



PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

Cascade People Picker

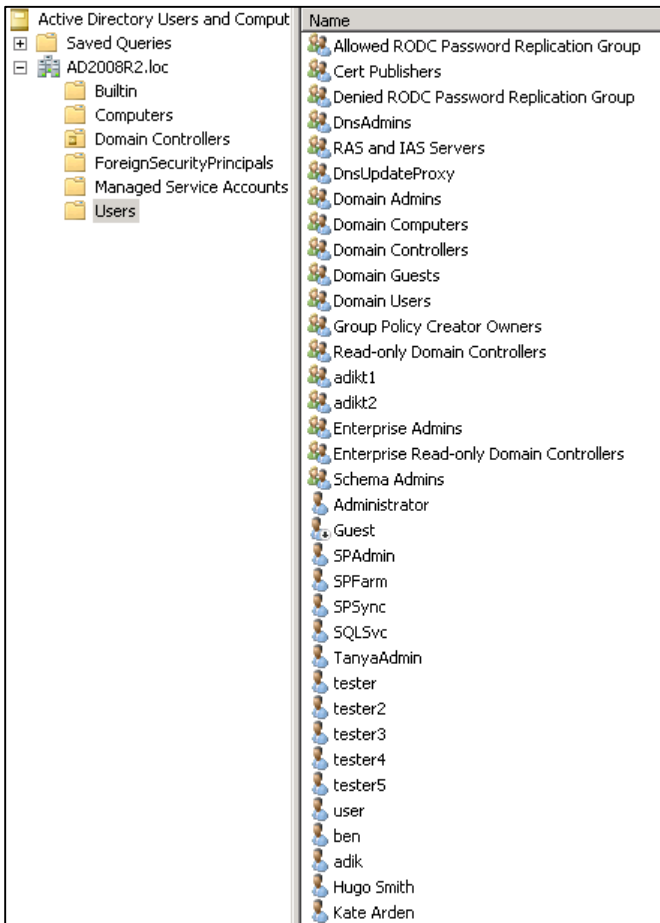
Product: PDF Share Forms Enterprise for SharePoint 2010

Contents

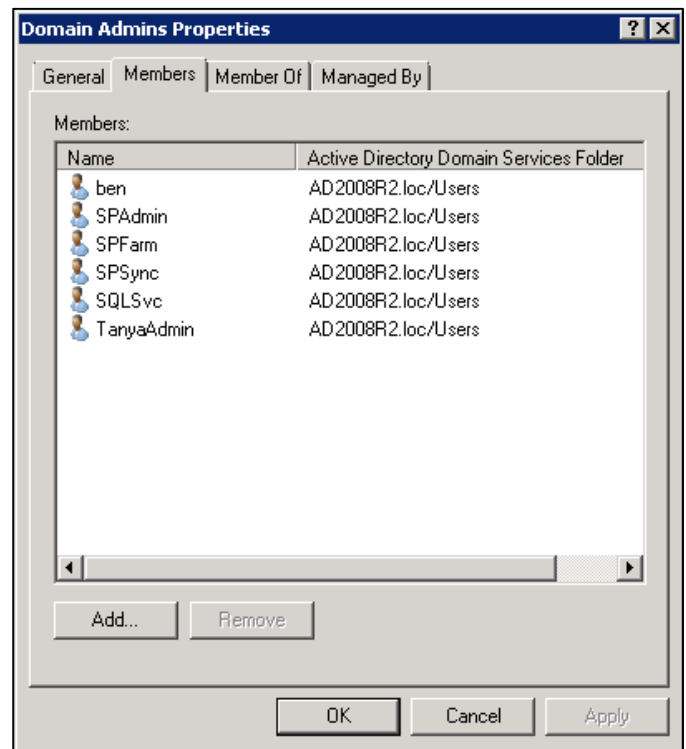
Active Directory	2
Assemblies	3
Template	4
Runtime.....	6

This guide describes how to create cascade people pickers with data from Active Directory. First, in people picker, we choose a group, and then people picker will be refreshed with users that belong to selected group.

Active Directory



This Active Directory's folder (Users) is used in the guide. There are stored users and groups. Each user can be a member of several groups. For example, "Domain Admins" group contain six users





Assemblies

Ensure that proper assemblies are added:

Go to Central Administration → PDF Forms Settings → Scripting assemblies

Add **System.DirectoryServices** assembly

Approved Assemblies

```

Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
Microsoft.Office.Server, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
Microsoft.Office.Server.UserProfiles, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
System.Web, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
Microsoft.SharePoint.Client, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
Microsoft.SharePoint.Client.Runtime, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
System.Xml.Linq, Version=3.5.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
System.Xml, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
Microsoft.BusinessData, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
System.DirectoryServices, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
    
```

