



# PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

## Form prepopulating during deployment

Product: PDF Share Forms Enterprise for SharePoint 2010

### Contents

Information storage .....	2
Static (Acro) form .....	5
Dynamic (XFA) form .....	10
Supplement to the guide .....	14



This guide describes how to make form prepopulated with data from SharePoint list during form deployment process. Such approach allows getting all necessary data once without additional queries and data loading during each form opening.

To achieve that we will use two forms:

First form (Information Storage form) is used to get information from list for further use in second form;

Second Form is form where we will copy information from first form to create form with prepopulated data.

This list (“Employees”) is used for demonstration:

Name	Surname	Age
John	Smith	45
Kate	Argent	25
Kim	Tomphson	33
James	Torn	54
Natalie	Arin	37
Alex	Ront	43
Alan	Green	40
Till	Abel	31
Elen	Adler	44
Clark	Lorens	64
Adams	Martin	29
Lisa	Edwards	28

+ Add new item

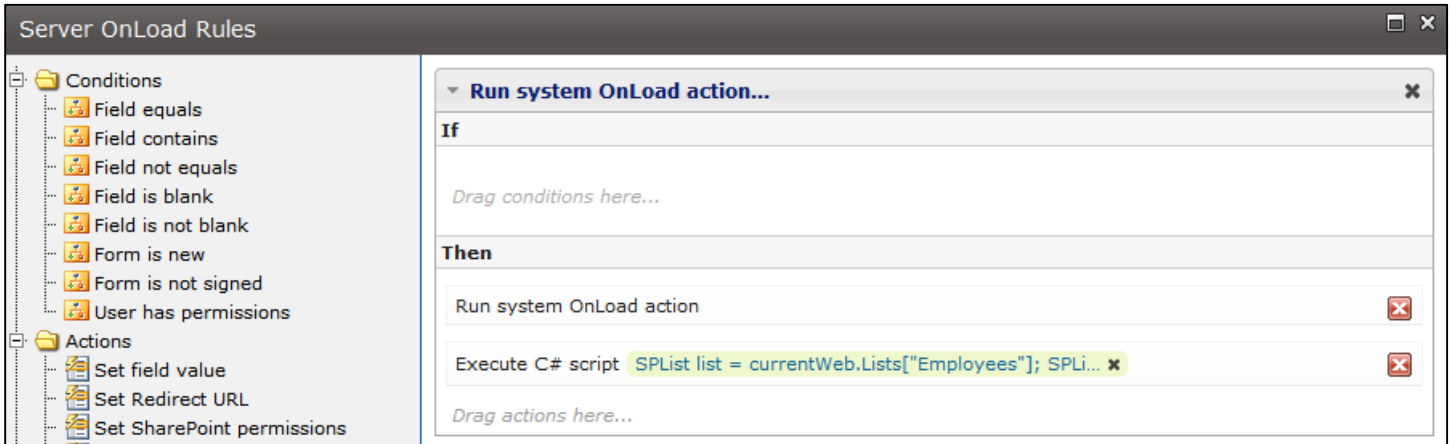
## Information storage

### Step 1. Template

The screenshot shows a form template interface. On the left is a sidebar with a 'Hierarchy' view containing 'Page 1' and 'InformationStorage'. The main content area features the PDF SHARE FORMS logo and the text 'Online, Offline, OnDemand' and 'PDF forms and SharePoint are better together'. Below this is a large, empty rectangular field labeled 'Information Storage'.

This template has only one multiline field – Information Storage. This field store data from SharePoint list.

**Step 2.** Add Form Load script. Navigate to **PDF Form Tools → Developer → Form Load**



**Script:**

```
//list must be specified for proper case
SPList list = currentWeb.Lists["Employees"];
//with red are marked list column names
SPListItemCollection itemsEmp = list.GetItems("Name", "Surname", "Age");
List<Dictionary<string,string>> users = new List<Dictionary<string,string>>();
foreach(SPListItem item in itemsEmp)
{
    Dictionary<string,string> newItem = new Dictionary<string,string>();
    newItem["Name"] = item["Name"].ToString();
    newItem["Surname"] = item["Surname"].ToString();
    newItem["Age"] = item["Age"].ToString();
    users.Add(newItem);
}
JavaScriptSerializer serializer = new JavaScriptSerializer();
data.resolveNode("InformationStorage").value = serializer.Serialize(users);
```

This script loads data from SharePoint list and store item structure in **Informations Storage** field.



**Step 3.** Deploy template, navigate to library and create this form

The screenshot shows a PDF form template with the PDF SHARE FORMS logo and the text "Online, Offline, OnDemand PDF forms and SharePoint are better together". A field labeled "Information Storage" is populated with the following JSON data:

```
[{"Name": "John", "Surname": "Smith", "Age": "45"}, {"Name": "Kate", "Surname": "Argent", "Age": "25"}, {"Name": "Kim", "Surname": "Tompson", "Age": "33"}, {"Name": "James", "Surname": "Torn", "Age": "54"}, {"Name": "Natalie", "Surname": "Arin", "Age": "37"}, {"Name": "Alex", "Surname": "Ront", "Age": "43"}, {"Name": "Alan", "Surname": "Green", "Age": "40"}, {"Name": "Till", "Surname": "Abel", "Age": "31"}, {"Name": "Elen", "Surname": "Adler", "Age": "44"}, {"Name": "Clark", "Surname": "Lorens", "Age": "64"}, {"Name": "Adams", "Surname": "Martin", "Age": "29"}, {"Name": "Lisa", "Surname": "Edwards", "Age": "28"}]
```

Field is populated with values from SharePoint list.

