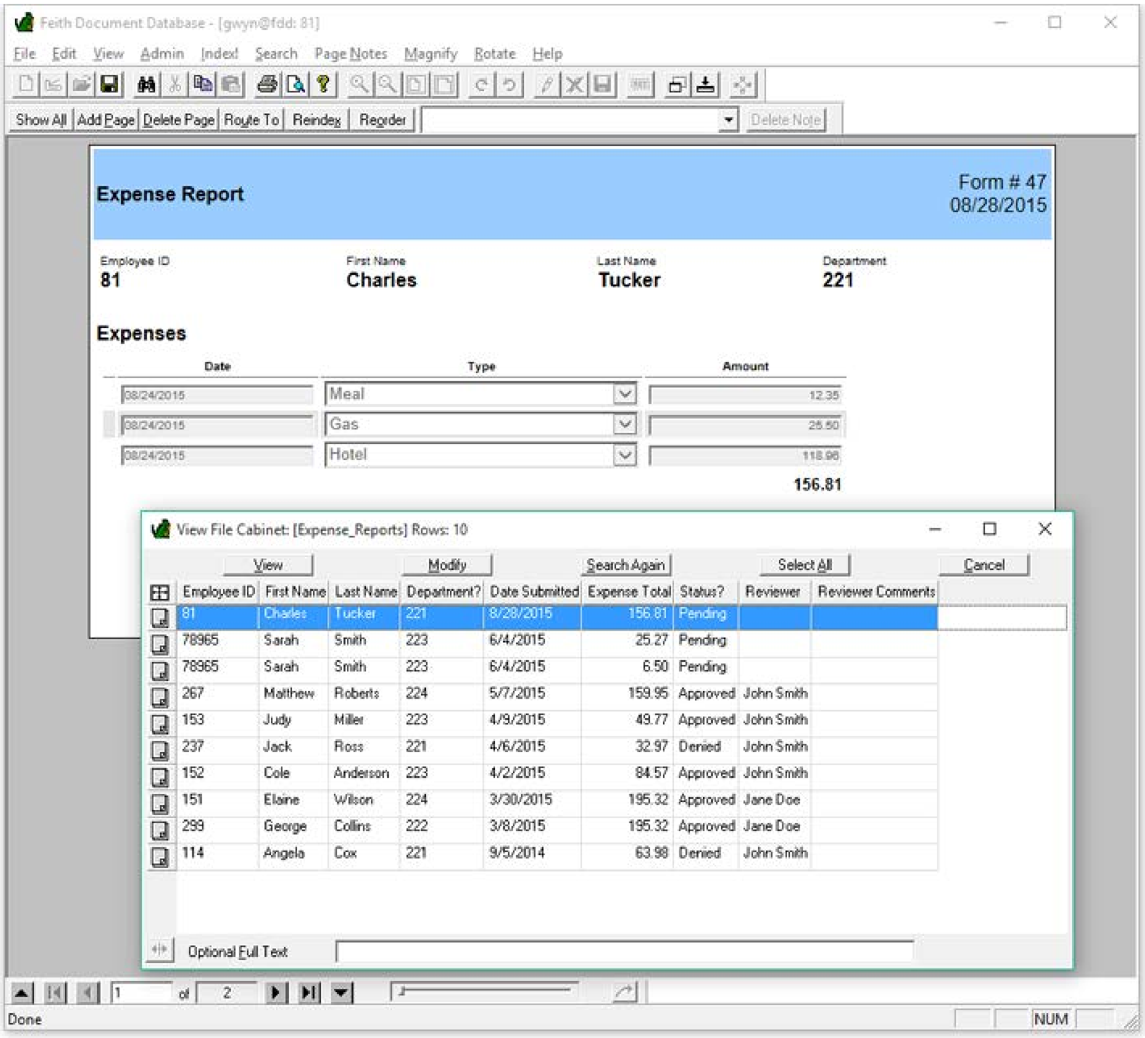
What Happens When a Form is Submitted?

When a form is submitted, it is stored as a new document in FDD. The submitted form is stored in the form's base file cabinet and indexed using the form field values that store to the base file cabinet fields. (Form field data can be stored in other locations. See Data Storage for more information.)

Note: If the form design does not enforce unique indexing, and if the form indexing values match an existing document, the submitted form will be appended as a new page to the existing document.

How to View Submitted Forms

To view the submitted form in FDD Client or in WebFDD, retrieve the document by searching on any of the indexing values and view the document. Refer to the FDD and WebFDD user guides for more information on searching and viewing.



Feith Document Database - [gwyn@fdd: 81]
File Edit View Admin Index! Search Page Notes Magnify Rotate Help
Show All Add Page Delete Page Route To Reindex Reorder Delete Note
View File Cabinet: [Expense_Reports] Rows: 10
View Modify Search Again Select All Cancel
Optional Full Text
of 2
Done
NUM

Expense Report

Form # 47
08/28/2015

Employee ID	First Name	Last Name	Department
81	Charles	Tucker	221

Expenses

Date	Type	Amount
08/24/2015	Meal	12.35
08/24/2015	Gas	25.50
08/24/2015	Hotel	118.96
		156.81

Employee ID	First Name	Last Name	Department?	Date Submitted	Expense Total	Status?	Reviewer	Reviewer Comments
81	Charles	Tucker	221	8/28/2015	156.81	Pending		
78965	Sarah	Smith	223	6/4/2015	25.27	Pending		
78965	Sarah	Smith	223	6/4/2015	6.50	Pending		
267	Matthew	Roberts	224	5/7/2015	159.95	Approved	John Smith	
153	Judy	Miller	223	4/9/2015	49.77	Approved	John Smith	
237	Jack	Ross	221	4/6/2015	32.97	Denied	John Smith	
152	Cole	Anderson	223	4/2/2015	84.57	Approved	John Smith	
151	Elaine	Wilson	224	3/30/2015	195.32	Approved	Jane Doe	
299	George	Collins	222	3/8/2015	195.32	Approved	Jane Doe	
114	Angela	Cox	221	9/5/2014	63.98	Denied	John Smith	