Add **System.DurectoryServices** namespace

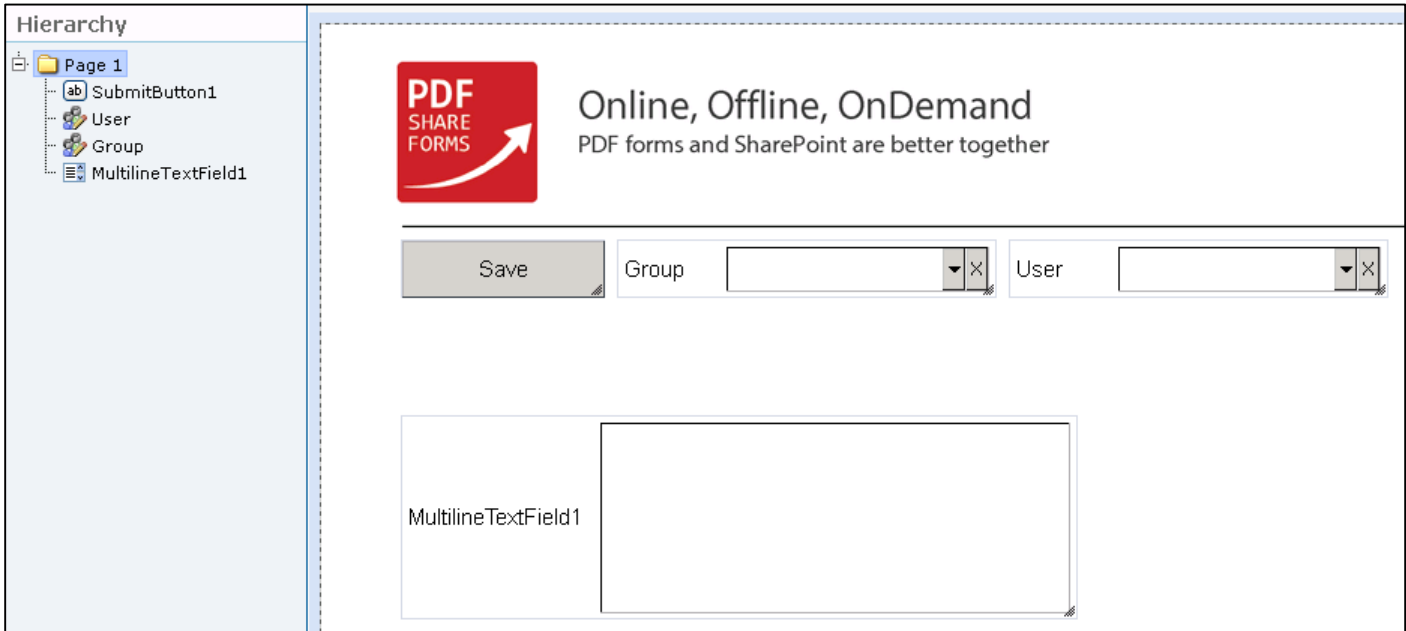
Approved Namespaces

```

Microsoft.SharePoint
System.Collections.Generic
System.Linq
System.Xml.Linq
System.Text
Microsoft.SharePoint.BusinessData.SharedService
Microsoft.BusinessData.MetadataModel
Microsoft.SharePoint.Administration
Microsoft.SharePoint.BusinessData.Runtime
Microsoft.BusinessData.Runtime
System.Data
System.Collections
Microsoft.SharePoint.BusinessData.Infrastructure
PDFForms.Services.BCS
System.Web.Script.Serialization
SPClient=Microsoft.SharePoint.Client
PDFForms.Services.ServerSideScripting
Microsoft.Office.Server
Microsoft.Office.Server.UserProfiles
System.DirectoryServices
    
```

