



# PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

[Making form for adding data to list](#)  
[and reading data from list](#)



This guide describes the steps to create the following scenario:

1. PDF forms are used for filling with data, they will not be saved in the library; all filled data will be stored in the required list.
2. Every record stored in the list can trigger workflow. The workflow will create PDF form for printing purposes.

## Preparing list

As an example, we will use the following list called „Person list“:

<input type="checkbox"/>	Name	Last Name	Age	Telephone
	Torek	Gork	56	123-456-789
	Andrew	Corvin	44	777-888-999
	Lara <small>NEW</small>	Cruise	37	222-444-777
	Bob <small>NEW</small>	Smith	35	111-222-333
	Mike <small>NEW</small>	Bonnegen	43	222-444-123

[+ Add new item](#)

It already has some data inside. We will add similar data to list columns using the form.

## \*Alternative approach

There is also another approach to the case described in this guide. Such steps are marked as “Alternative” and are different from the approach described in “Dynamic XFA form” and in “Creating workflow”.

Difference between approaches:

**Regular** approach:

- Form is not sensitive to changes in the list: all data will be populated via workflow.  
It means that the data in a form will have values which have been stored in the list at the moment of launching workflow.

**Alternative** approach:

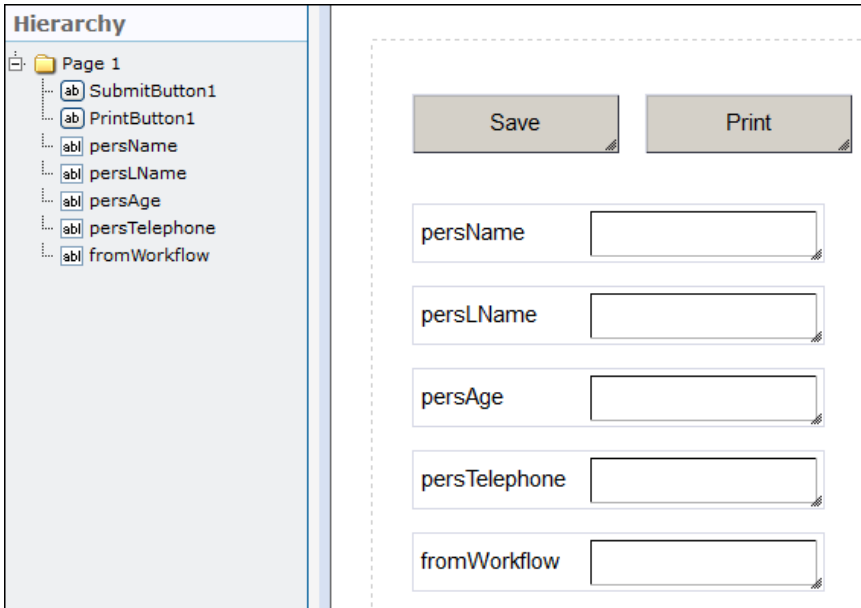
- Form is sensitive to changes in the list: all data will be populated via “Form Load” rule.  
It means that the data in a form will be renewed each time form is opened and will be equal to an item in the list.

**NOTE:**

1. These steps are not necessary. They show the different way to accomplish the goal.  
Choose “Regular” steps or “Alternative”, you cannot combine them.
2. “Alternative” steps are placed under its “Regular” equivalent. All steps without “Alternative” equivalent below are usable in both variants.
3. Do not mix “Alternative” steps with the “Regular” ones.

