



PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

Populating fields from list

dependent on selected value in dropdown

Preparing list

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

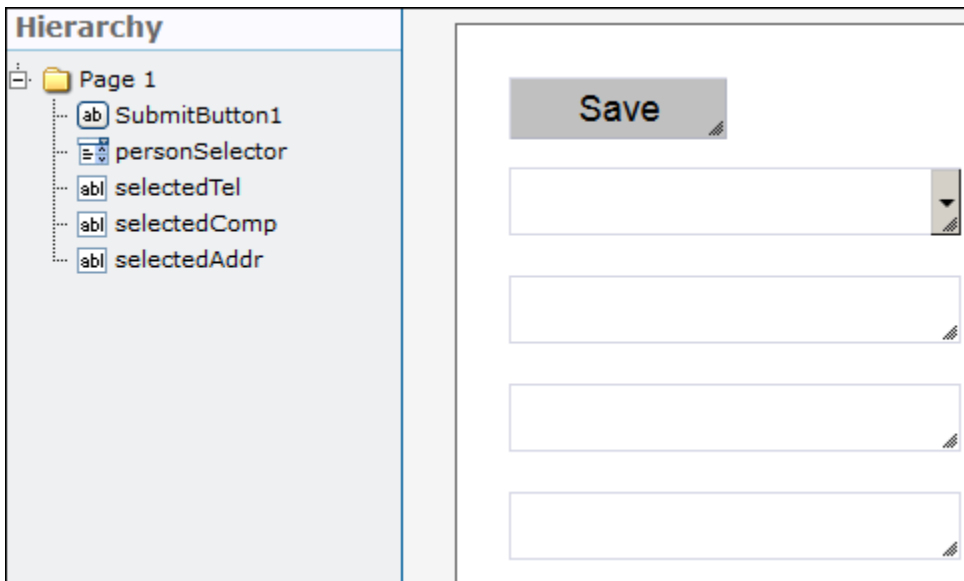
<input type="checkbox"/>	Name	Telephone	Company	Address
	Andrew	222-333-444	New company	Earth
	David	4534332	OpMetals	Latvia
	Tom	26349846	OpMetals	Latvia
	Peter	123456	Some company	some address
	Dan	11223344	OpMetals	Latvia
	John	111-222-333-44	New company	Earth
	Mike	66-55-44-33	Some company	some address

[+ Add new item](#)

„Name“ will be populated inside proppdown field; rest values will populate text fields.

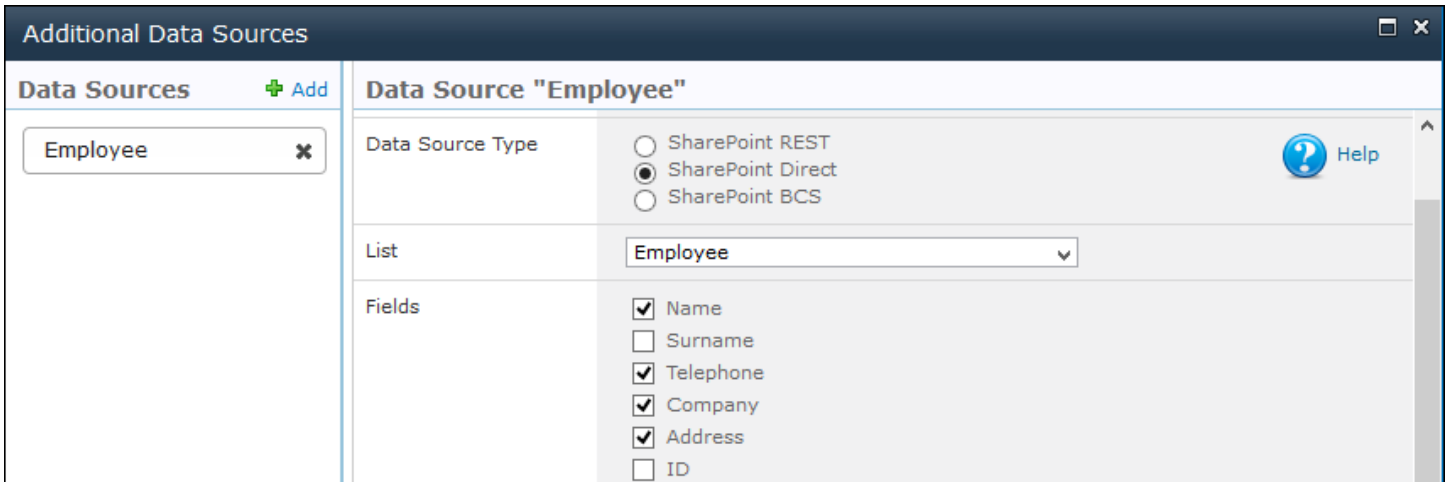
Static Acro forms

As an example, we will use the following template:



This template has one submit button, dropdown field (personSelector) and text fields which will be populated with values dependent on value selected in dropdown field.