View Form in Browser

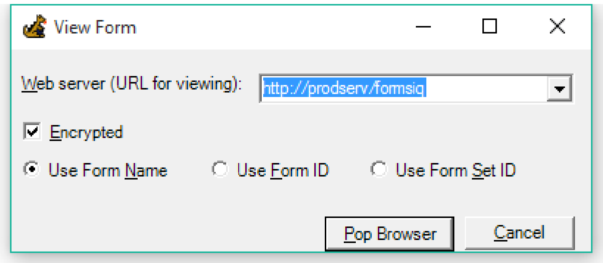
View a form in a browser to fill it out and submit it. You can also see how the form will display to end users.

Note: When viewing a form in a browser, the last saved version of the form is displayed. If you are currently editing the form and have made changes that have not yet been saved, these changes will not be reflected when viewing the form in a browser.

To view a form in a browser:

1. Open the form in the Designer. See Open Form for more information.
2. Select **Form>View in Browser**. The **View Form** dialog opens.
3. Select the Forms iQ **Web server** you want to use to view the form. These server entries are created in Feith Control Panel's **Servers** module. See Feith Control Panel User Guide for more information.
4. Optionally view the form with a URL that is **Encrypted** instead of plain text.
5. Choose how you want to identify the form in the URL:

   - **Use Form Name:** The name of the form set. The highest, visible form version in the form set will be viewed.
   - **Use Form ID:** The ID of the specific form version. The specific form version will be viewed.
   - **Use Form Set ID:** The ID of the whole form set. The highest, visible form version in the form set will be viewed.
   - See Forms Sets and Versions for more information.

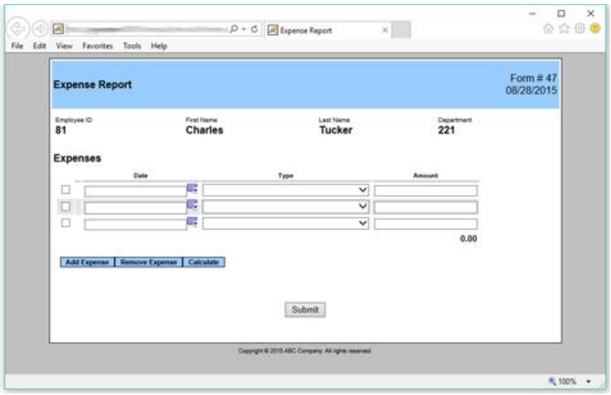


Web server (URL for viewing): http://prodserv/formsiq

☑ Encrypted

● Use Form Name ○ Use Form ID ○ Use Form Set ID

6. Click **Pop Browser**. The form is opened in a new browser window.



Expense Report

Form # 47
08/28/2015

Employee ID 81

First Name Charles

Last Name Tucker

Department 221

Expenses

	Date	Type	Amount
□			
□			
□			
			0.00

Copyright © 2015 ABC Company. All rights reserved.

View Form Details

View a form's details, including who modified it last, what file cabinet its built on, internal IDs, comments, and more. You can view the details on a specific form version or you can view the details for all the form versions in a form set.

See Form Sets and Versions for more information on form sets and form versions.

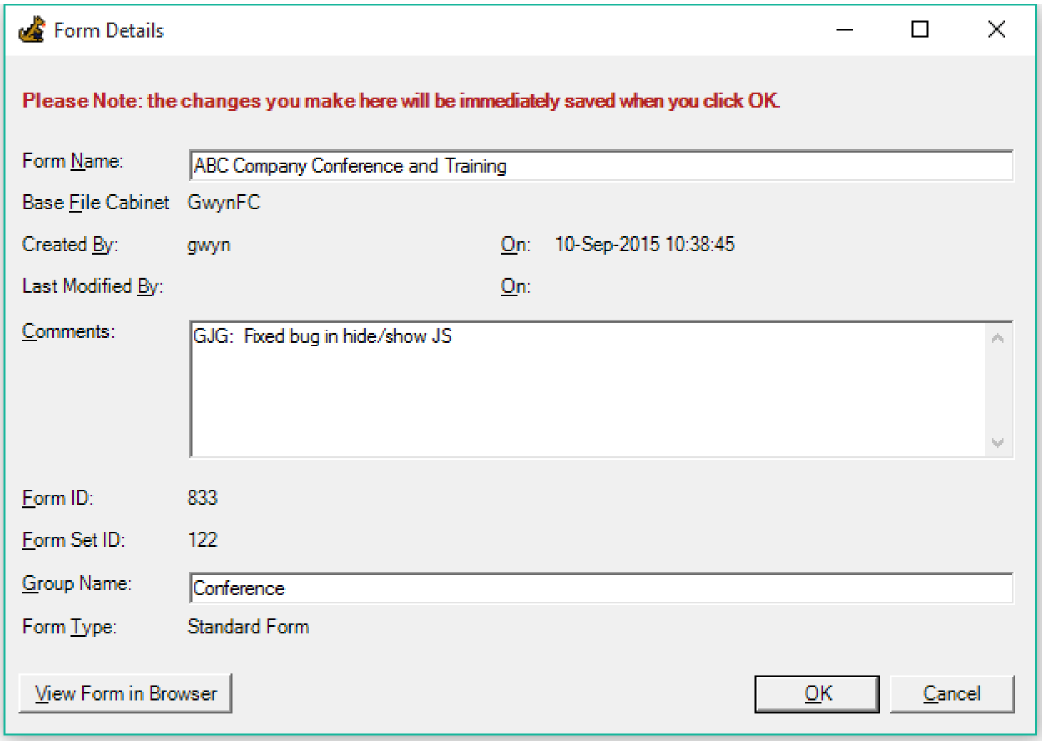
Note: Form details are only available for saved forms, not new forms.