**Step 4.** Download this form

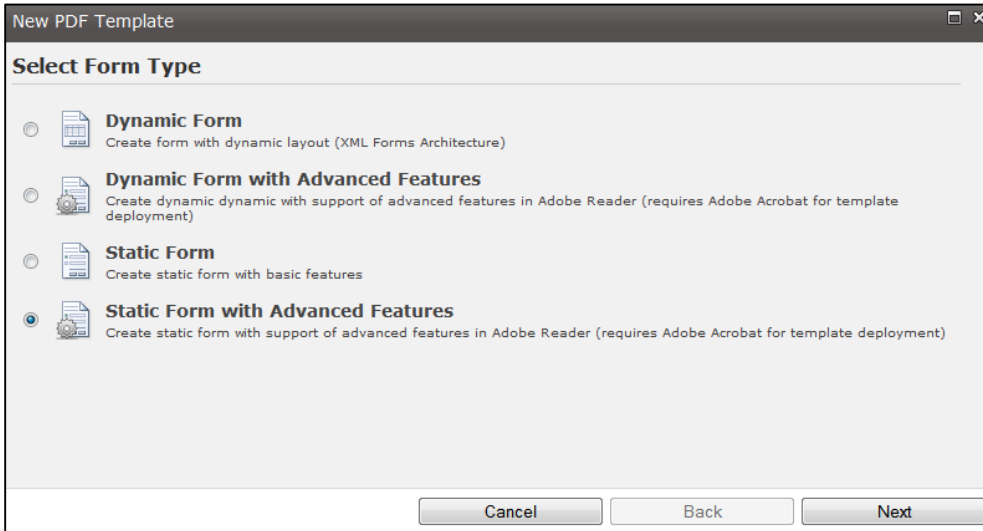
The screenshot shows a PDF viewer interface with a toolbar at the top. The main content area displays the same PDF form template as in Step 3, with the "Information Storage" field populated with the same JSON data:

```
[{"Name": "John", "Surname": "Smith", "Age": "45"}, {"Name": "Kate", "Surname": "Argent", "Age": "25"}, {"Name": "Kim", "Surname": "Tompson", "Age": "33"}, {"Name": "James", "Surname": "Torn", "Age": "54"}, {"Name": "Natalie", "Surname": "Arin", "Age": "37"}, {"Name": "Alex", "Surname": "Ront", "Age": "43"}, {"Name": "Alan", "Surname": "Green", "Age": "40"}, {"Name": "Till", "Surname": "Abel", "Age": "31"}, {"Name": "Elen", "Surname": "Adler", "Age": "44"}, {"Name": "Clark", "Surname": "Lorens", "Age": "64"}, {"Name": "Adams", "Surname": "Martin", "Age": "29"}, {"Name": "Lisa", "Surname": "Edwards", "Age": "28"}]
```

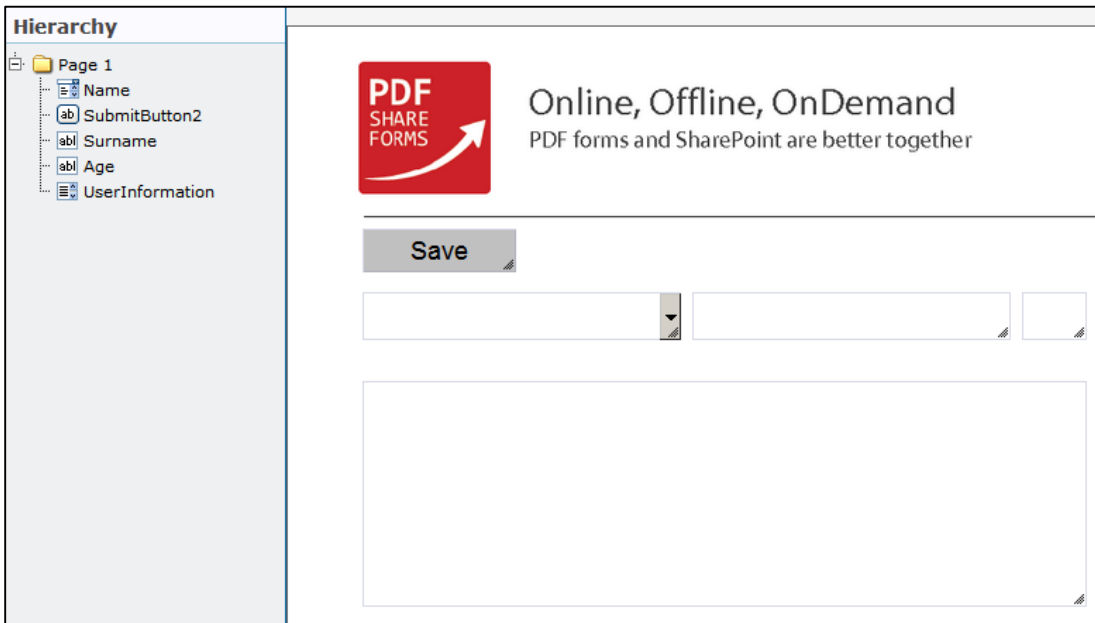
This information will be used in deployment process of the PDF form described in following guide parts.

## Static (Acro) form

### Step 1. Create new Static Form with Advanced Features



### Step 2. Template



This template has following controls:

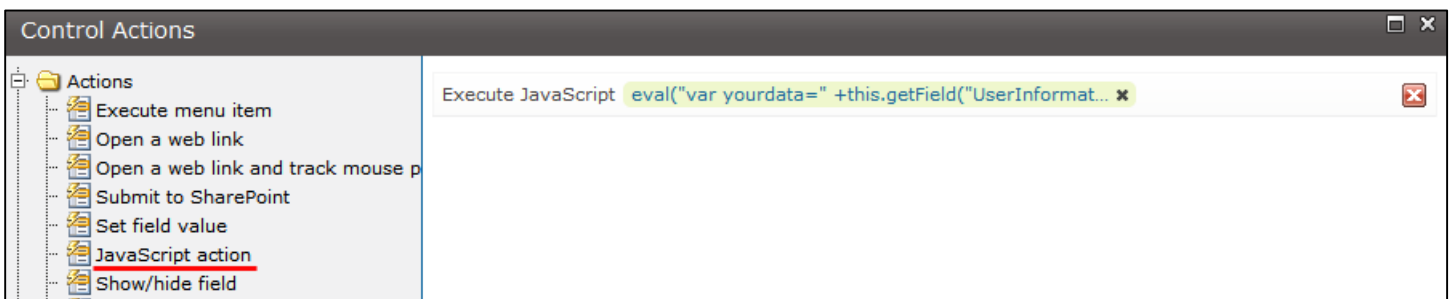
- Name – dropdown field which will populate “Surname” and “Age” fields;
- Surname and Age – text fields that will be populated according on dropdown’s value;
- UserInformation – multiline text field, will store items from SharePoint list;
- Submit button – submit to SharePoint library.