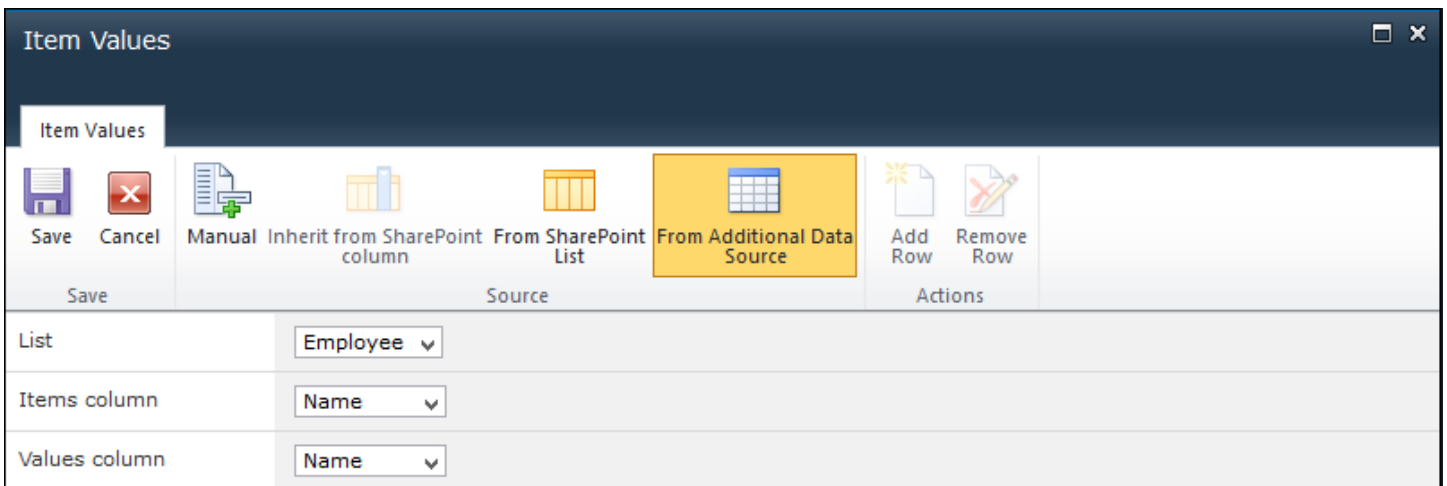
Step 1. Create data source for “Employee” list: *PDF Form Tools* → *Developer* → *Data Sources*.



Select values which will be used inside the form: Name, Telephone, Company and Address.

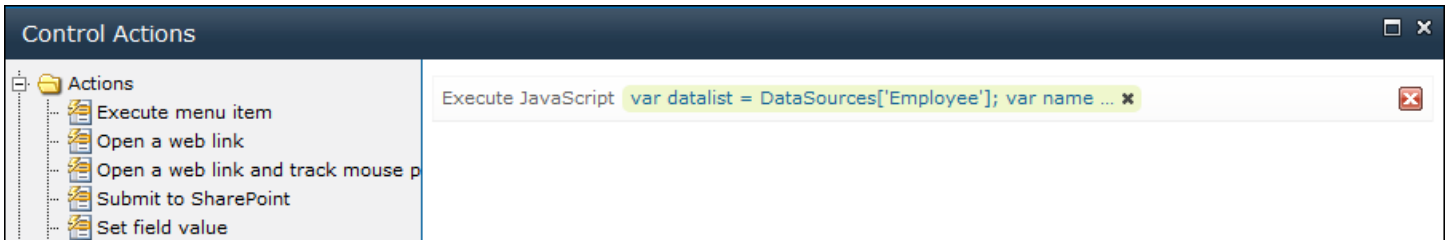
Step 1.2. Name this data source (“Employee” in our case) and save it.

Step 2. Add “Employee” data source to dropdown field “personSelector”: *PDF Form Tools* → *Properties* → *Item Values*.



- Select “From Additional Data Source” source, as “List” use has created data source “Employee” in Step 1;
- For “Items column” and “Values column” select “Name”, so that we can select person name in the dropdown field.

Step 3. Add action to dropdown “Blur” event: *PDF Form Tools* → *Properties* → *Actions* → *Blur*.