Form Version Details

To view the details of a specific form version:

- With a form open in the Designer, select **Form>Details**. Or in the **Open Form** dialog, select a form version and click **View Details**.

  You can modify the **Form Name**, **Comments**, or **Group Name**. You can also **View Form in Browser**.



Please Note: the changes you make here will be immediately saved when you click OK.

Form Name: ABC Company Conference and Training

Base File Cabinet GwynFC

Created By: gwyn On: 10-Sep-2015 10:38:45

Last Modified By: On:

Comments: GJG: Fixed bug in hide/show JS

Form ID: 833

Form Set ID: 122

Group Name: Conference

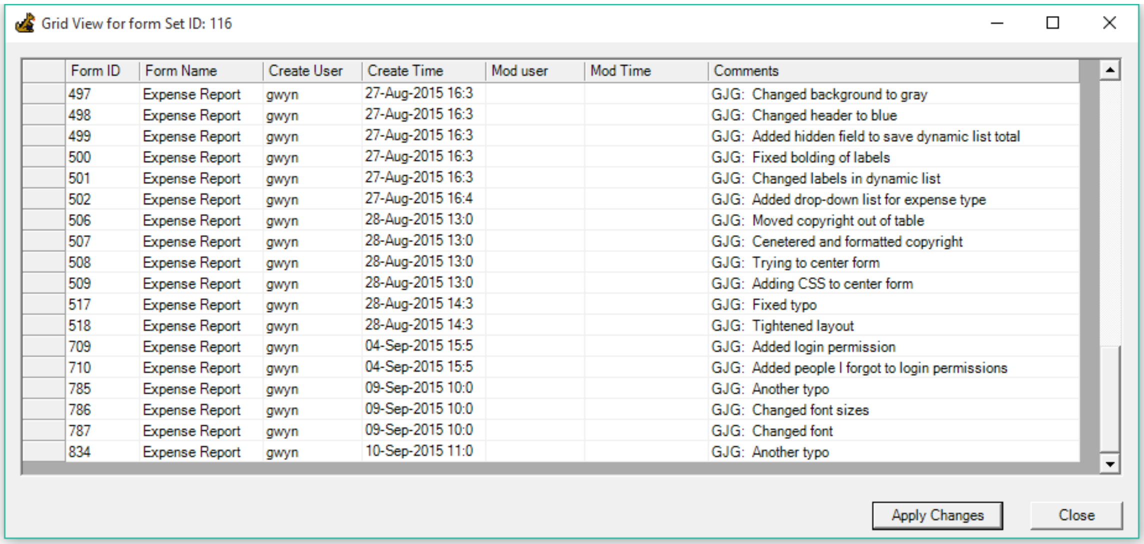
Form Type: Standard Form

Form Set Details

To view the details of all the form versions in a form set:

- In the **Open Form** dialog, select a form set and click **View Details**.

You can edit **Form Name** or **Comments** in the table and click **Apply Changes** to save your changes.



Grid View for form Set ID: 116

Form ID	Form Name	Create User	Create Time	Mod user	Mod Time	Comments
497	Expense Report	gwyn	27-Aug-2015 16:3			GJG: Changed background to gray
498	Expense Report	gwyn	27-Aug-2015 16:3			GJG: Changed header to blue
499	Expense Report	gwyn	27-Aug-2015 16:3			GJG: Added hidden field to save dynamic list total
500	Expense Report	gwyn	27-Aug-2015 16:3			GJG: Fixed bolding of labels
501	Expense Report	gwyn	27-Aug-2015 16:3			GJG: Changed labels in dynamic list
502	Expense Report	gwyn	27-Aug-2015 16:4			GJG: Added drop-down list for expense type
506	Expense Report	gwyn	28-Aug-2015 13:0			GJG: Moved copyright out of table
507	Expense Report	gwyn	28-Aug-2015 13:0			GJG: Cenetered and formatted copyright
508	Expense Report	gwyn	28-Aug-2015 13:0			GJG: Trying to center form
509	Expense Report	gwyn	28-Aug-2015 13:0			GJG: Adding CSS to center form
517	Expense Report	gwyn	28-Aug-2015 14:3			GJG: Fixed typo
518	Expense Report	gwyn	28-Aug-2015 14:3			GJG: Tightened layout
709	Expense Report	gwyn	04-Sep-2015 15:5			GJG: Added login permission
710	Expense Report	gwyn	04-Sep-2015 15:5			GJG: Added people I forgot to login permissions
785	Expense Report	gwyn	09-Sep-2015 10:0			GJG: Another typo
786	Expense Report	gwyn	09-Sep-2015 10:0			GJG: Changed font sizes
787	Expense Report	gwyn	09-Sep-2015 10:0			GJG: Changed font
834	Expense Report	gwyn	10-Sep-2015 11:0			GJG: Another typo

Clone Form

Clone a form design, creating a new version or an entirely new form set.

To clone a form:

1. Select **File>Open**. The **Open Form** dialog opens.
2. Select the form version you want to clone.
3. Click **Clone**. You are prompted as to whether you would like to create a new form set.
4. Click **Yes** to create an entirely new form set. Click **No** to create a new form version within the same form set. The clone is created and displayed in the list.