Template

Step 1. Prepare template



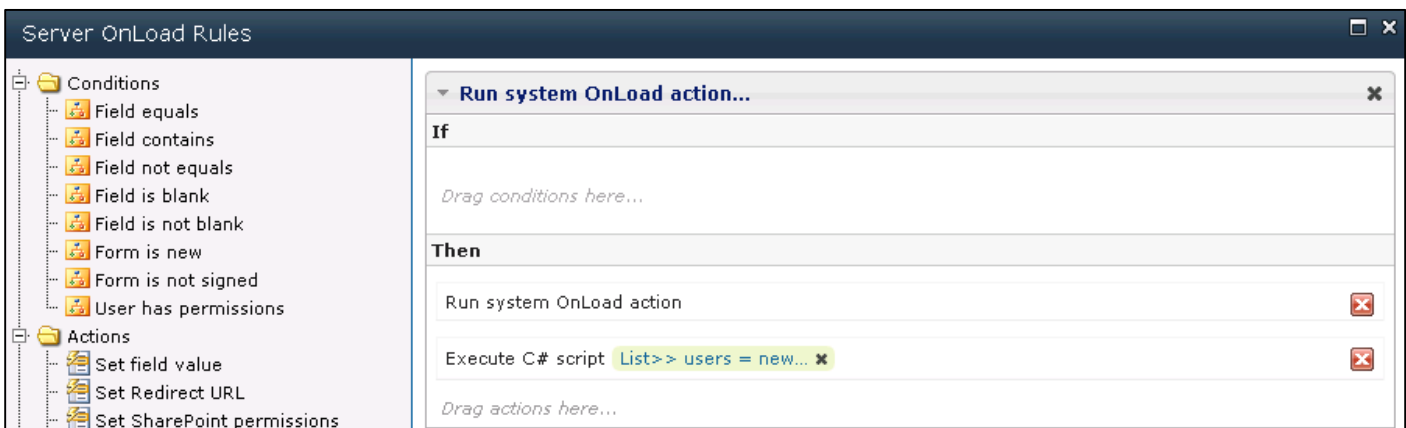
This template has multiline text field, a submit button and two people picker fields:

MultilineTextField1 – is used to store all users in a structure, that describes user names and the members of the group;

Group – people picker field that is populated with group names;

User – people picker field that is used to store users which belong to group chosen in “Group” field.

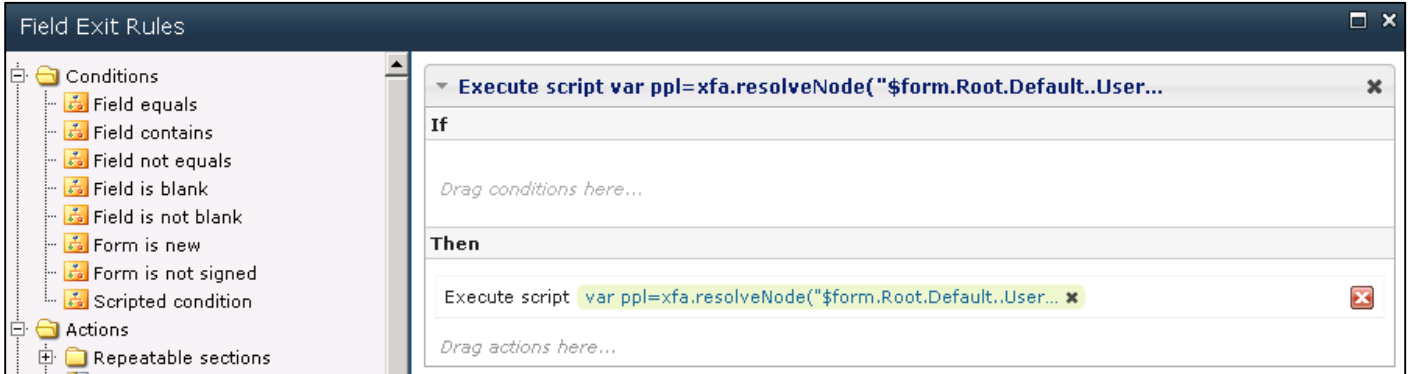
Step 2. Add Execute C# script to Form Load rules. Navigate to PDF Forms Tools → Developer → Form Load



Script

This script populates “Group” people picker and MultilineTextField1 with proper values.

Step 3. Add **Execute script** to Field Exit rules for “**Group**” people picker field.



Script

```
//clear User people picker
xfa.resolveNode("$form.Root.Default..User")._editor._dropdown._listbox.clearItems();
var peoplePickersNode = xfa.resolveNode('$data.__PdfFormsSystem.PeoplePickerData');
eval("PeoplePickerData=" + peoplePickersNode.value + ";");
eval("var yourdata=" + xfa.resolveNode('$data..MultilineTextField1').value + ";");
if(PeoplePickerData.uiControl5.length!=0){
    PeoplePickerData.uiControl5 = [];
}
//populate User people picker depending on Group people picker value
for(var i=0; i<yourdata.length;i++){
    for(var j=0; j<yourdata[i].Values.length;j++){
        var str = yourdata[i].Values[j].toString();
        var group = str.match("CN=(.*?),");
        if(group[1].toString()==xfa.resolveNode("$data..Group").value.toString())
        {
            PeoplePickerData.uiControl5[PeoplePickerData.uiControl5.length]=
{"Name":yourdata[i].Name[0], "Value":yourdata[i].Name[0]};
        }
    }
}
xfa.resolveNode('$data.__PdfFormsSystem.PeoplePickerData').value =
PeoplePickerData.toSource().match("([^(].*)\\\)")[1].toString();
var refreshPP = xfa.resolveNode("$form..User");
PdfForms.systemDataLoaded = false;
PdfForms.Controls.PeopleEditor.Initialize(refreshPP._editor);
```



NOTE: People Picker's id can be found by going to **PDF Form Tools** → **Page Design** → **XML** next to needed People Picker name

The screenshot shows the 'XML' view in the PDF Form Tools interface. The code includes JavaScript for field events and XML for the field definition. The XML snippet is as follows:

```

<subform xmlns="http://www.xfa.org/schema/xf-a-template/2.8/" name="Group" id="uiControl6" x="132pt" y="96pt">
  <bind match="none" />
</subform>

```

Step 4. Deploy template.