**Step 3.** Add Global JavaScript. Navigate to **PDF Form Tools → Developer → Global JavaScript**

```
var val = this.getField("Name").value;
eval("var yourdata=" + this.getField("UserInformation").value + ";");
var dropdown = this.getField("Name");
var i = 0;
for(i = 0;i< yourdata.length; i++ ){
    dropdown.insertItemAt(yourdata[i]["Name"],yourdata[i]["Name"]);
}
this.getField("Name").value = val;
```

This script populates “Name” dropdown with information that is stored in “UserInformation” field.

**Step 4.** Add field Exit script to “Name” dropdown. Choose “Name” dropdown and then navigate to **PDF Form Tools → Properties → Actions → Exit – add JavaScript action**



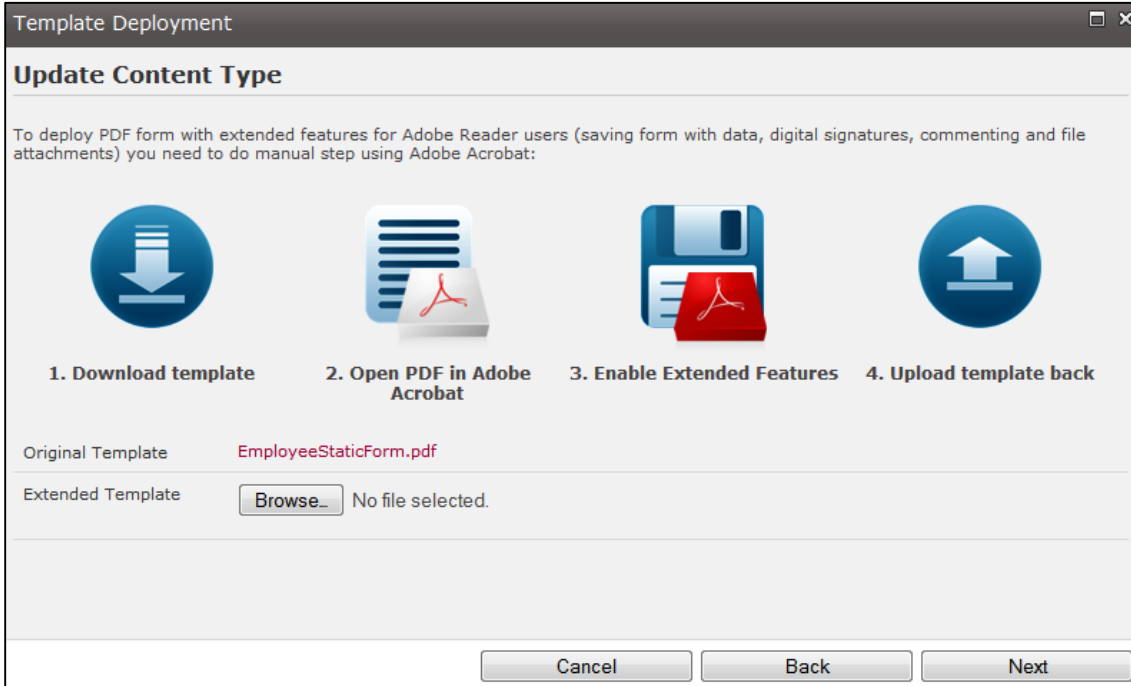
**Script:**

```
eval("var yourdata=" +this.getField("UserInformation").value+ ";");
var user = this.getField("Name").value;
var i = 0;
for (i = 0;i<yourdata.length; i++){
    if(yourdata[i]["Name"] == user){
        this.getField("Surname").value = yourdata[i]["Surname"];
        this.getField("Age").value = yourdata[i]["Age"];
    }
}
```

This script compare dropdown’s value with information in “Information Storage” field; when it finds similar – it populates “Surname” and “Age” fields with associated information.