Republish Form

Republish a new version of the form set with the same properties and design of the selected form version.

To republish a form:

1. Select **File>Open**. The **Open Form** dialog opens.
2. Select the form version you want to republish. If you go to the **Latest Version View**, you can select multiple form versions to republish at once. Use **CTRL+click** or **SHIFT+click** to select multiple form versions.
3. Click **Republish**. The new form version is created in the same form set and displayed in the list.

Hide or Show Form

A form is either **Visible** to end users or **Hidden** from end users. You may want to hide the form from end users while it is being developed then make the form visible once it is ready.

When a form is accessed by its name or form set ID, the highest, visible form version in the form set will come up. Note that if a form is accessed by its form ID, the specific form version will come up, even if it is hidden. See Form Sets and Versions for more information.

To hide a form when saving:

- On the **Save Form** dialog, uncheck the **Visible** option. See Save Form for more information.

To hide a visible form:

1. Select **File>Open**. The **Open Form** dialog opens.
2. Select the form version you want to hide.
3. Click **Hide**. The form is hidden and its icon changes to ⛔.

   To see hidden forms in the **Open Form** list, check on the **Hidden** display option at the bottom.

To make a hidden form visible:

1. Select **File>Open**. The **Open Form** dialog opens.
2. Select the hidden form version you want to make visible.

   To see hidden forms in the **Open Form** list, check on the **Hidden** display option at the bottom.
3. Click **Show**. The form is made visible.

Note: When viewing the form list in the **Open Form** dialog, you can display hidden forms by turning on the **Hidden** display option. See Open Form View Options for more information.

Form Groups

Group forms logically by assigning them a form group (e.g. "Reports"). When opening or managing forms you can filter the form list based on a form group. See Open Form View Options for more information.

To assign a form group:

1. View the form details. See View Form Details for more information.
2. Enter a **Group Name**.
3. Click **OK**. The form group is assigned.

Delete Form

To delete a form:

1. Select **File>Open**. The **Open Form** dialog opens.
2. Select the form version or form set you want to delete. Deleting a form set deletes all of its form versions.
3. Click **Delete**. You are prompted to confirm the delete.

   Notes:

   - If you delete the last form version in a form set, the form set is also deleted.
   - You cannot delete a form version if saved drafts or submitted form documents are using that design. See Submit Form and Configure Save Draft for more information.

     In this case, the Designer will give you the option to hide the form version instead. See Hide or Show Form for more information.
4. Click **Yes**. The form version or form set is deleted.

Form Design Report

View a report that details the design of the form, including the form fields' properties, dynamic list columns' properties, permissions, and more.

To view a report of the form design:

1. Select **File>FormsiQ Report**. The **Save As** dialog opens.
2. Browse to the desired location.
3. Click **Save**. The report is saved in .xml format and opens automatically.

Maintain Other Tables List

"Other" tables are database tables or views that are not FDD file cabinets or lookup tables. You may want to use these tables to:

- Provide option list values for select or lookup fields
- Autofill data into fields
- Store data from fields
- Verify sign-in data

A table or view must be entered in the Other Tables List in order to make it available for these features.

Note: In order to maintain the Other Tables List, you must have the **Edit Forms iQ SQL** task permission.

See the following for instructions:

- Add Table
- Delete Table

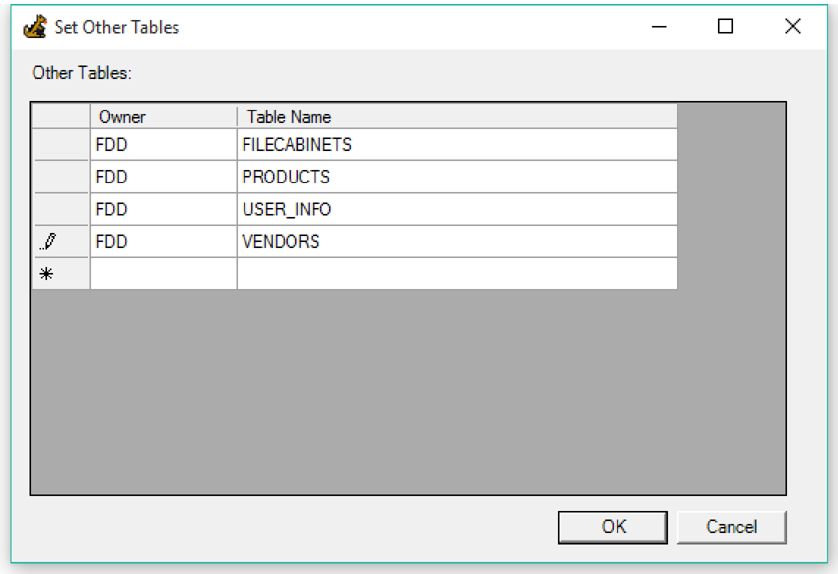
Add Table

To add a table to the other tables list:

1. Select **Tools>Create / Maintain Other Tables List**. The **Set Other Tables** dialog displays.

2. Click in the empty row at the bottom of the list.

3. Enter the table's **Owner** and the **Table Name**.



	Owner	Table Name
	FDD	FILECABINETS
	FDD	PRODUCTS
	FDD	USER_INFO
✎	FDD	VENDORS
*		

4. Click **OK**. Your changes are saved.

Delete Table

To delete a table from the other tables list:

1. Select **Tools>Create / Maintain Other Tables List**. The **Set Other Tables** dialog displays.
2. Select the row of the table you want to delete by clicking on the gray area to the left of the row.
3. Hit the **DELETE** key. The table is deleted from the list.
4. Click **OK**. Your changes are saved.