## Dynamic XFA form

As an example, we will use the following template:



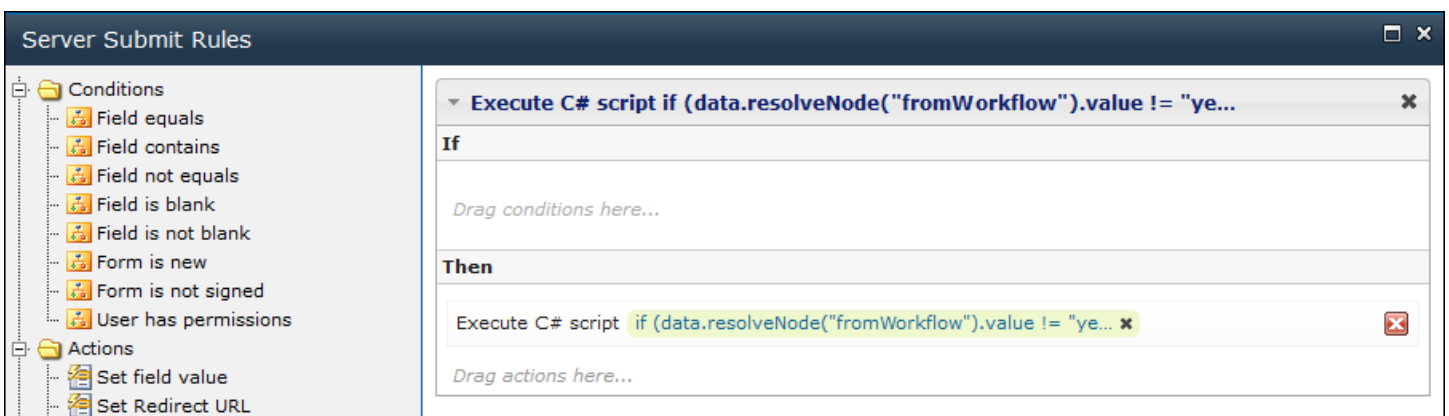
This template has one submit button, print button and text fields.

Text fields “persName”, “persLName”, “persAge” and “persTelephone” will be used to store entered data;

“fromWorkflow” field will check if the form is used for adding data to list or for printing purposes.

It must be hidden, so that user can’t change value inside it.

**Step 1.** Modify “Form Submit” rules: *PDF Form Tools* → *Developer* → *Form Submit*.



**Step 2.** Remove “Save Form” action.

It must be removed, so that the form can’t be saved to the library.

**Step 3.** Add “Execute script” action.

**Code:**

```
if (data.resolveNode("fromWorkflow").value != "yes")  
{  
    SPList targetList = currentWeb.Lists["Person list"];  
    SPLListItem item = targetList.Items.Add();  
    item["Title"] = data.resolveNode("persName").value;  
    item["Last_x0020_Name"] = data.resolveNode("persLName").value;  
    item["Age"] = data.resolveNode("persAge").value;  
    item["Telephone"] = data.resolveNode("persTelephone").value;  
    item.Update();  
}
```


This code will save entered data to the list (Person list).

At the beginning of the script, it checks if “fromWorkflow” field has “yes” value inside it. We need this in order to check if the form is created to add values to the list (“fromWorkflow” is empty), or it is created by the workflow for printing purposes (“fromWorkflow” value is set to “yes”).

**NOTE:** In order to add items to the list, there is a need to use column internal names.

To check the internal name of column, complete the following steps:

1. Open “List Settings” rules: **List Tools** → **List** → **List Settings**.
2. Select the column you want to check (Last Name in our case)



Column (click to edit)	Type
Name	Single line of text
Last Name	Single line of text
Age	Single line of text
Telephone	Single line of text
PersonID	Single line of text
Created By	Person or Group
Modified By	Person or Group

3. Check the end of the URL in browser:

[http://yoursite/layouts/FldEdit.aspx?List={107C4FEE-C0A0-4E35-9288-590DC25E5F48}&Field=Last\\_x0020\\_Name](http://yoursite/layouts/FldEdit.aspx?List={107C4FEE-C0A0-4E35-9288-590DC25E5F48}&Field=Last_x0020_Name)

Where “Field=Last\_x0020\_Name” is the internal name of column.