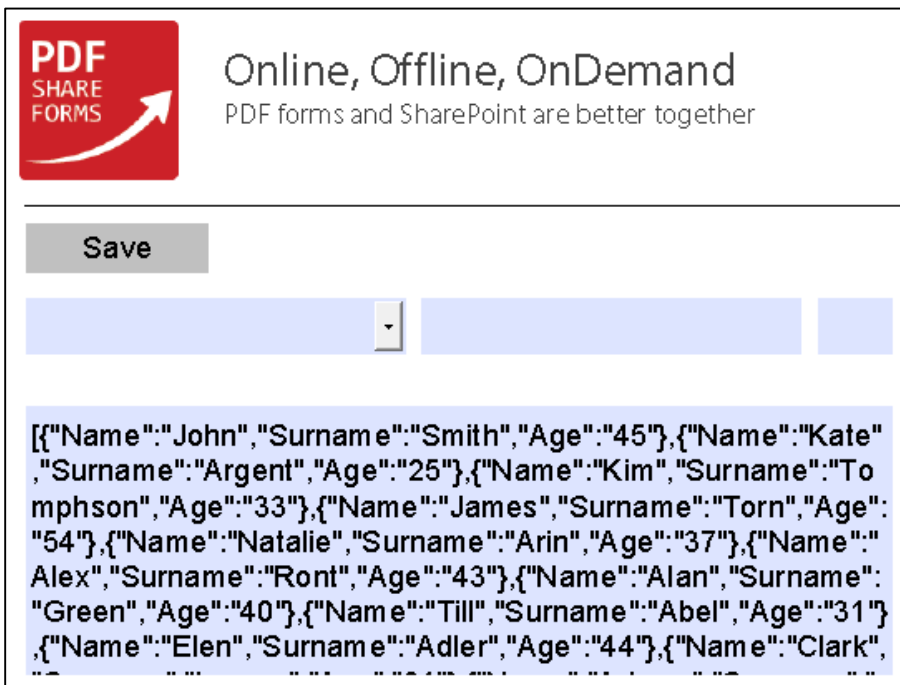
**Step 5. Deploy template**



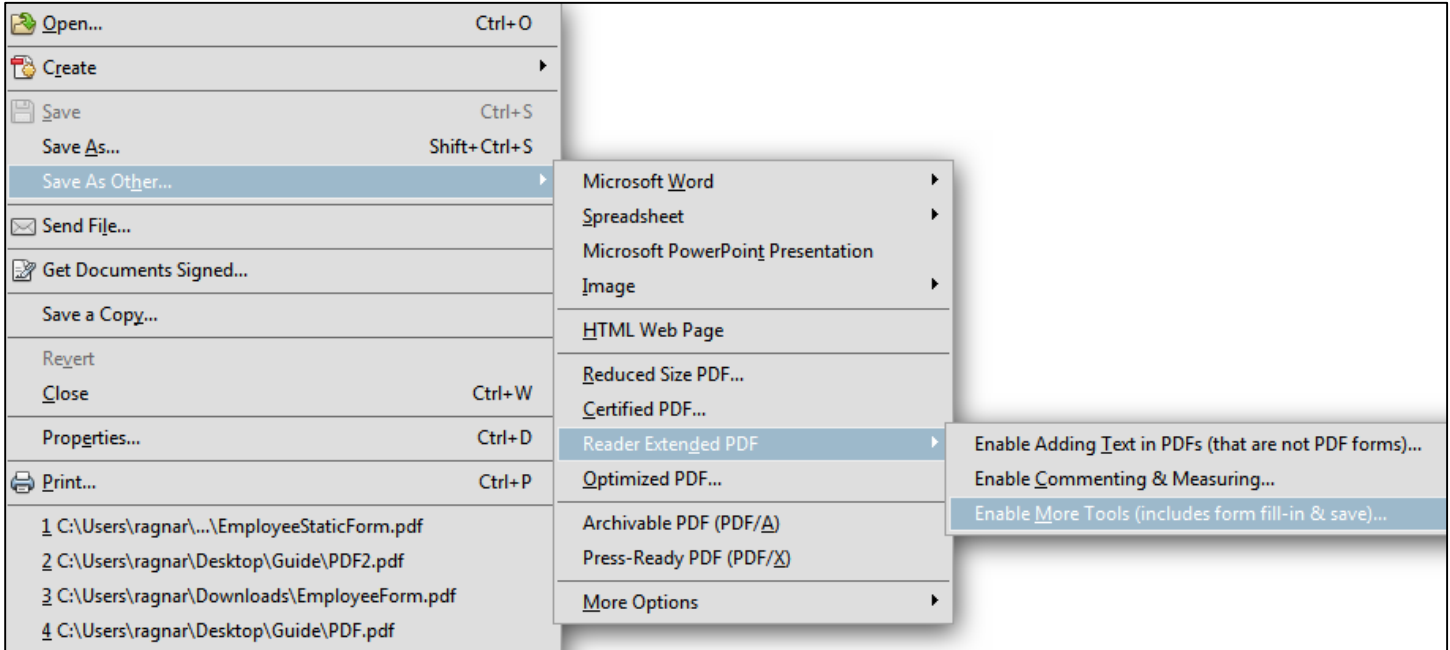
Download this form to activate advanced features and add data.

**Step 5.1** Open this form in Adobe Acrobat

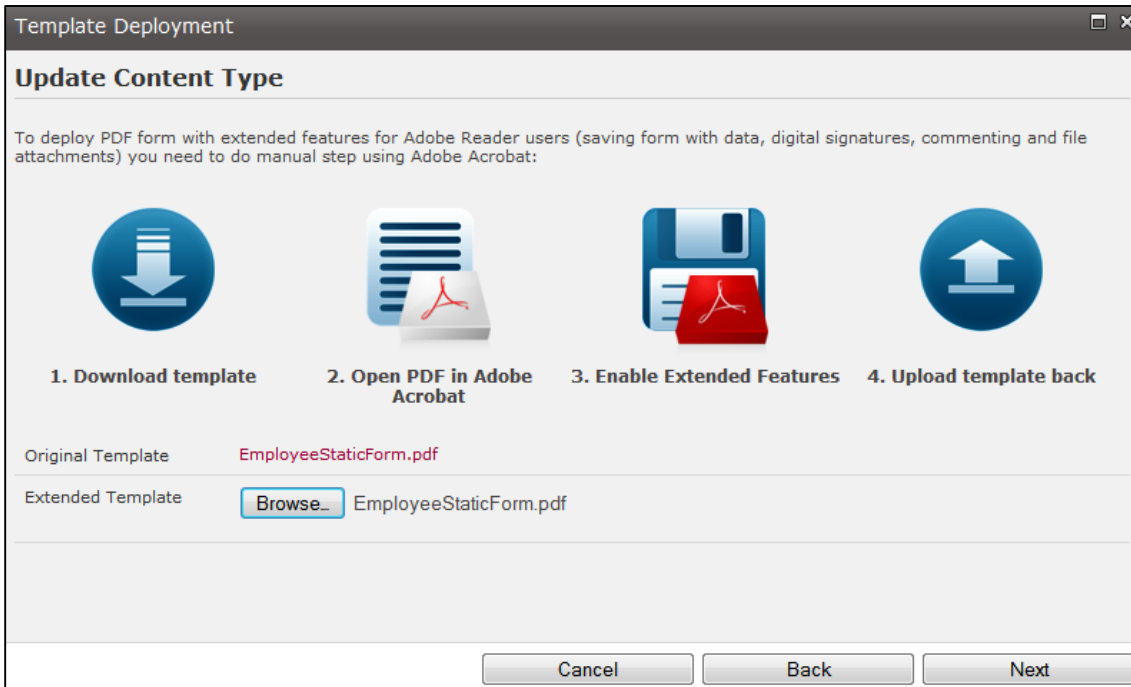
**Step 5.2** Open form that was downloaded in [step 4](#) of “Information Storage” part and copy information from it to form that is used in deployment process



**Step 5.3** Save form with advanced features – **File** → **Save as other** → **Reader Extended PDF** → **Enable More Tools**



**Step 5.4** Upload this form to finish deployment process




Click “Next” and finish deployment.