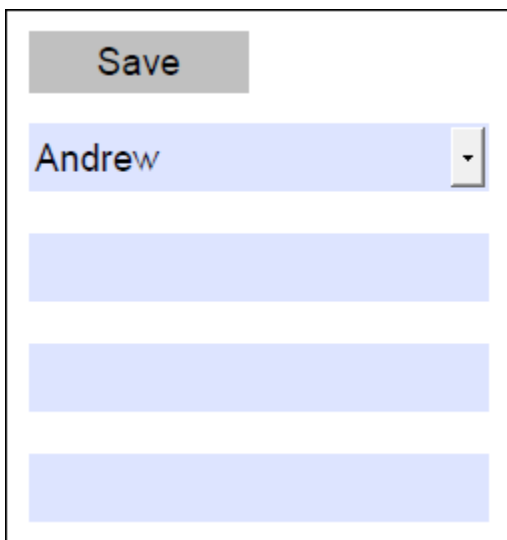
Step 4. Add “JavaScript action” script.

Code:

```
var datalist = DataSources['Employee'];
var name = this.getField("personSelector").value;
var i = 0;
for(i = 0;i< DataSources['Employee'].length; i++ ){
    if(datalist[i]['Name'] == name){
        this.getField("selectedTel").value = datalist[i]['Telephone'];
        this.getField("selectedComp").value = datalist[i]['Company'];
        this.getField("selectedAddr").value = datalist[i]['Address'];
    }
}
```

This code will populate text fields (selectedTel for telephone, selectedComp for compny name, selectedAddr for address) dependent on value currently selected in dropdown field (personSelector).

Step 5. Deploy the template and create a form.





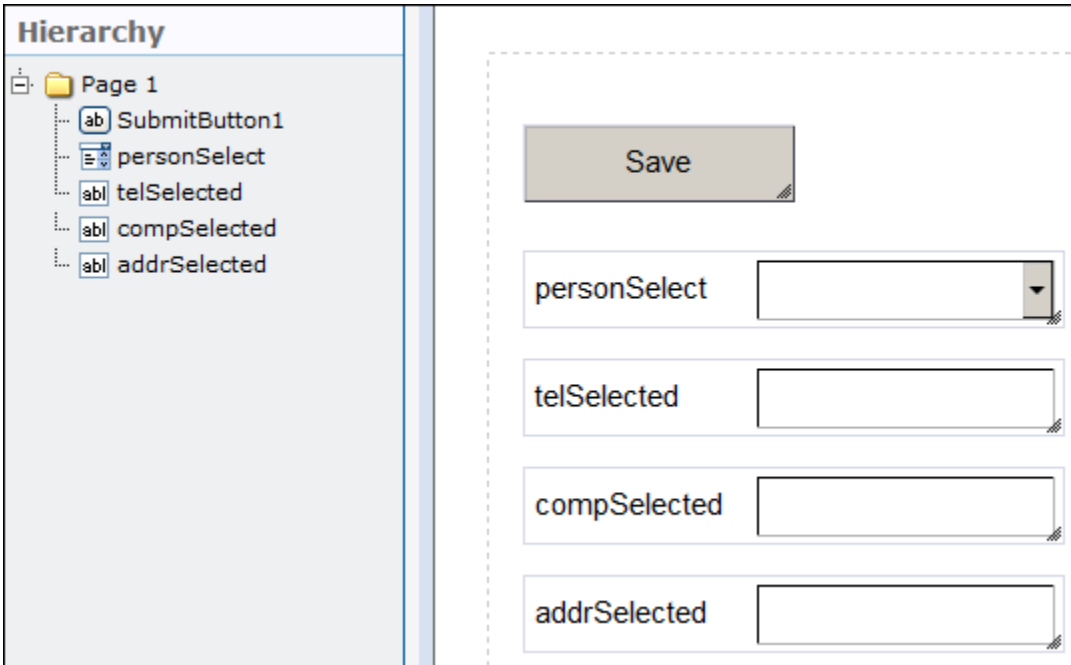
Step 5.1. Select some value in dropdown field.

A screenshot of a form interface. At the top is a grey button labeled "Save". Below it is a dropdown menu with "Mike" selected. Underneath the dropdown are four text input fields, each containing a value: "66-55-44-33", "Some company", and "some address". The fourth field is empty.

As can be seen, data related to value selected in dropdown field is populated to text fields.