Export and Import Forms

Export

Export a form design to a .html file, then use the file to import the form design to another FDD database (e.g. from a test system to a production system).

To export a form design:

1. If the form uses a lookup table for option list properties in a select or lookup field, autofill, or sign-in verification, consider whether you want the lookup table values to be exported with the form. You can choose to exclude the values, exporting only the lookup table's structure, in the Global Defaults. See Global Defaults for more information.

2. Open the form you want to export. See Open Form for more information.

3. Select **File>Export**. The **Save As** dialog opens.

4. Browse to the desired location.

5. Click **Save**. The form design is exported to a .html file.

Note: The .html export file is intended to be imported into Forms iQ Designer and should not be modified in another application. If you want to modify a form design in another application, like a text editor, use the **Open With** option instead. See Open Page With Other Editor for more information.

Import

Import a .html export file containing a form design to the desired FDD database (e.g. from a test system to a production system).

The import process prompts you with questions on the file cabinet and lookup tables used in the form design. The import process will also handle other tables used in the form design automatically. See the following for more information on how these objects are used in the form design:

- The form has a base file cabinet and file cabinets are also used for option list properties in a select or lookup field, autofill, or sign-in page verification.

- Lookup tables are used for option list properties in a select or lookup field, autofill, or sign-in page verification.

- Other tables are used for option list properties in a select or lookup field, autofill, sign-in page verification, or data storage.

Note: In order to create file cabinets and lookup tables on import of a form, you must be a Feith Admin (on Oracle) or a Database Admin (on MS SQL Server) and have the **Create File Cabinets** and **Create/Modify Lookup Tables** task permissions. See Feith Control Panel User Guide for more information on user types and task permissions.

To import a form design:

1. Select **File>Import**. The **Open** dialog opens.

   Alternatively, you can import into an existing form set. To do this, select **File>Open**, select the desired form set, and click **Import**. You will be prompted to confirm the form set you are importing into and then the import process will proceed normally.

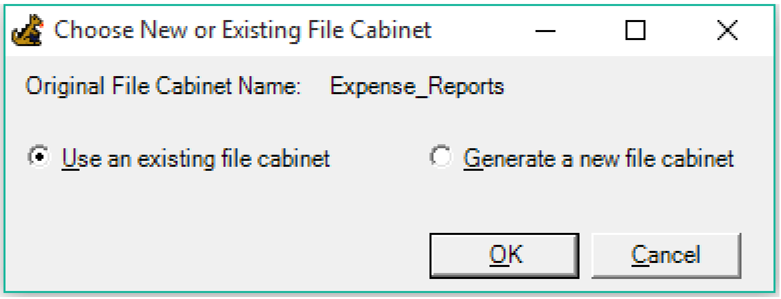
2. Browse to and select the .html export file, then click **Open**.

   Note: If you happen to be importing the form to the same FDD database from which you exported it, you will be prompted as to whether you want to use the same file cabinet and lookup table mappings. Click **Yes** to keep the file cabinet and lookup table mappings and the import is completed. Click **No** to map to different file cabinets and lookup tables in the import process.

3. In the **Choose New or Existing File Cabinet** screen, choose whether you want to:

   - **Use an existing file cabinet:** Use a file cabinet that already exists.

   - **Generate a new file cabinet:** Create a new file cabinet for this form.

   Choose New or Existing File Cabinet

   Original File Cabinet Name: Expense_Reports

   ● Use an existing file cabinet   ○ Generate a new file cabinet

   Click **OK**.

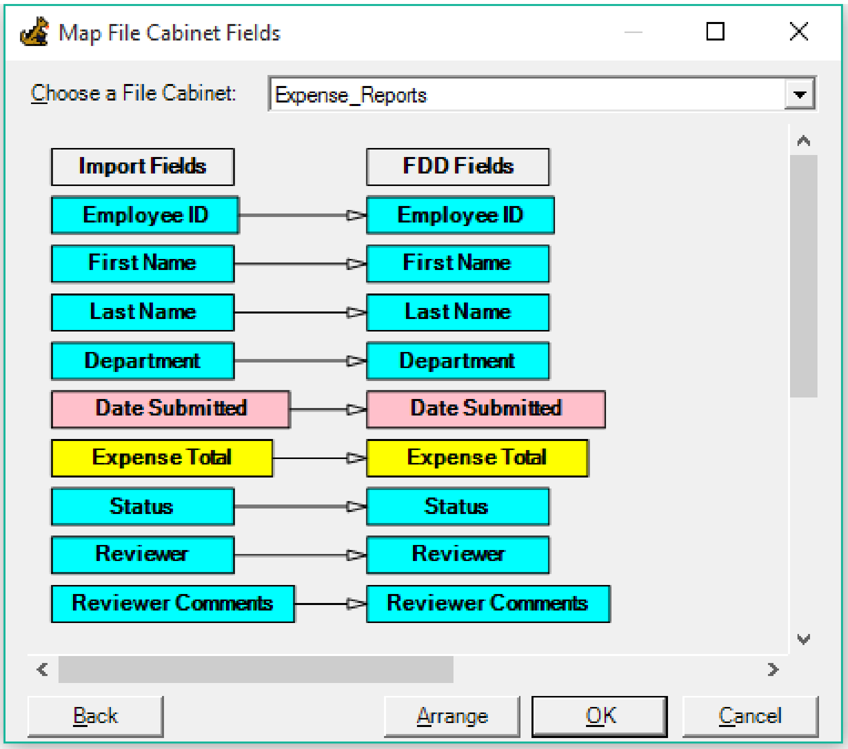
4. Depending on your previous selection, you are prompted to either:

- **Map File Cabinet Fields**, if you chose **Use an existing file cabinet**:

    1. In the **Map File Cabinet Fields** screen, **Choose a File Cabinet** on which you want to build the imported form.
    2. Map the export file's file cabinet fields to the selected file cabinet fields to change file cabinet assignment in the form. String fields are blue, numeric fields are yellow, and date fields are pink. Fields must be mapped to the same type and cannot be mapped to a field of shorter length.

