



# PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

## Populating dropdown from the upper level list

Data sources use current web for searching for available lists. This guide is describing how to populate dropdown fields if form is deployed on the sub site, but source list is located on the root level.

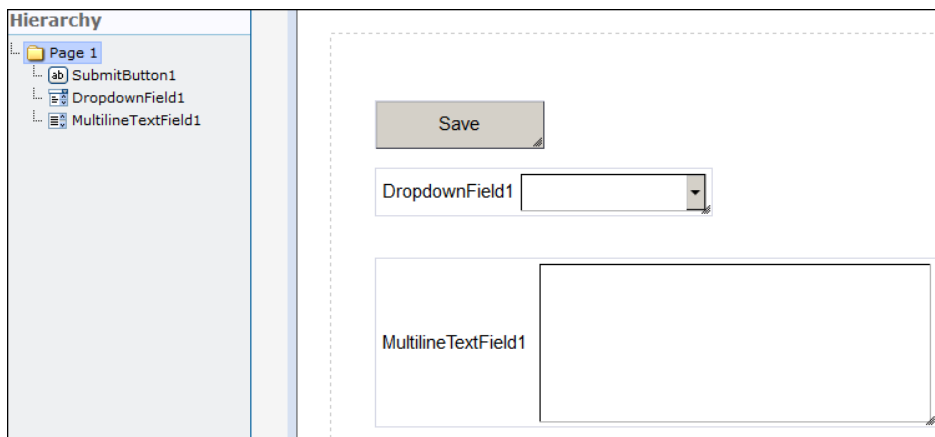
Site structure used in this example:

This example is based on the following site structure:

- <http://yoursite.com/sites/labs1/PDF%20Forms%20Templates> - PDF templates library;
- <http://yoursite.com/sites/labs1/Lists/Companies> - source list;
- <http://yoursite.com/sites/labs1/site2/subsite/SubSiteLib> – target library on site2 sub site;

## Dynamic XFA Forms

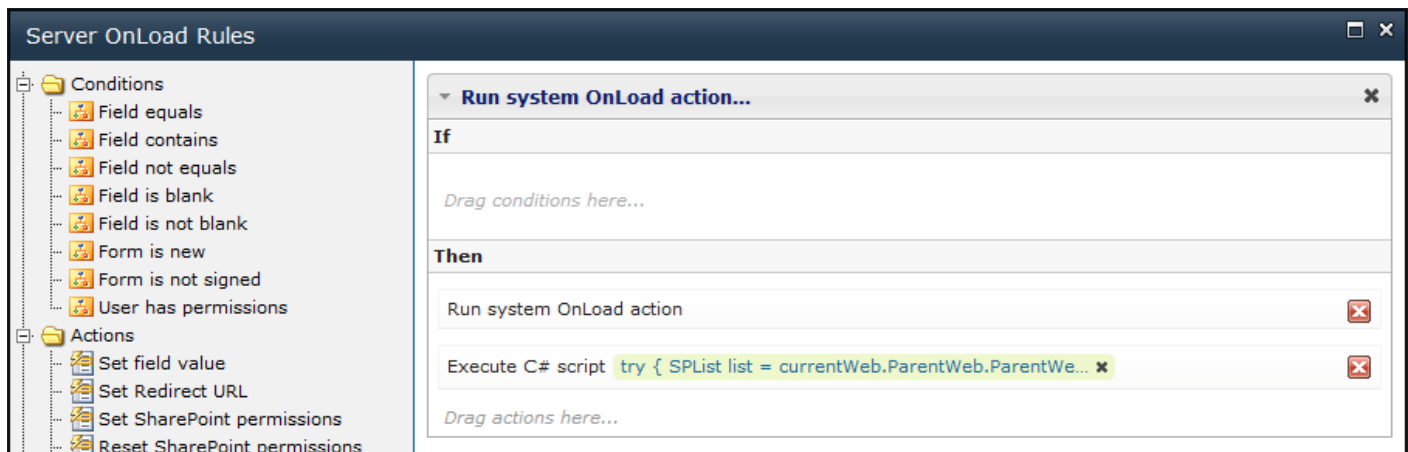
### Step 1. Preparing a template



This template has “SubmitButton1”, “DropdownField1” and “MultilineTextField1” which will be used to store list data.

**NOTE:** “MultilineTextField1” must be set to “hidden”.

### Step 2. Adding script to “Form Load” action (PDF form Tools → Developer → Form Load):



Add “Execute script” action under “Run system OnLoad action”.