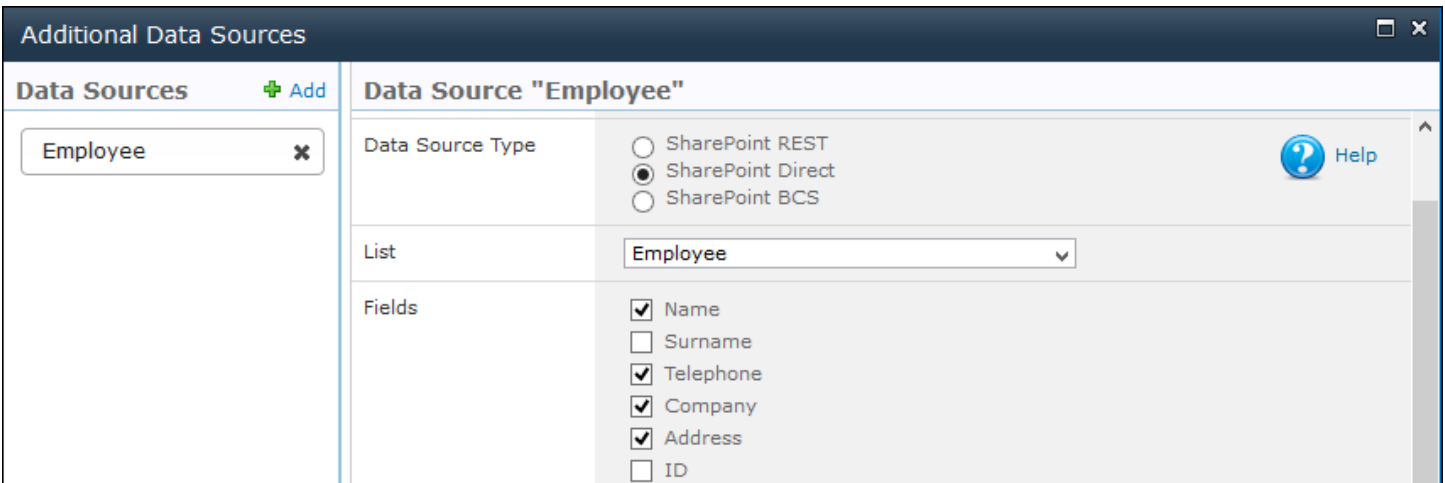
Dynamic XFA forms

As an example, we will use the following template:



This template has one submit button, dropdown field (personSelect) and text fields which will be populated with values depending on value selected in dropdown field.

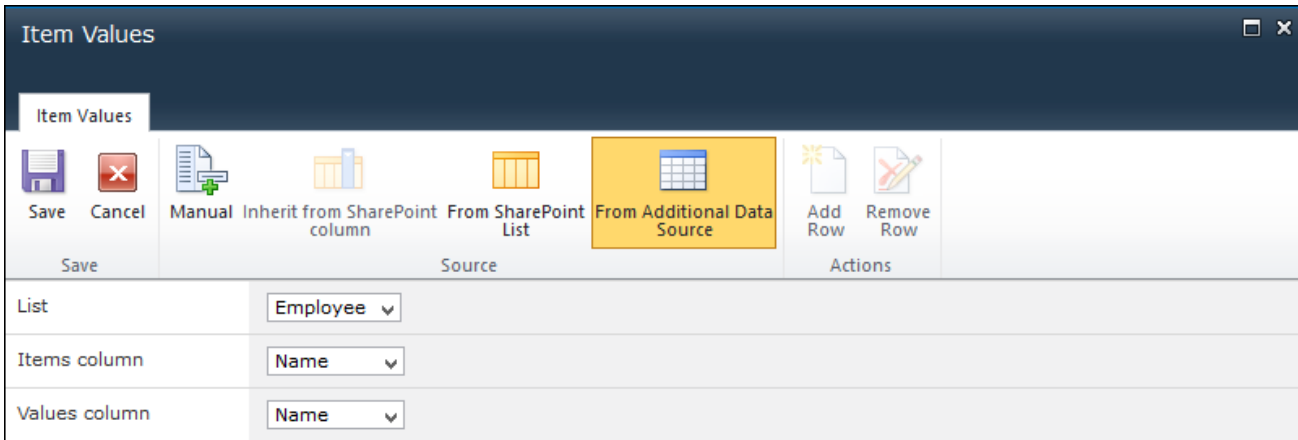
Step 1. Create data source for “Employee” list: *PDF Form Tools* → *Developer* → *Data Sources*.



Select values which will be used inside the form: Name, Telephone, Company and Address.