       Click on a field on the left and drag to a field on the right. An arrow is drawn, from left to right, indicating that the export file's field maps to the selected file cabinet's field. Map as many fields as needed. To delete a mapping, select the desired arrow and hit the **DELETE** key.

       Tip: When you are mapping the form's base file cabinet, not all fields have to be mapped. If you choose not to map one of the file cabinet fields, a form field that was storing to that file cabinet field will store to the document instead. Note that at least one form field must store to the form's base file cabinet prior to saving the form.



Choose a File Cabinet: Expense_Reports

Import Fields	FDD Fields
Employee ID	Employee ID
First Name	First Name
Last Name	Last Name
Department	Department
Date Submitted	Date Submitted
Expense Total	Expense Total
Status	Status
Reviewer	Reviewer
Reviewer Comments	Reviewer Comments

    3. Click **OK**. The file cabinet fields are mapped and file cabinet assignment is updated for the form being imported.

- **Set Up New File Cabinet**, if you chose **Generate a new file cabinet**:

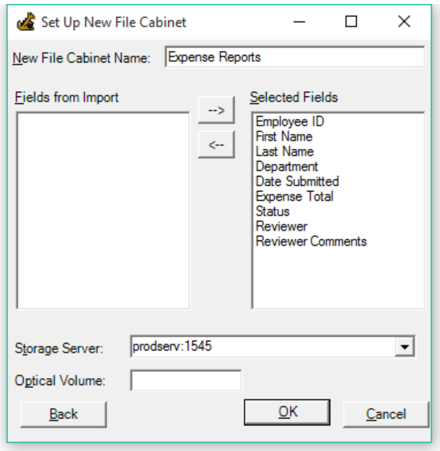
  1. In the **Set Up New File Cabinet** screen, enter the **New File Cabinet Name**.

  2. By default, all fields are selected to include in the new file cabinet. Optionally de-select a field by selecting it and clicking the left arrow button <--. The field is moved to the **Fields from Import** list and will not be included in the new file cabinet.

     To include a field in the new file cabinet, just move it back to the **Selected Fields** list by clicking the right arrow button.

  3. Select the **Storage Server** where the pages in the file cabinet will be stored.

  4. Optionally enter an **Optical Volume**.



Set Up New File Cabinet

New File Cabinet Name: Expense Reports

Fields from Import

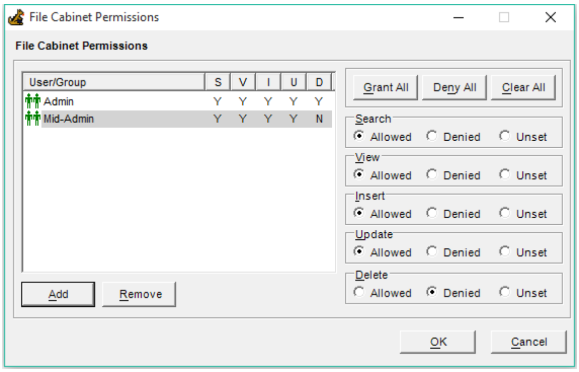
Selected Fields:
Employee ID
First Name
Last Name
Department
Date Submitted
Expense Total
Status
Reviewer
Reviewer Comments

Storage Server: prodserv:1545

Optical Volume:

  5. Click **OK**. The file cabinet is created and the **File Cabinet Permissions** dialog opens.

  6. Optionally assign users resource permission to the file cabinet. Click **Add** and select one or more users and groups to assign. Optionally change the **Search**, **View**, **Insert**, **Update**, and **Allowed** permissions to **Allowed**, **Denied**, or **Unset**. See Feith Control Panel User Guide for more information on file cabinet resource permissions.



File Cabinet Permissions

File Cabinet Permissions

User/Group	S	V	I	U	D
Admin	Y	Y	Y	Y	Y
Mid-Admin	Y	Y	Y	Y	N

Search: ● Allowed ○ Denied ○ Unset

View: ● Allowed ○ Denied ○ Unset

Insert: ● Allowed ○ Denied ○ Unset

Update: ● Allowed ○ Denied ○ Unset

Delete: ○ Allowed ● Denied ○ Unset

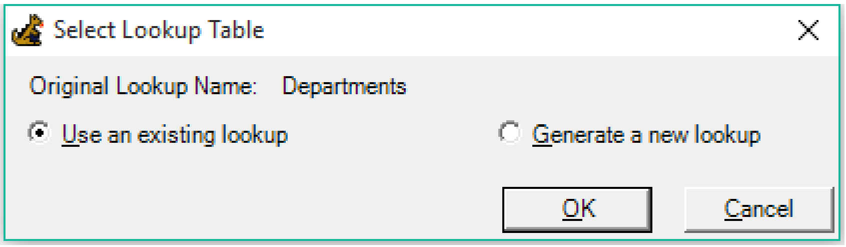
  7. Click **OK**. Any selected users or groups are assigned to the file cabinet. The file cabinet assignment for the imported form is updated to use this new file cabinet.