**Step 3 (\*Alternative).** Add “Execute script” action.

**Code:**

```
if (data.resolveNode("fromWorkflow").value != "")
{
    SPList list = currentWeb.Lists["Person list"];
    int id = int.Parse(data.resolveNode("fromWorkflow").value);
    SPLListItem item = list.GetItemById(Convert.ToInt32(id));
    data.resolveNode("persName").value = item["Title"].ToString();
    data.resolveNode("persLName").value = item["Last_x0020_Name"].ToString();
    data.resolveNode("persAge").value = item["Age"].ToString();
    data.resolveNode("persTelephone").value = item["Telephone"].ToString();
}
```

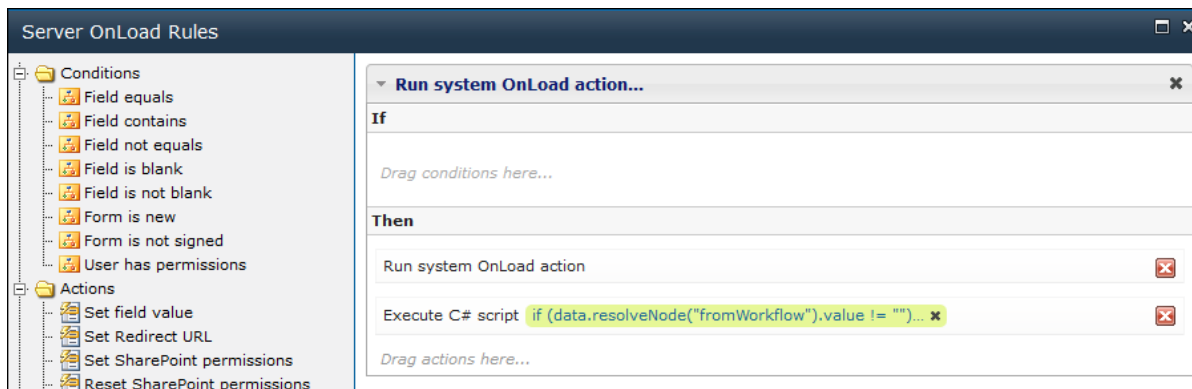
This code will save entered data to the list (Person list).

At the beginning of the script, it checks if “fromWorkflow” field has any value inside it.

We need this in order to check if the form is created to add values to the list (“fromWorkflow” is empty), or it is created by the workflow for printing purposes (“fromWorkflow” value is set to “ID” of currently used list item).

**NOTE:** In order to add items to the list, there is a need to use column internal names.

**Step 4 (\*Alternative).** Modify “Form Load” rules: *PDF Form Tools* → *Developer* → *Form Load*.



**Step 5 (\*Alternative).** Add “Execute script” action under “Run system OnLoad action”.

**Code:**

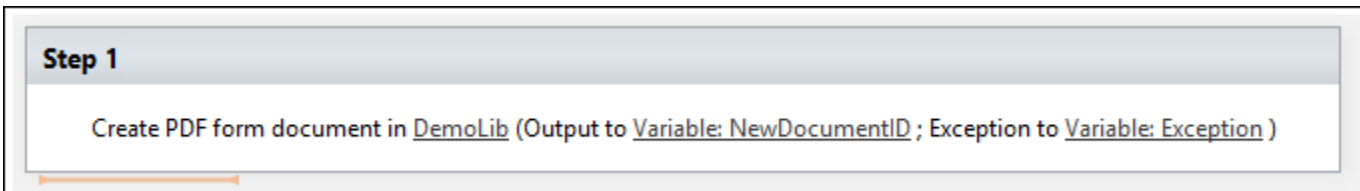
```
if (data.resolveNode("fromWorkflow").value != "")
{
    SPList list = currentWeb.Lists["Person list"];
    int id = int.Parse(data.resolveNode("fromWorkflow").value);
    SPLListItem item = list.GetItemById(Convert.ToInt32(id));
    data.resolveNode("persName").value = item["Title"].ToString();
    data.resolveNode("persLName").value = item["Last_x0020_Name"].ToString();
    data.resolveNode("persAge").value = item["Age"].ToString();
    data.resolveNode("persTelephone").value = item["Telephone"].ToString();
}
```

This code will populate data from list (Person list) using value placed in “fromWorkflow” field.