**Step 6.** Open form in a library and choose any value in the dropdown



Online, Offline, OnDemand  
PDF forms and SharePoint are better together

---

Save

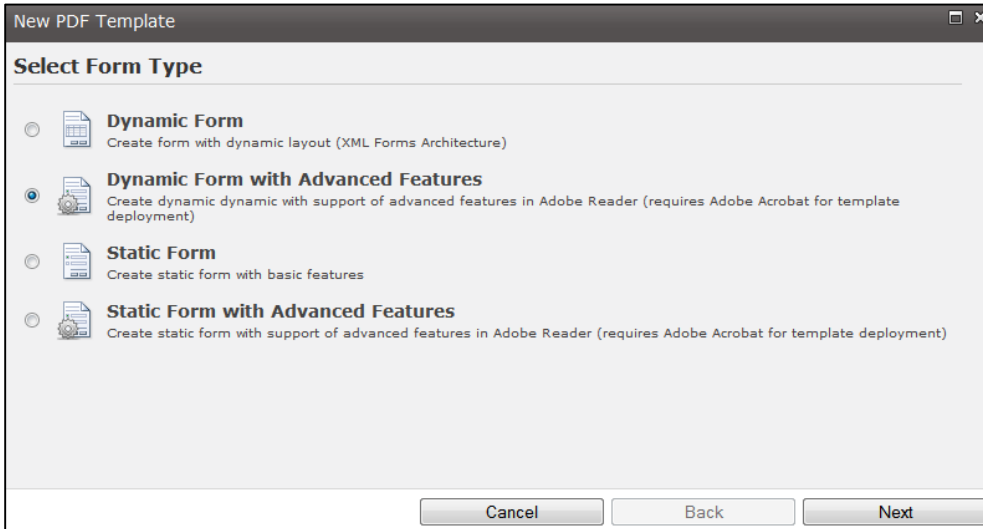
Elen	-	Adler	44
------	---	-------	----

```
[{"Name":"John","Surname":"Smith","Age":"45"}, {"Name":"Kate","Surname":"Argent","Age":"25"}, {"Name":"Kim","Surname":"Tomphson","Age":"33"}, {"Name":"James","Surname":"Torn","Age":"54"}, {"Name":"Natalie","Surname":"Arin","Age":"37"}, {"Name":"Alex","Surname":"Ront","Age":"43"}, {"Name":"Alan","Surname":"Green","Age":"40"}, {"Name":"Till","Surname":"Abel","Age":"31"}, {"Name":"Elen","Surname":"Adler","Age":"44"}, {"Name":"Clark","Surname":"Lorens","Age":"64"}, {"Name":"Adams","Surname":"Adams","Age":"35"}]
```

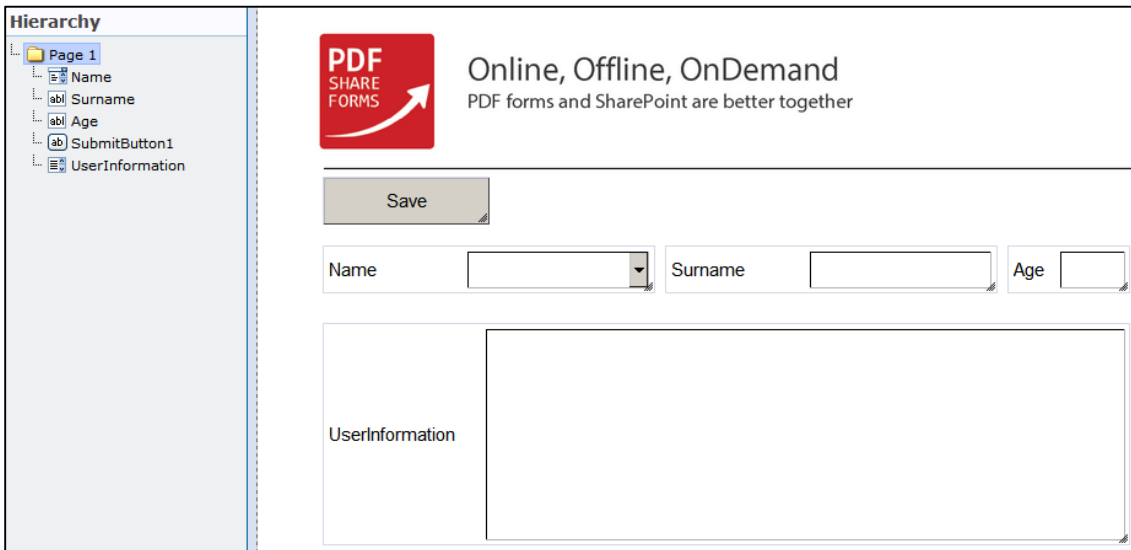
**NOTE:** Multiline text field “User Information” can be set to hidden.

## Dynamic (XFA) form

### Step 1. Create Dynamic Form with Advanced Features



### Step 2. Template



This template has following controls:

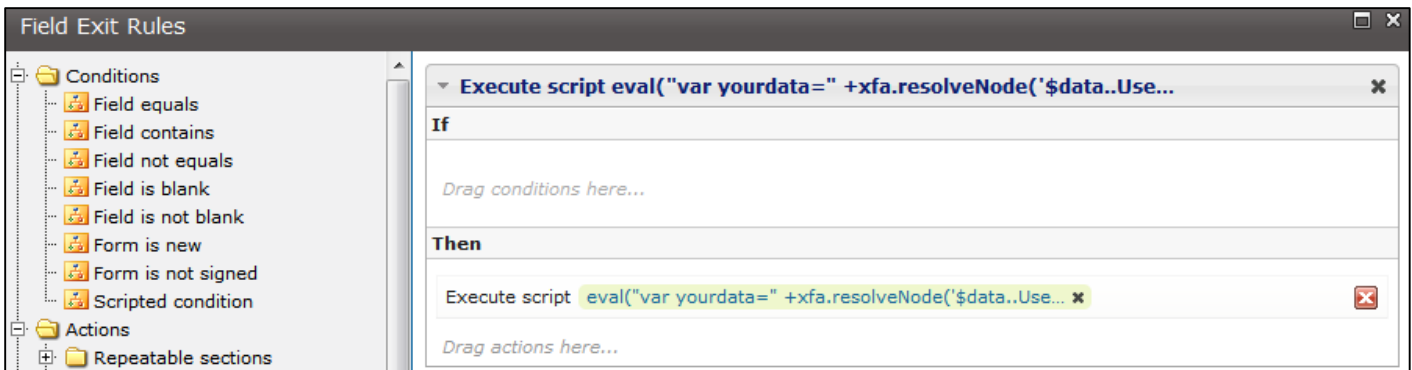
- ☑ Name – dropdown field which will populate “Surname” and “Age” fields;
- ☑ Surname and Age – text fields that will be populated according on dropdown’s value;
- ☑ UserInformation – multiline text field, will store items from SharePoint list;
- ☑ Submit button – submit to SharePoint library.