5. If you used more file cabinets in the form design, you will be prompted for those file cabinets as well. Repeat the previous steps as needed.

6. If there is a lookup table in your form design, you are prompted with the **Select Lookup Table** screen. Choose whether you want to:

   - **Use an existing lookup:** Use a lookup table that already exists.

   - **Generate a new lookup:** Create a new lookup table for this form.

   Select Lookup Table

   Original Lookup Name: Departments

   ● Use an existing lookup ○ Generate a new lookup

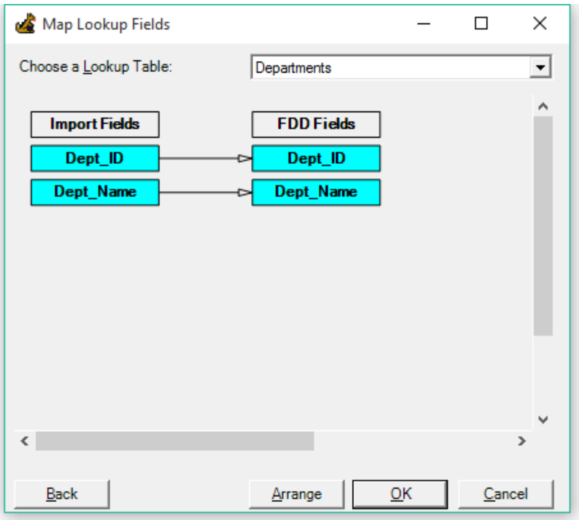
   Click **OK**.

7. Depending on your previous selection, you are prompted to either:

   - **Map Lookup Fields**, if you chose **Use an existing lookup**:

     1. In the **Map Lookup Fields** screen, **Choose a Lookup Table** on you want to use for your form.
     2. Map the export file's lookup table columns to the selected lookup table columns to change the lookup assignment in the imported form. String columns are blue, numeric columns are yellow, and date columns are pink. Columns must be mapped to the same type and cannot be mapped to a column of shorter length.

        Click on a column on the left and drag to a column on the right. An arrow is drawn, from left to right, indicating that the export file's column maps to the selected lookup table's column. Map as many columns as needed. To delete a mapping, select the desired arrow and hit the **DELETE** key.

        Map Lookup Fields

        Choose a Lookup Table: Departments

        Import Fields → FDD Fields

        Dept_ID → Dept_ID

        Dept_Name → Dept_Name

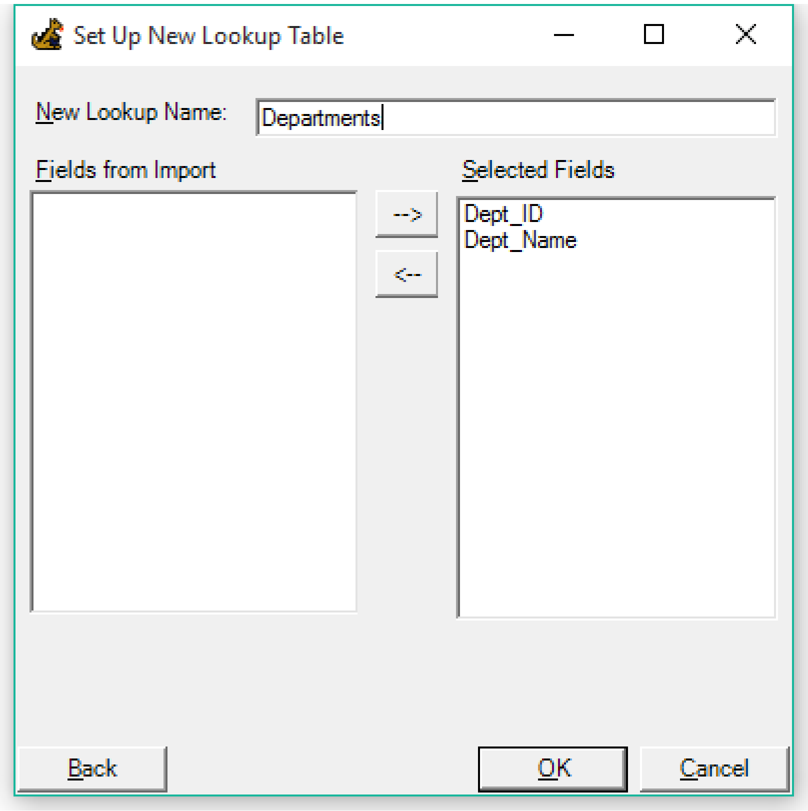
     3. Click **OK**. The lookup table columns are mapped and the lookup assignment is updated for the form being imported.

- **Set Up New Lookup Table**, if you chose **Generate a new lookup**:

    1. In the **Set Up New Lookup Table** screen, enter the **New Lookup Name**.

    2. By default, all columns are selected to include in the new lookup table. Optionally de-select a column by selecting it and clicking the left arrow button <--. The column is moved to the **Fields from Import** list and will not be included in the new lookup table.

       To include a column in the new lookup table, just move it back to the **Selected Fields** list by clicking the right arrow button.

       New Lookup Name: Departments

       Selected Fields
       Dept_ID
       Dept_Name

    3. Click **OK**. The lookup table is created. The lookup assignment is updated for the form being imported.

       Note: If you opted to exclude the lookup table values on export, the lookup table structure will be created but it will not contain any values. See Export and Global Defaults for more information.

8. If you used more lookup tables in the form design, you will be prompted for those lookup tables as well. Repeat the previous steps as needed.

