



FileNet Forms Manager

Solution Integration Guide

Release 5.0.0

August 2005

FileNet is a registered trademark of FileNet Corporation.
All other product and brand names are trademarks or registered trademarks of their respective companies.
Due to continuing product development, product specifications and capabilities are subject to change without notice.

Copyright © 2003, 2005 FileNet Corporation. All rights reserved.

FileNet Corporation
3565 Harbor Boulevard
Costa Mesa, California 92626
800.FILENET (345.3638)
Outside the U.S., call:
1.714.327.3400
www.filenet.com

Notices

FileNet Forms Manager documentation contains information proprietary to FileNet Corporation (FileNet). Because of on going product development, changes to product specifications and capabilities may be made without notice. You may not disclose or use any proprietary information or reproduce or transmit any part of this documentation in any form or by any means, electronic or mechanical, for any purpose, without written permission from FileNet. FileNet has made every effort to keep the information in the documentation current and accurate as of the date of publication or revision. However, FileNet does not guarantee or imply that the documentation is error-free or accurate with regard to any particular specification. **In no event will FileNet be liable for direct, indirect, special incidental, or consequential damages resulting from any defect in the documentation, even if advised of the possibility of such damages.** No FileNet agent, dealer, or employee is authorized to make any modification, extension, or addition to the above statements. FileNet may have patents, patent applications, trademarks, copyrights, or intellectual property rights covering subject matter in this document. Furnishing this document does not provide any license to these patents, trademarks, copyrights, or other intellectual property. FileNet is a registered trademark of FileNet Corporation. Copyright © 2003, 2005 FileNet Corporation. All rights reserved.

All other brands, products, and company names mentioned are trademarks of their respective owners.

Typographical Conventions

Where applicable, this document uses the conventions in the following table to distinguish elements of text.

Convention	Usage
UPPERCASE	Environment variables, status codes, utility names.
Bold	Paths and file names, program names, clickable user-interface elements (such as buttons), and selected terms such as command parameters or environment variables that require emphasis.
<i>Italic</i>	User-supplied variables and new terms introduced in text.
<italic>	User-supplied variables that replace everything between and including the angle bracket delimiters (< and >).
Monospace	Code samples, examples, display text, and error messages.

Table of Contents

Table of Contents	3
Introduction	4
Definitions	4
Parameters for prefill data	5
Data parameter	5
URL-encoded data parameter	5
Query string parameters	6
Using encoding calculation functions	6
Parameter ordering	6
Invalid parameters	7
Invalid syntax	7
Invalid data	7
Multiple values in one field	7
Formatting Data	7
Formatted cells	8
Prefilling a table	8
Passing data to a calculated cell	9
Overriding calculations	9
Passing data to a mapped field	10
Passing an empty value	10
Limitations	10
Auto-incrementing fields	10
Fields with a length limit	10
Lookup fields	10

Introduction

This guide describes how to open a new eForm using a URL. With this feature, users can open an eForm by clicking a link on a web site or on any document that supports hyperlinks. This guide also describes how to add parameters on the URL query string so that the eForm opens with data prefilled in the fields that you specify on the string.

Definitions

Field and cell: “Field” refers to the graphic object on the eForm that holds information. Each field contains a “cell,” a holding place for data. In this document, you will see the term “cell” used when referring to the name-value pairs that you add to the URL query string. The term “field” is used when referring to the areas on the eForm that are prefilled with data.


eForm: In this document, the form that the user works with is referred to as an “eForm.”

Open an eForm using a URL

You can create a link to an eForm that resides in an eForms Central library and place the link on any document that supports hyperlinks (for example, a web page or a Microsoft Word document). When the user clicks the link, the eForm opens.

To obtain a link to an eForm

1. Navigate to the desired template in the eForms Central library.
2. Select and copy the template ID for the template.

<input type="checkbox"/>	▲ Title	Template ID	Created	Modified	Type
<input type="checkbox"/>	 Damage Assessment	WC - Assessment	4/11/2005	4/11/2005	ITX Form Template

3. In a document that supports hyperlinks, type the URL path to the template using the following format:
`http://<server_name>/<library_name>/Form.aspx?template=local,`
where *server_name* is the name of the server, and *library_name* is the name of the library.
4. Paste the template ID at the end of the URL path. If the template ID contains reserved characters (for example, spaces, hyphens), url-encode the characters. The following example uses a server named “FMServer,” a library named “Claims” and the template ID “WC - Assessment.”
`http://FMServer/Claims/Form.aspx?template=local, WC%20%2d%20Assessment`

Parameters for prefill data

To have the eForm open with prefill data, you must use two additional parameters.

- a “data” parameter
- a “url-encoded data” parameter

NOTE Both parameters must be properly url-encoded. See [“Query string parameters” on page 6](#).