**Step 3.** Add On Form Ready Script. Navigate to **PDF Form Tools → Developer → On Form Ready Script**

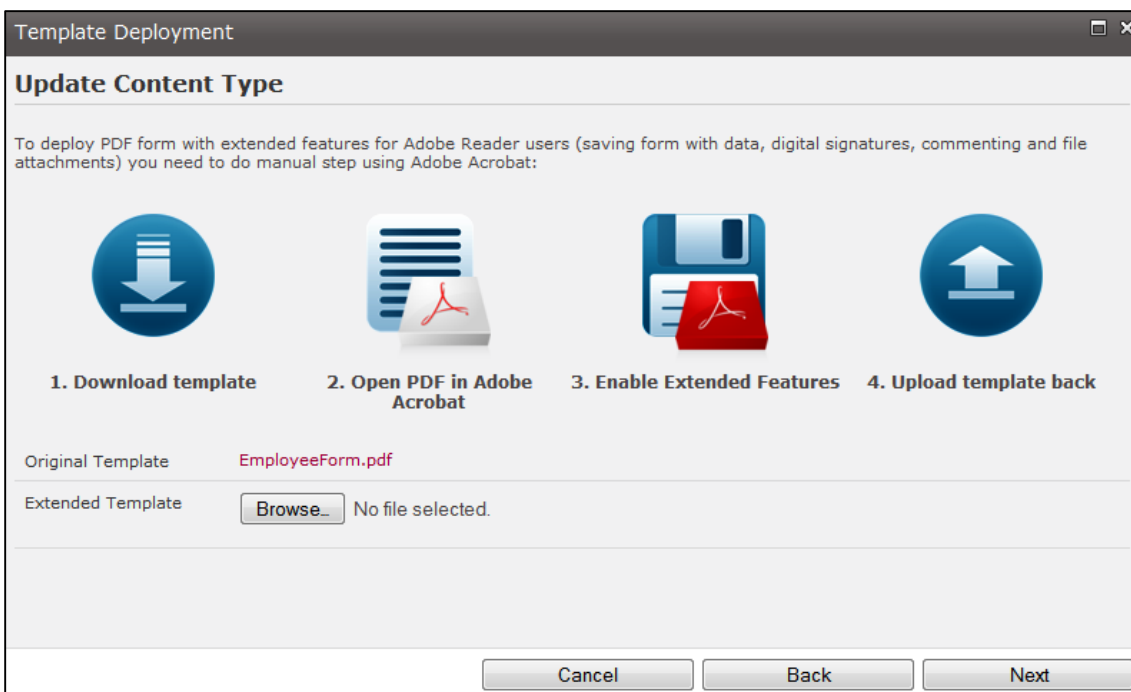
```
eval("var yourdata=" +xfa.resolveNode('$data..UserInformation').value+ ");");
var dropdown = xfa.resolveNode('$form.Root.Default..Name');
dropdown.clearItems();
var i = 0;
for(i = 0;i< yourdata.length; i++ ){
    dropdown.addItem(yourdata[i]["Name"],yourdata[i]["Name"]);
}
```

This script populates “Name” dropdown with information that is stored in “UserInformation” field.

**Step 4.** Add Field Exit script to “Name” dropdown. Choose “Name” dropdown and then navigate to **PDF Form Tools → Properties → Actions → Field Exit** – add **Execute script** action.



**Step 5.** Deploy template



Download this form to activate advanced features and add data.



**Step 5.1** Open this form in Adobe Acrobat

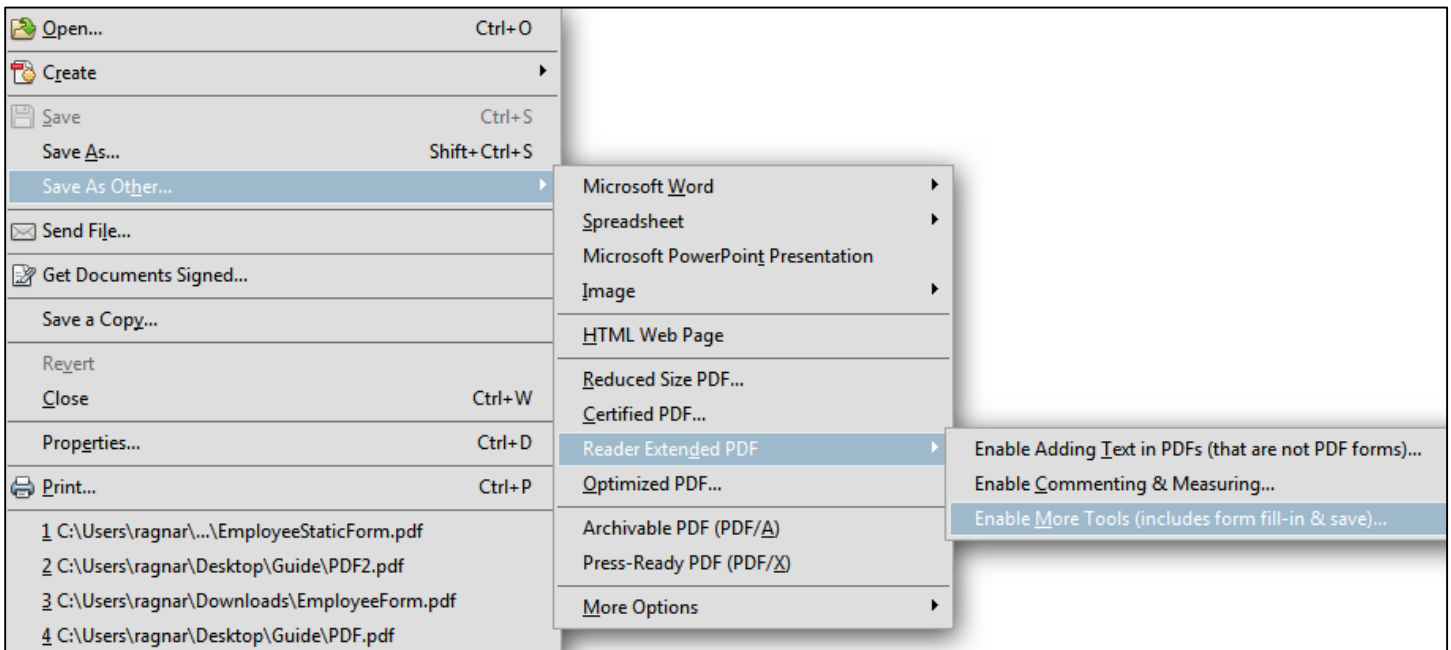
**Step 5.2** Open form that was downloaded in [step 4](#) of “Information Storage” part and copy information from it to form that is used in deployment process