**NOTE:** In order to add items to the list, there is a need to use column internal names.

## Creating workflow

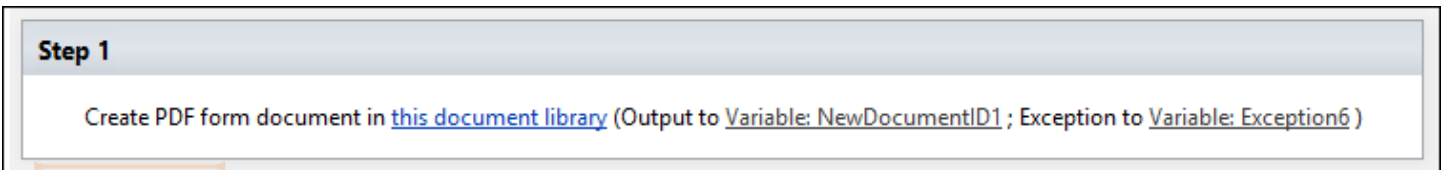
As an example, we will use the following workflow:



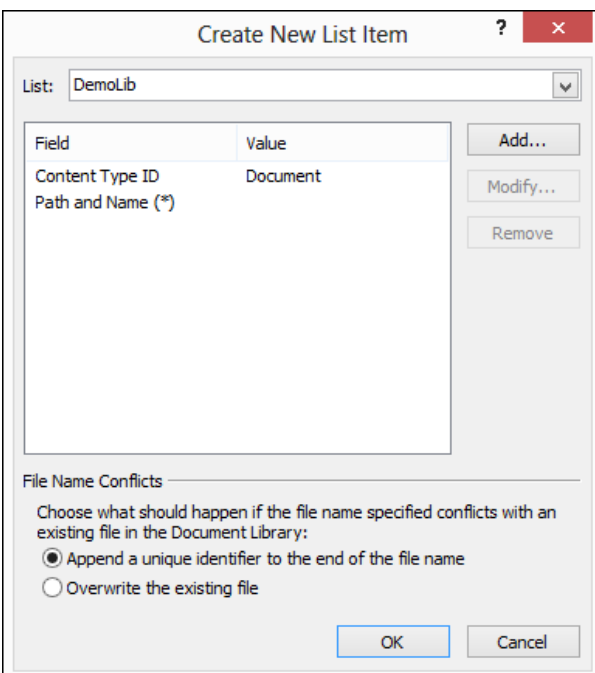
It will create PDF form action. The action will create a form from the template we have made in “Dynamic XFA Form” section of this guide and place it to the “DemoLib” library.

To create the workflow, complete the following steps:

**Step 1.** Add “Create PDF form document” action: **Action** → **PDF Share Forms Enterprise** → **Create PDF form document**.



**Step 2.** Fill “this document library” section out :



**Step 3.** Select a list you want to use (“DemoLib” in our case), change its content type from the document to the content type we have created and change “Path and Name (\*)”.

“Path and Name (\*)” is form’s created name (here it was made in String Builder: “New\_Doc\_[%Current Item:Name%]” ).

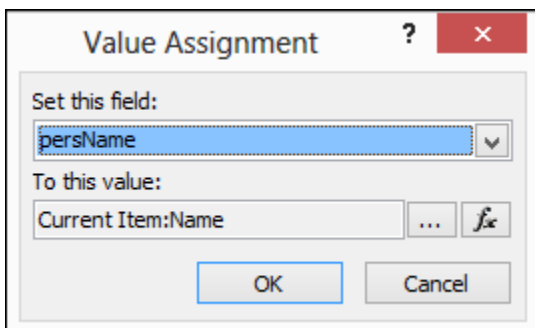


**Step 4.** Add values to fields in the created form.

Click on “Add...” button, choose field name in “Set this field” field and choose value in “To this value”.