9. If the form used another table, the table and columns will be located and mapped automatically.

   If the table and columns used by the form are *not* on this FDD database, you will receive some warnings during the import and locations where the table was assigned will be changed as follows:

   - Option list properties will be changed to a **Data Source** of **Lookup Table** and it will be as if you had typed in table information by hand. Note that the table is still missing, so you will need to either add the table or change the option list properties.
   - Autofill objects will be deleted.
   - Sign-in page verification properties will be set to **None**.
   - Data storage objects will be deleted.

10. If the form used Feith Developer CSS and JavaScript objects, and those object do not exist on this FDD database, then you will receive a message that those objects are not available. Click **Yes** to continue with the import anyway and remove the references to those objects. Click **No** to cancel import of the form. See Set Form Properties for more information.

11. The form is imported and you receive a prompt that confirms the import is complete.

12. Save the new form you just imported on this FDD database. See Save Form for more information.

    If you imported into an existing form set, the imported form will be saved as a new version of that form set. Otherwise, saving the imported form creates an entirely new form set.

Note: Form permissions are not migrated when a form is exported and imported into a different database. Permissions will need to be re-set after the form is imported. See Permissions for more information.

Appendix

Command Line Arguments

Forms iQ Designer may be run from a command prompt to export and import form designs.

Arguments

ARGUMENT	EXAMPLE	DESCRIPTION
[user]	jsmith	Login user name.
[password]	jsmith123	Login user's password
[database connect string]	fdd	Database to which you want to connect.
[operation]	export *or* import	The operation you want to perform, either "export" or "import".
[form id]	5	When exporting, the Form ID of the form design you want to export. The Form ID can be found by opening the desired form and selecting **Form>Details**.
[export/import file name]	C:\Users\John\Desktop\NewContact.htm	When exporting, the name of the form design export file and, if desired, the path to the location where you want the file saved. The file must have the extension ".htm". If no path is specified, the export file is saved in your operating system user's folder (e.g. C:\Users\John). When importing, the path and name of the form design export file to import.

When exporting, arguments must be given in the following order delimited by a semicolon:

```
[user];[password];[database connect string];export;[form id];[export file name]
```

When importing, arguments must be given in the following order delimited by a semicolon:

```
[user];[password];[database connect string];import;[import file name]
```

Note that none of the arguments may contain spaces.

Examples

For example, exporting a form design called "New Contact" to a file called "NewContact.htm" on your desktop:

```
C:\Feith\FormsIQ\designer\formsiq.exe
jsmith;jsmith123;fdd;export;5;C:\Users\John\Desktop\NewContact.htm
```

For example, importing a form design from a file on your desktop called "NewContact.htm":

```
C:\Feith\FormsIQ\designer\formsiq.exe
jsmith;jsmith123;fdd;import;C:\Users\John\Desktop\NewContact.htm
```

Glossary

A

Alternate View: A custom print view that you tailor for users who click the Printer Friendly View button after submitting a form.

Autofill: Automatically populates fields as the user fills out the form. For example, autofill might be configured to retrieve personal information, such as address and phone number, based on an entered name value, then populate fields with these values.

C

Cloning Data: Configure a form to have fields that can be cloned, allowing the end user to create a new form with values cloned from an existing, submitted form document.

D

Dynamic List: Allows the user to enter multiple lines of detail pertaining to a single item. For example, a dynamic list might be used on an expense report form to capture multiple lines of expenses.

F

Field Set: Restrict access to certain form fields - the "field set" - based on the user who logged into the form.

File Cabinet: A storage area in the FDD system for submitted Forms iQ form documents. A form must be built on a file cabinet.

File Upload Field: Allows the user to browse and select a file to upload with the form.

Form ID: The internal ID of a form version within a form set.

Form Set: Each time you create a new form, a form set is created for the new form. Multiple versions of the form can be created within this set. The form set is identified by an internal Form Set ID.

Forms iQ Server: The web, server component of Forms iQ that actually runs the forms in the browser.

G

Go Live: Users who have access to a submitted form page can click the Go Live button and update the page.

Grid: Displays data from a SQL query in grid format. The grid field can also display data in a "tree" control, grouping like data under an expandable section.

H

Hidden Field: Stores data that is not visible on the form.

L

Label Field: Displays read-only text; the value of the label is set based on the default value or autofill option.

Login: Users need to provide an FDD login in order to access and submit the form.

Lookup Field: Provides a list of values for the end user to select from.

Lookup Table: A list of suggested vales that can be set on, say, a lookup field or select field.

O

Other Table: Database tables or views that are not FDD file cabinets or lookup tables.

P

Password Field: Similar to a textbox field except when data is entered into the field the characters display as asterisks for privacy.

R

Regular Expression: Provides a way to define a string pattern, say for a validation to make sure data is entered correctly.

S

Select Field: Provides the user with a list of values to select from; a drop-down list.

Show History: An end user can click the Show History link on a submitted form and view the history of changes to the page.

Sign-In: A sign-in page is added before the form, prompting the user to enter sign-in data to allow them to access the form or sign-in data is populated automatically (the user doesn't see the sign-in page) in order to autofill data on the main form.

Signature Field: Accepts a signature, which signs the form. The signature is based on standard PKI (Public Key Infrastructure).

Smart Icon: An image from a defined set is displayed conditionally, such as based on the value of a field.

SQL: The Structured Query Language used to retrieve data from the database.

Storage of Data: Form field data is stored in the form's base file cabinet, another database table, or the form document itself.

V

Validation: An expression that checks the validity of data entered by the end user in the form.

Verification (for Sign-In): Check the user-entered sign-in data against a file cabinet, lookup table, or another table, restricting access to the form unless they enter the correct information.

Revision History

Date	Version	Author	Description
10/22/2015	9.2.4.1	GJG	Updated documentation for this new version.