The screenshot shows a PDF form with the following elements:

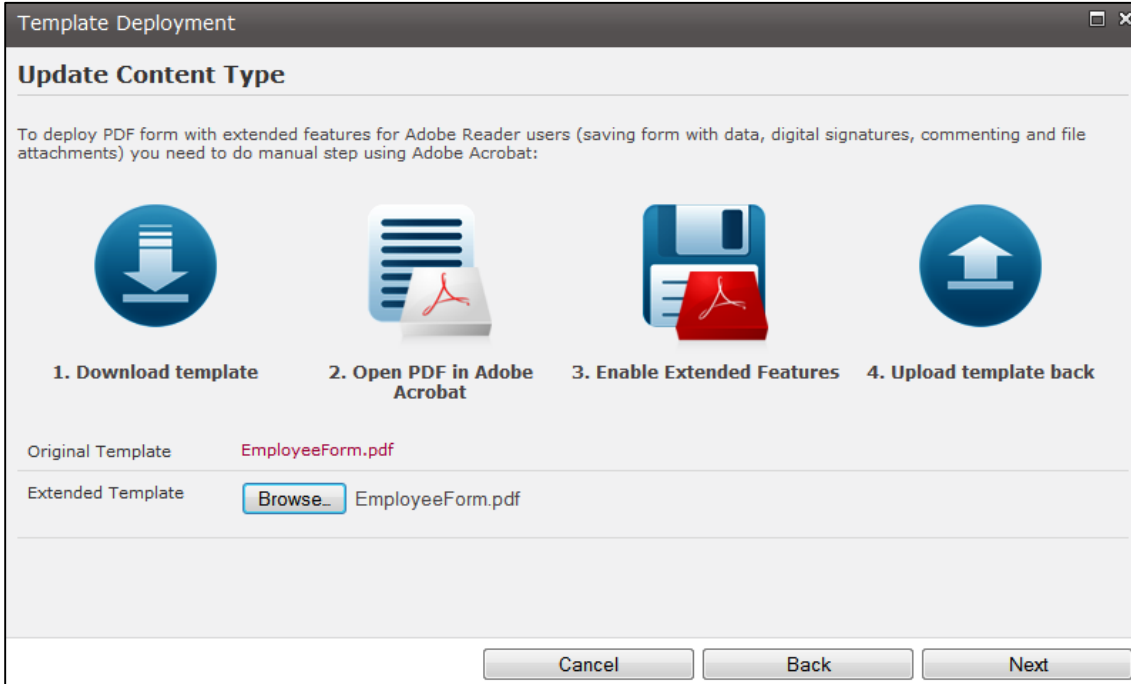
- Title:** Online, Offline, OnDemand
- Subtitle:** PDF forms and SharePoint are better together
- Buttons:** A 'Save' button is located at the top left.
- Form Fields:** There are three input fields labeled 'Name', 'Surname', and 'Age'.
- UserInfo Field:** A large text area labeled 'UserInfo' contains a JSON array of employee data:
 

```
[{"Name": "John", "Surname": "Smith", "Age": "45"},
  {"Name": "Kate", "Surname": "Argent", "Age": "25"},
  {"Name": "Kim", "Surname": "Tompson", "Age": "33"},
  {"Name": "James", "Surname": "Torn", "Age": "54"},
  {"Name": "Natalie", "Surname": "Arin", "Age": "37"},
  {"Name": "Alex", "Surname": "Ront", "Age": "43"},
  {"Name": "Alan", "Surname": "Green", "Age": "40"},
  {"Name": "Till", "Surname": "Abel", "Age": "31"},
  {"Name": "Elen", "Surname": "Adler", "Age": "44"}]
```

**Step 5.3** Save form with advanced features – **File** → **Save as other** → **Reader Extended PDF** → **Enable More Tools**

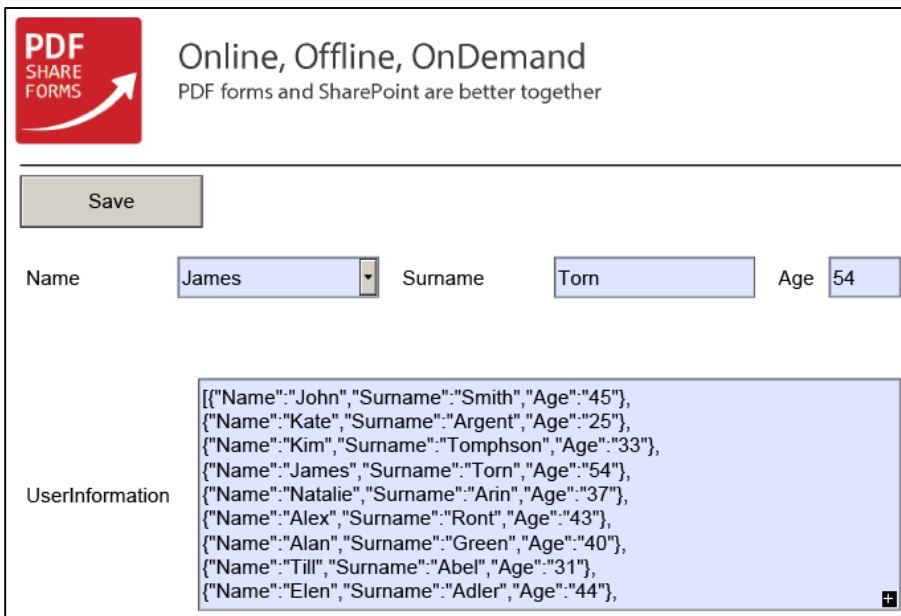


**Step 5.4** Upload this form to finish deployment process



Click "Next" and finish deployment.