Runtime

Step 1. Open form and chose some value in “Group” field

The screenshot shows a PDF form with a 'Save' button and two dropdown menus. The first dropdown is labeled 'Group' and is open, showing a list of domain groups including Schema Admins, Domain Computers, Domain Controllers, Enterprise Admins, Cert Publishers, Domain Admins, Domain Users, Domain Guests, and Group Policy Creator O. The second dropdown is labeled 'User' and is currently empty. Below the dropdowns, a 'MultilineTextField1' contains the following JSON data:

```

[{"Name":["SPAdmin"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SPFarm"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SQLSvc"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SPSvc"],"Values":["CN=Domain

```



Step 2. Schema Admins group has only one user.

The screenshot shows a PDF form with a 'Save' button, a 'Group' dropdown menu set to 'Schema Admins', and a 'User' dropdown menu containing 'TanyaAdmin'. Below the form, a MultilineTextField1 contains a list of JSON objects representing users and their group memberships.

```
MultiLineTextField1 [{"Name":["SPAdmin"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SPFarm"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SQLSvc"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SPSync"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}]
```

Step 3. Chose other group

The screenshot shows the same PDF form, but the 'Group' dropdown menu is now set to 'Domain Admins'. The 'User' dropdown menu now lists several users: SPAdmin, SPFarm, SQLSvc, SPSync, TanyaAdmin, and ben. The MultilineTextField1 content remains the same as in the previous screenshot.

```
MultiLineTextField1 [{"Name":["SPAdmin"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SPFarm"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SQLSvc"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}, {"Name":["SPSync"],"Values":["CN=Domain Admins,CN=Users,DC=AD2008R2,DC=loc"]}]
```

NOTE: "User" field's refresh script is placed in "Group" field's **Field Exit** rule, so "User" field will be refreshed only when "Group" field value is chosen and user exits field.

TanyaAdmin user is member of both groups.

Form Load script

```

List<Dictionary<string,List<string>>> users = new List<Dictionary<string,List<string>>>();
List<Dictionary<string,string>> groups = new List<Dictionary<string,string>>();
List<Dictionary<string,string>> userList = new List<Dictionary<string,string>>();
DirectoryEntry entry = new DirectoryEntry("LDAP://CN=Users, DC=AD2008R2,DC=loc"); //this part
describes path to folder users are taken from. Should be specified
DirectorySearcher dSearch = new DirectorySearcher(entry);
dSearch.Filter = "(|(objectClass=user)(objectClass=group))";
SearchResultCollection allResults = dSearch.FindAll();
foreach (SearchResult sResultSet in allResults){
    if(sResultSet.Properties.Contains("cn") &&
sResultSet.Properties.Contains("userPrincipalName")){
        List <string> membof = new List <string>();
        foreach(String s in sResultSet.Properties["memberOf"]){
            membof.Add(s);
        }
        List <string> nname = new List <string>();
        nname.Add(sResultSet.Properties["cn"][0].ToString());
        Dictionary<string,List<string>> user = new Dictionary<string,List<string>>()
        {
            {"Name",nname},
            {"Values", membof}
        };
        users.Add(user);
    }
    if(sResultSet.Properties.Contains("cn") && sResultSet.Properties.Contains("groupType")){
        Dictionary<string,string> uGroup = new Dictionary<string,string>()
        {
            {"Name",sResultSet.Properties["cn"][0].ToString()},
            {"Value",sResultSet.Properties["cn"][0].ToString()}
        };
        groups.Add(uGroup);
    }
}
JavaScriptSerializer serializer = new JavaScriptSerializer();
var temp =
serializer.Deserialize<Dictionary<string,object>>(data.resolveNode("__PdfFormsSystem.PeoplePickerD
ata").value);
temp.Add("uiControl6",groups); //This is People Picker's id (marked with red) which must be
specified for different cases
temp.Add("uiControl5",userList); //This is People Picker's id (marked with red) which must be
specified for different cases
JavaScriptSerializer serializer2 = new JavaScriptSerializer();
data.resolveNode("MultilineTextField1").value = serializer2.Serialize(users);
data.resolveNode("__PdfFormsSystem.PeoplePickerData").value = serializer.Serialize(temp);

```

[Back to guide](#)