Code:

```
SPList list = currentWeb.Site.RootWeb.Lists["Companies"];
SPListItemCollection items = list.GetItems("Title");
List<string> myList = new List<string>();
foreach(SPListItem item in items)
{
    myList.Add(item["Title"].ToString());
}
//serialize data
JavaScriptSerializer serializer = new JavaScriptSerializer();
data.resolveNode("MultilineTextField1").value = serializer.Serialize(myList);
```

This script will load data from “Title” of the SharePoint list “Companies” to “MultilineTextField1”. To get list was used following line of code: “SPList list = currentWeb.Site.RootWeb.Lists[“Companies”];”. It is getting required list by descending from sub site (which is “currentWeb”) to the root using “Site.RootWeb”.

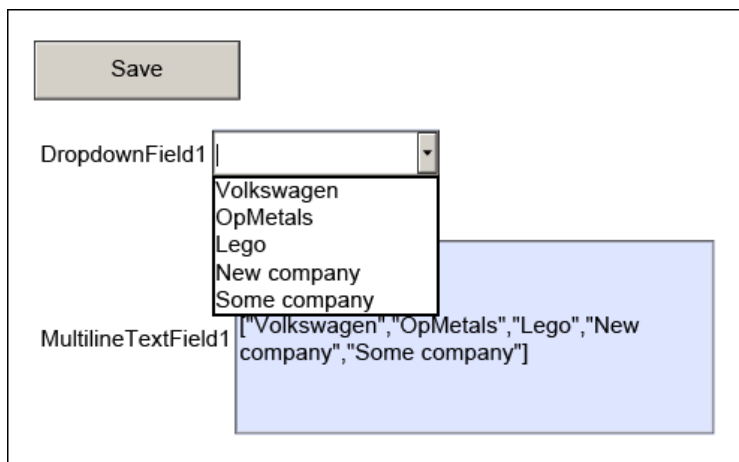
**Step 3.** Add “On Form Ready Script” (PDF form Tools > Developer > On Form Ready Script):

Code:

```
eval("var yourdata=" + xfa.resolveNode('$data..MultilineTextField1').value + " ");
//Populate data from Companies list to dropdown field if it is empty
var dropdown = xfa.resolveNode('$form.Root.Default..DropdownField1');
var i = 0;
if (dropdown.length == 0){
    for(i = 0;i< yourdata.length; i++ ){
        dropdown.addItem(yourdata[i],yourdata[i]);
    }
}
```

This script will populate “DropdownField1” with data from “MultilineTextField1”.

**Step 4.** Save and deploy template to the sub site, create new form

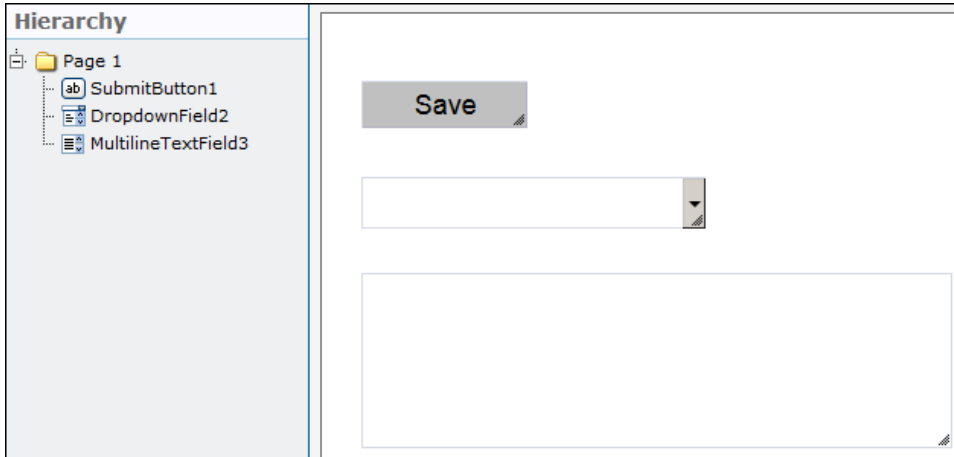


“DropdownField1” has all data from “Title” column of the “Companies” list.