Data parameter

The data parameter must be named “data.” Its value specifies the name of a second parameter which contains the url-encoded data that is used to prefill the eForm. You can use any valid parameter name for the second parameter. The following “data” parameter value specifies that the “myPrefill” parameter contains the actual prefill data.

```
urlencoded, myPrefill
```

URL-encoded data parameter

The url-encoded data parameter, referenced by the data parameter, provides a url-encoded collection of name-value pairs that make up the prefill data. Each name-value pair consists of the cell’s symbolic name and its prefill data. In the following example, the parameter value specifies that a cell called “LastName” will prefill with the value “Smith.”

```
Lastname=Smith
```

NOTES

- The cell name and the cell value must be url-encoded (for example, Name=John%20Smith).
- The cell name and the label of the corresponding field on the eForm may not always be the same. This is because for each cell on the template, the eForm author can specify a cell name and a field title. The user sees the title in the title area of the field on the eForm. When you reference a cell in a name-value pair, use the cell name, not the title.

Query string parameters

All query string parameter values must be url-encoded. Therefore, if it's placed on the query string, the url-encoded data parameter value must be double-encoded as follows.

1. Create the url-encoded data parameter value.

Example: Name=Alexander%20Nelson&Job=A%26W&Age=18

2. Url-encode the result again.

Example: Name%3dAlexander%2520Nelson%26Job%3dA%2526W%26Age%3d18

The following query string includes both the data parameter (in blue) and the url-encoded data parameter (in red).

```
http://FMServer/Claims/Form.aspx?template=local, WC%20%2d%20Assessment&data=
urlencoded,myPrefill&myPrefill=Name%3dAlexander%2520Nelson%26Job%3dA%2526W%26Age%3d18
```

If a user clicked this link, an eForm would open prefilled with "Alexander Nelson" in the Name cell, "A&W" in the Job cell and "18" in the Age cell.

NOTE Unexpected results will occur if the data parameters are not properly encoded. See "[Invalid parameters](#)" on page 7 for more information.

Using encoding calculation functions

If, in eForms Designer, you use the Hyperlink button style to create dynamic links on an eForm, you can use the URLEncode and URLEncoding external functions to conveniently build the data parameters. The following example shows the same link as above but with an embedded calculation (shown in red text) that uses the encoding functions.

```
http://FMServer/Claims/Form.aspx?template=local, WC%20%2d%20Assessment&data=
urlencoded,myPrefill&myPrefill=<<External("URLEncode", (External("URLEncoding",
"Name,Job,Age", Cell1, Cell2, Cell3))) >>
```

If a user clicked the hyperlink button on the eForm, a new eForm would open and values from Cell1, Cell2, and Cell3 on the original eForm would prefill the Name, Job, and Age cells on the new eForm.

NOTE If you create a link from a calculation, make sure that the calculation includes properly formatted data. See "[Formatting Data](#)" on page 7.

For more information about the Hyperlink button feature in eForms Designer, see "Configure a button with Hyperlink" in the eForms Designer Help.

Parameter ordering

The fields on the eForm are filled with the data that you include on the URL in the same order as it occurs in the url-encoded data parameter. Therefore, if you want fields to fill in a certain order (for example, to have one calculation fire before another), you must place the name-value pairs in that same order on the query string.

Invalid parameters

If you use incorrect syntax in the data parameters, unexpected results will occur. Following are some of the behaviors that may occur when invalid syntax or invalid data exists on the query string.

Invalid syntax

- If the syntax is incorrect for the name-value pairs, various results may occur. For example, if you leave a space character unencoded or single-encoded, the space may display on the eForm as “+.” For example, “Name=John Smith” (unencoded) or “Name=John%20Smith” (single-encoded) may appear on the eForm as John+Smith.
- If the syntax is incorrect for the data parameters, various results may occur depending on where the invalid syntax occurs, including the following:
 - The eForm may open but the fields don’t prefill.
 - The eForm may open but fields don’t prefill as expected.
 - Error messages may appear, such as “The ‘data’ parameter is invalid.”

Invalid data

Invalid data, or data that is not correctly formatted for the cell type, is ignored. No value populates the field that is associated with the invalid data, but the other fields populate correctly. For example, if a field on the eForm called Time was configured to accept numerical values only, the following url-encoded data parameter would include invalid data (shown in red text):

```
Time%3done%26Organization%3dAllied
```

On the eForm, the Time field would remain empty, but the Organization field would prefill with “Allied.”

Multiple values in one field

If you attempt to pass in more than one value to a single-value field, all the values appear in the field, separated by comma delimiters. For example, if you include

```
Name%3dJane%26Name%3dEmily%26Name%3dBeth
```

 in the url-encoded data parameter, the Name field will prefill with “Jane,Emily,Beth.”