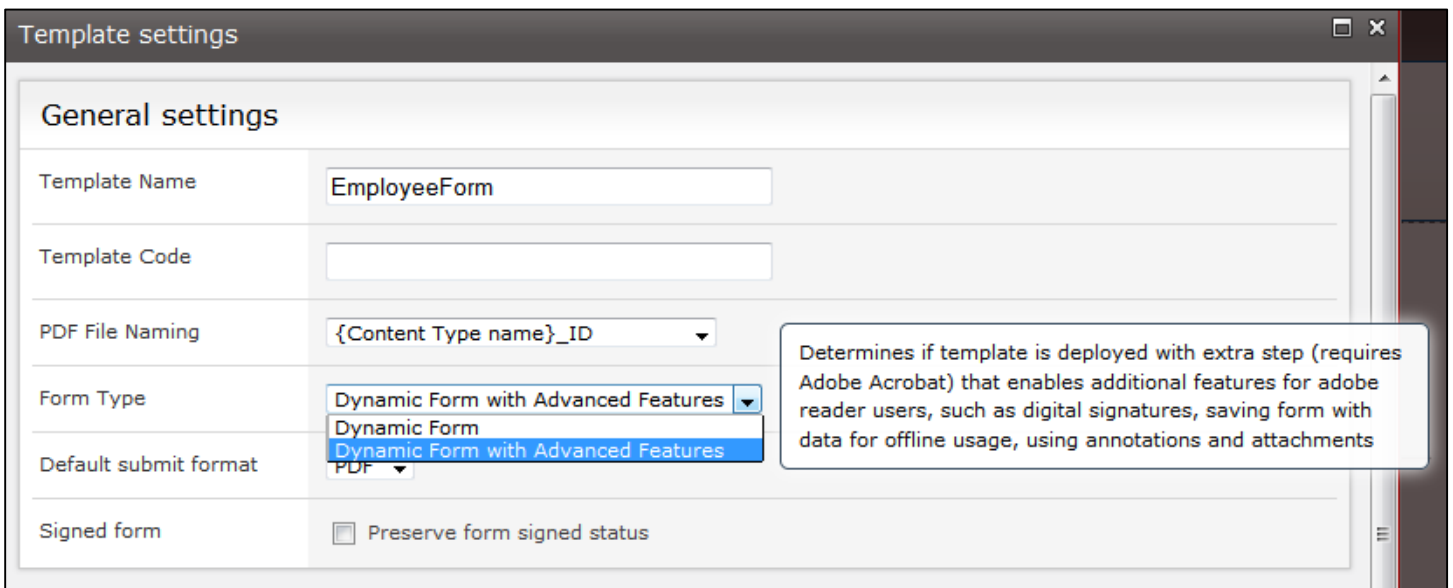
**Step 6.** Open form in a library and choose any value in the dropdown



## Supplement to the guide

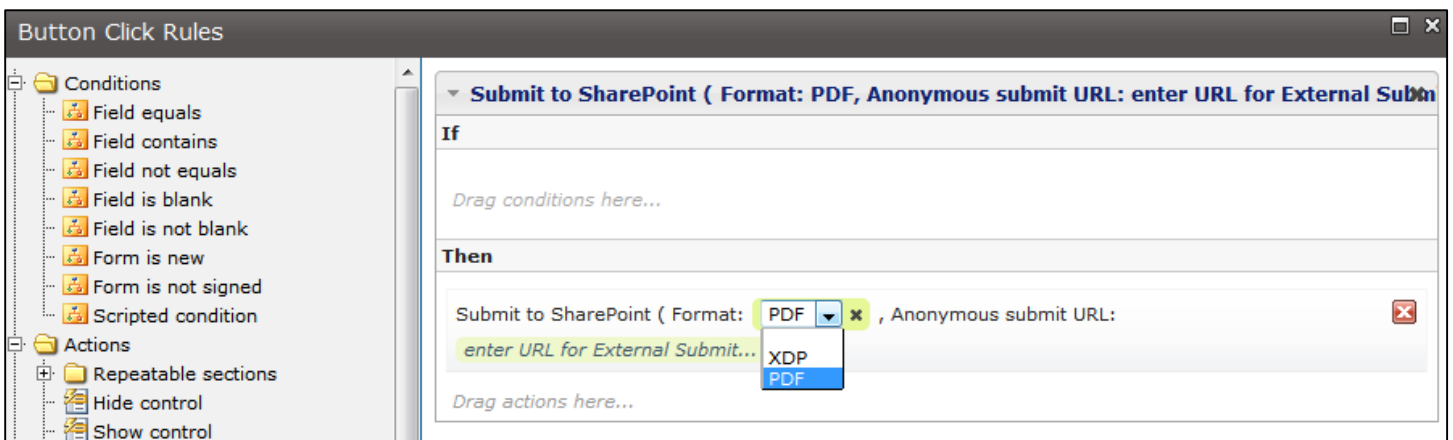
If simple, without advanced features template is being used and there is need to change it for the same but with advanced features – follow this steps:

**Step 1.** Open template setting. Navigate to **PDF Share Forms → Home → Setting** and change form type from “Dynamic Form” to “Dynamic Form with Advanced Features” (“Static Form” to “Static Form with Advanced Features” in case with static form)



The screenshot shows the 'Template settings' dialog box. Under 'General settings', the 'Form Type' dropdown is open, showing 'Dynamic Form with Advanced Features' selected. A tooltip explains that this option determines if the template is deployed with an extra step (requiring Adobe Acrobat) for additional features like digital signatures and attachments. Other settings include 'Template Name' (EmployeeForm), 'PDF File Naming' ({Content Type name}\_ID), and a checkbox for 'Signed form' (Preserve form signed status).

**Step 2.** Change submit format in “Submit” button. Choose “Submit” button and navigate to **PDF Form Tools → Properties → Actions** then change format from “XDP” to “PDF” (rom “FDF” to “PDF” in case with static form)



The screenshot shows the 'Button Click Rules' dialog box. The 'Then' section contains an action 'Submit to SharePoint ( Format: PDF, Anonymous submit URL: enter URL for External Submit...)'. A dropdown menu is open over the 'Format' field, showing 'PDF' selected, with 'XDP' and 'PDF' as options.

**Step 3.** Save and deploy.