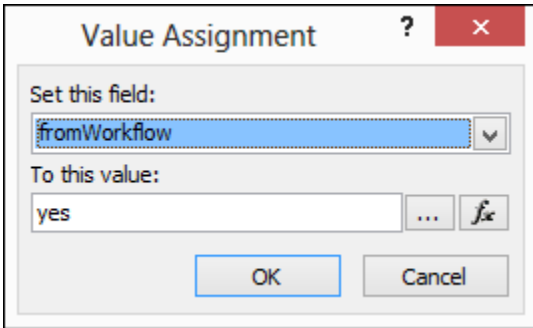
Here workflow will be launched from the list with data, so all necessary values are stored in “Current Item” data source.

For example, the following values have been used to add the name:



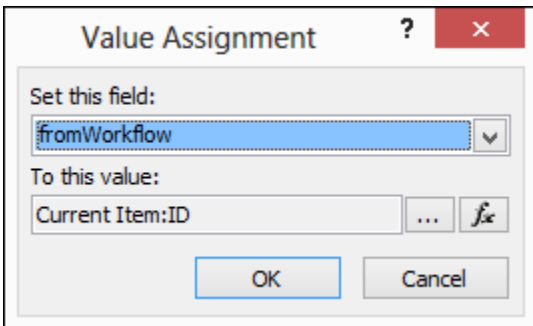
**Step 5.** Add the rest of values to form fields in the workflow action.

**NOTE:** Field “fromWorkflow” does not have column in the list, so you will need to add the value as a text:

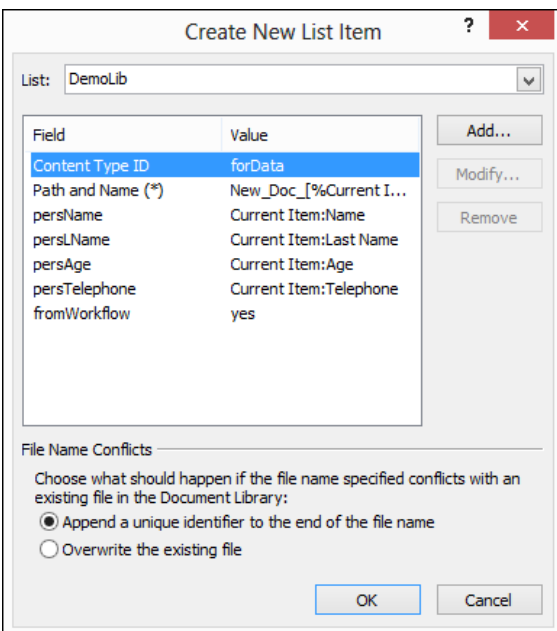


**Step 5 (\*Alternative).** Add the rest of values to form fields in the workflow action.

Field “fromWorkflow” will store currently selected list item’s ID.



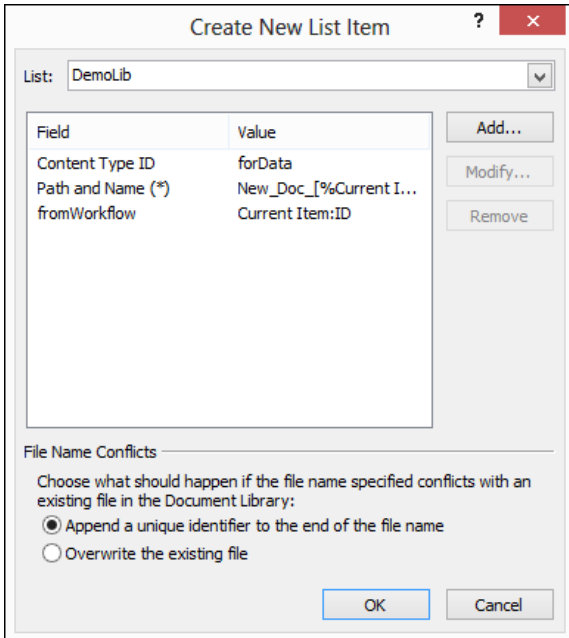
## Runtime result



**NOTE:** You can read more about PDF Share Forms Workflow actions in [PDF Share Forms User guide](#).



## Runtime result (\*Alternative)



The screenshot shows a 'Create New List Item' dialog box with the following details:

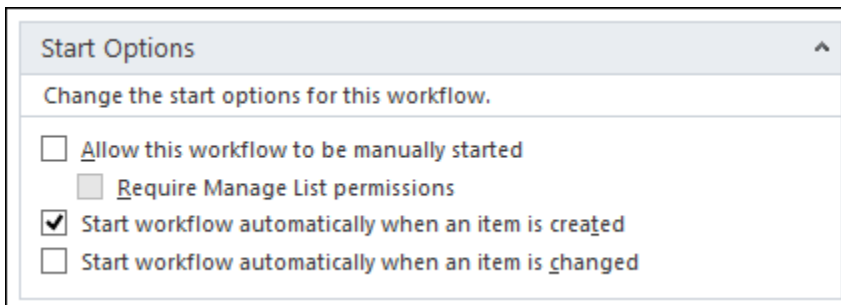
- List:** DemoLib
- Field/Value Table:**

Field	Value
Content Type ID	forData
Path and Name (*)	New_Doc_[%Current I...
fromWorkflow	Current Item:ID
- File Name Conflicts:**
  - Choose what should happen if the file name specified conflicts with an existing file in the Document Library:
  - Append a unique identifier to the end of the file name
  - Overwrite the existing file

**NOTE:** You can read more about PDF Share Forms Workflow actions in [PDF Share Forms User guide](#).

### Step 6. Choose workflow “Start Options”.

In our case it is set to “Start workflow automatically when an item is created”.



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

- Allow this workflow to be manually started
  - Require Manage List permissions
- Start workflow automatically when an item is created
- Start workflow automatically when an item is changed

### Step 7. Save and publish workflow.

## Runtime

**Step 1.** Create a new form from the content type, deploy it to the library and fill it with data:

Save
Print

persName

persLName

persAge

persTelephone

fromWorkflow

**NOTE:** “fromWorkflow” field must be “hidden”, it is set to “visible” only for demonstration purposes.

**Step 2.** Save the form.

As can be seen, you are redirected to the library. Though there is no form as it has not been saved. All data went straight to the list.

<input type="checkbox"/>	Type	Name	Modified
There are no items to show in this view of the "aaa2" document library. To add a new item, click "New" or "Upload".			
<a href="#">+ Add document</a>			

**Step 3.** Go to the “Person list” list.

<input type="checkbox"/>	Name	Last Name	Age	Telephone	listToForm
	Torek	Gork	56	123-456-789	Completed
	Andrew	Corvin	44	777-888-999	Completed
	Lara <small>NEW</small>	Cruise	37	222-444-777	Completed
	Bob <small>NEW</small>	Smith	35	111-222-333	Completed
	Mike <small>NEW</small>	Bonnegen	43	222-444-123	Completed
	John <small>NEW</small>	Olsen	38	123-456-789	In Progress
<a href="#">+ Add new item</a>					

As can be seen, the list has been populated with another person and the workflow is already running.

**Step 4.** Go to the library “DemoLib” to check the newly created form.

<input type="checkbox"/>	Type	Name	Modified
		New_Doc_John <small>NEW</small>	8/1/2013 3:54 AM
<a href="#">+ Add document</a>			

It has been created with the name we set in the workflow: “New\_Doc\_” + Item’s name.

**Step 5.** Open the form and check its data.

<input type="button" value="Save"/>		<input type="button" value="Print"/>	
persName	<input type="text" value="John"/>		
persLName	<input type="text" value="Olsen"/>		
persAge	<input type="text" value="38"/>		
persTelephone	<input type="text" value="123-456-789"/>		
fromWorkflow	<input type="text" value="yes"/>		

As can be seen, all data is in its place and “fromWorkflow” field is populated with value “yes”.

**Step 5 (\*Alternative).** Open the form and check its data.

Save	Print
persName	John
persLName	Olsen
persAge	38
persTelephone	123-456-789
fromWorkflow	13

As can be seen, all data is in its place and “fromWorkflow” field is populated with value “13”, which is “ID” of currently populated list item.