Formatting Data

Depending on the type of cell that you are passing data to, you must use particular formats for that data. The following table lists the format that is expected for each data type and provides an example of what the value must look like in the data parameter.

NOTE These specific formats are required, regardless of the locale of the current user.

Cell type	Expected data format	Value example
Boolean	0 (for false) or 1 (for true)	0
Character	Same as the cell's formatting configuration. Example: (###) ###-####	(250) 988-0246
Date	YYYY-MM-DD	1959-10-26
Number	Number (up to 9 decimal places) with a period for the decimal separator and with no comma separators.	1350.24
Time	HH24:MM:SS	13:24:36

Formatted cells

When you pass a value to a formatted cell, the data is formatted according to that cell's configuration. For example, if you pass the value "1959-10-26" to a Date cell that has been configured with the format, "Day, Month DD, YYYY," the value prefilled in the Date cell will display as "Wednesday, October 26, 1959."

Prefilling a table

In order to pass data to a table, each column must be represented with multiple parameters. For example, if you want to fill a table cell (or column) with four rows of data, you must reference the same cell name in each of the four name-value pairs.

Here is a sample table:

Name	Age
Robert	28
Belinda	45
Michael	21
Yves	54

To prefill the table with the values shown in the two columns, create the double url-encoded data parameter as follows (for information about url-encoding, see ["Query string parameters" on page 6](#)):

```
Name%3dRobert%26Name%3dBelinda%26Name%3dMichael%26Name%3dYves%26Age%3d28%26Age%3d45%26Age%3d21%26Age%3d54
```


Passing data to a calculated cell

Calculations fire automatically when you pass valid data to a field that is referenced in a calculation.

Overriding calculations

When you pass data to fields that trigger calculations, the results depend on the order in which you place the name-value pairs in the url-encoded data parameter.

For example, suppose that you have fields on the eForm named Cell1, Cell2, and Cell3. Cell2 is configured with a calculation that gets the value of Cell1. Cell3 is configured with a calculation that concatenates the values of Cell1 and Cell2. Therefore, if you pass a value of “A” to Cell1, the calculations fire automatically and the values that display on the eForm are “A” in Cell1, “A” in Cell2, and “A A” in Cell3.

If you also pass data to Cell2, however, you can get different results in Cell3 depending on the order in which the name-value pairs for Cell1 and Cell2 appear on the string.

The following table outlines the data that is returned to Cell3 according to which name-value pairs are placed on the query string and in what order.

Name-value pairs	Cell1	Cell2 calculation: gets Cell1	Cell3 calculation: Concat Cell1,Cell2
Cell1=A	A	A	A A
Cell2=B		B	B
Cell1=A&Cell2=B	A	B	A B
Cell2=B&Cell1=A	A	A	A A

In the third row of the above table, Cell1 comes before Cell2 on the query string. When the URL is invoked, the first field to prefill is Cell1 (with “A”). This triggers Cell2’s calculation and Cell2 fills with “A.” The next name-value pair on the string (“Cell2=B”) overrides the calculated value in Cell2 and fills that field with “B.” The calculation in Cell3 refires and becomes “A B.”

However, in the fourth row of the table, where Cell2 comes before Cell1 on the query string, the first field to prefill is Cell2 (with “B”). This triggers the calculation in Cell3 which becomes “B.” The next name-value pair on the string (“Cell1=A”) fills Cell1 with “A.” This value triggers the calculations in Cell2 and Cell3 and overrides the calculation in Cell3 with “A A.”

Passing data to a mapped field

When you pass data to a field on the eForm that is already prefilled with a document property, the value passed on the query string overrides the mapped data.

Passing an empty value

If you want to prefill a cell with an empty value, you can include an empty value in the url-encoded data parameter. If the cell was previously mapped or calculated to have a value, this value will be replaced with the empty value. In the following example, any preexisting value in the Name cell will be replaced with an empty value.

```
Name%3d%26EmployeeID%3d32541
```

Limitations

The following limitations exist for prefill data:

Auto-incrementing fields

You cannot pass a value to an auto-incrementing field because the auto-increment action fires after the data is passed to the field and overwrites the prefill data.

Fields with a length limit

When you pass data to a field that has been configured in eForms Designer with a length limit (for example, the characters that a user can enter is limited to 10), the field will hold more characters than has been specified. For example, if you pass 20 characters to a field that is limited to 10 characters, all 20 characters will appear in the field on the eForm.

Lookup fields

If you pass a value to a lookup field, the data prefills the field but doesn't trigger the lookup action. This is true even when the user tabs out of the prefilled lookup field.