**NOTE:** “MultilineTextField1” must be set to “hidden” – it is shown here only for demonstration purposes.

## Static AcroForms

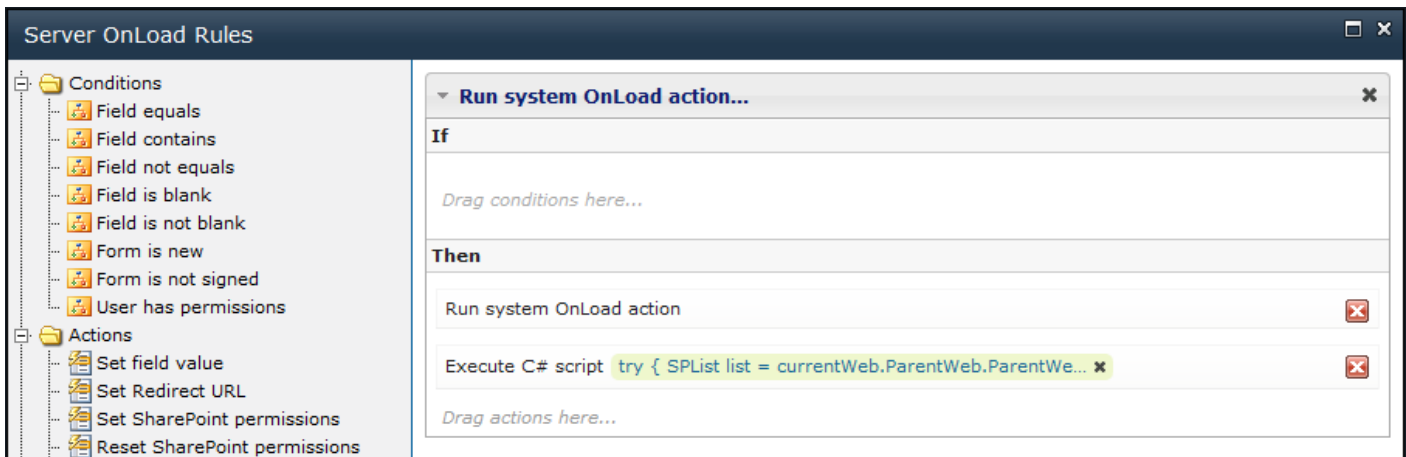
### Step 1. Preparing a template



This template has “SubmitButton1”, “DropdownField2” and “MultilineTextField3” which will be used to store list data.

**NOTE:** “MultilineTextField3” must be set to “hidden”.

### Step 2. Adding script to “Form Load” action (PDF form Tools → Developer → Form Load):



Add “Execute script” action under “Run system OnLoad action”.

Code:

```
SPList list = currentWeb.Site.RootWeb.Lists["Companies"];
SPListItemCollection items = list.GetItems("Title");
List<string> myList = new List<string>();
foreach(SPListItem item in items)
{
    myList.Add(item["Title"].ToString());
}
//serialize data
JavaScriptSerializer serializer = new JavaScriptSerializer();
data.resolveNode("MultilineTextField3").value = serializer.Serialize(myList);
```

This script will load data from “Title” of the SharePoint list “Companies” to “MultilineTextField3”. To get list was used following line of code: “SPList list = currentWeb.Site.RootWeb.Lists[“Companies”];”. It is getting required list by descending from sub site (which is “currentWeb”) to the root using “Site.RootWeb”.

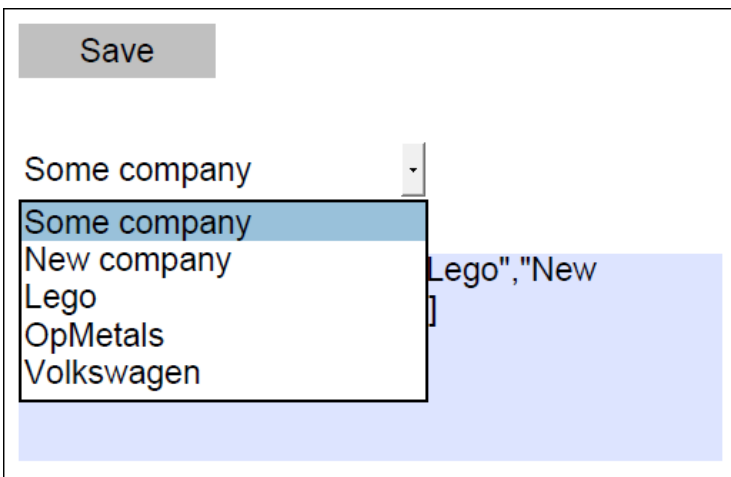
**Step 3.** Add “Global JavaScript” (PDF form Tools > Developer > Global JavaScript):

Code:

```
eval("var yourdata=" +this.getField("MultilineTextField3").value+ " ");
//Populate data from Companies list to dropdown field if it is empty
var dropdown = this.getField("DropdownField2");
var i = 0;
if (dropdown.numItems == 0){
  for(i = 0;i< yourdata.length; i++ ){
    dropdown.insertItemAt(yourdata[i]);
  }
}
```

This script will populate “DropdownField2” with data from “MultilineTextField3”.

**Step 4.** Save and deploy template to the sub site, create new form



“DropdownField2” has all data from “Title” column of the “Companies” list.

**NOTE:** “MultilineTextField3” must be set to “hidden” – it is shown here only for demonstration purposes.