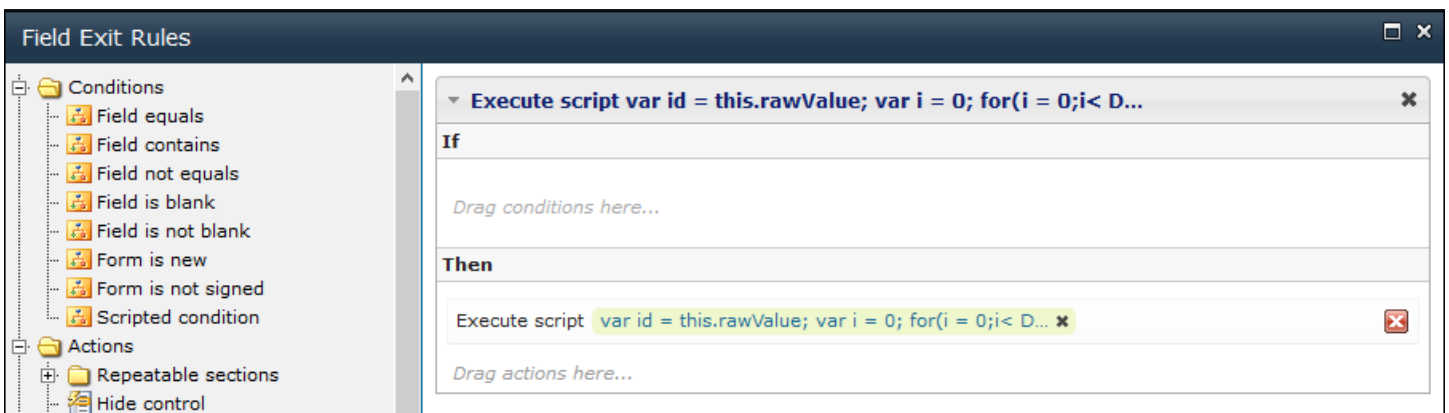
Step 1.2. Name this data source (“Employee” in our case) and save it.

Step 2. Add “Employee” data source to dropdown field “personSelector”: *PDF Form Tools* → *Properties* → *Item Values*.



- Select “From Additional Data Source” source, as “List” use has created data source “Employee” in Step 1;
- For “Items column” and “Values column” select “Name”, so that we can select person name in the dropdown field.

Step 3. Add action to dropdown “Field Exit” event: *PDF Form Tools* → *Properties* → *Field Exit*.



Step 4. Add “Execute script” action.

Code:

```
var id = this.rawValue;
var i = 0;
for(i = 0; i < DataSources.d.Employee.length; i++) {
    if(DataSources.d.Employee[i]["Name"] == id) {
        xfa.resolveNode("$data..telSelected").value =
DataSources.d.Employee[i]["Telephone"];
        xfa.resolveNode("$data..compSelected").value =
DataSources.d.Employee[i]["Company"];
        xfa.resolveNode("$data..addrSelected").value =
DataSources.d.Employee[i]["Address"];
    }
}
```

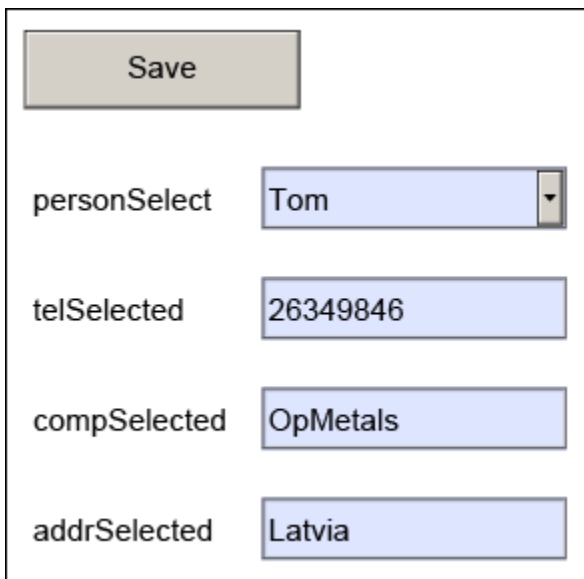
This code will populate text fields (“telSelected” for telephone, “compSelected” for company name, “addrSelected” for address) dependent on value currently selected in dropdown field (this.rawValue).

Step 5. Deploy the template and create a form.



A screenshot of a form interface. At the top left is a grey button labeled "Save". Below it are four text input fields, each with a label to its left: "personSelect", "telSelected", "compSelected", and "addrSelected". All four text fields are currently empty.

Step 5.1. Select some value in dropdown field.



A screenshot of the same form interface as above. The "personSelect" dropdown field now contains the text "Tom". The "telSelected" text field contains "26349846", the "compSelected" text field contains "OpMetals", and the "addrSelected" text field contains "Latvia". The "Save" button remains at the top left.

As can be seen, data related to value selected in dropdown field is populated to text fields.