



# PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

## Custom Cascade People Picker

Product: PDF Share Forms Enterprise for SharePoint 2010

### Content

First approach .....	2
Second approach.....	8
Additional information.....	13

This guide describes approaches to create custom cascade people picker in static (acro) form, which allows getting required user information.



## First approach

**Step 1.** Add “Region” property in user profile properties. Navigate to **Central Administrator → Manage service applications → User Profile Service → Manage User Properties → New Property**

SharePoint 2010 Central Administration > Add User Profile Property

Use this page to add a property for user profiles.

\* Indicates a required field

**Property Settings**  
 Specify property settings for this property. The name will be used programmatically for the property by the user profile service, while the display name is the label used when the property is shown. After the property is created the only property setting you can change is the display name.

Name: \*  
 Region

Display Name: \*  
 Region

Type:  
 string (Single Value)

Length:  
 25

Configure a Term Set to be used for this property

Default User Profile Subtype

**Sub-type of Profile**  
 Please select the sub-type of user profiles with which you want to associate this profile property.

This will add “Region” parameter for all SharePoint users.

**NOTE:** Name that was given to property will be used in scripting.

**Step 1.1** Add “Region” property to all users:

My Site My Newsfeed My Content My Profile Find People

View My Profile as seen by: Everyone

What's happening?

**Hugo Smith**

hugo@email.com

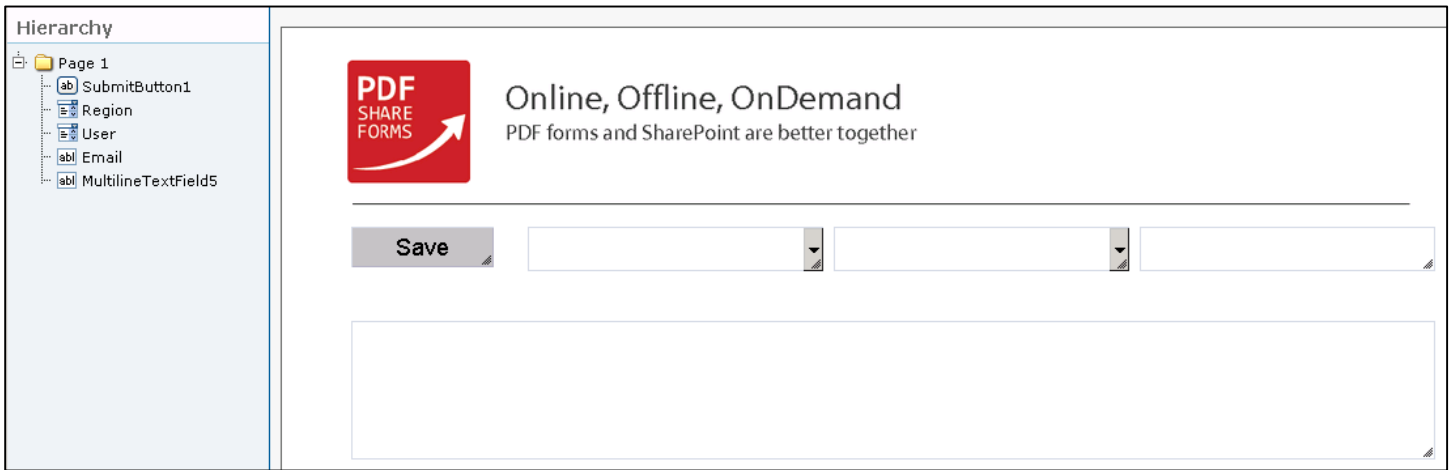
Region : Florida

Work e-mail : hugo@email.com

Edit My Profile

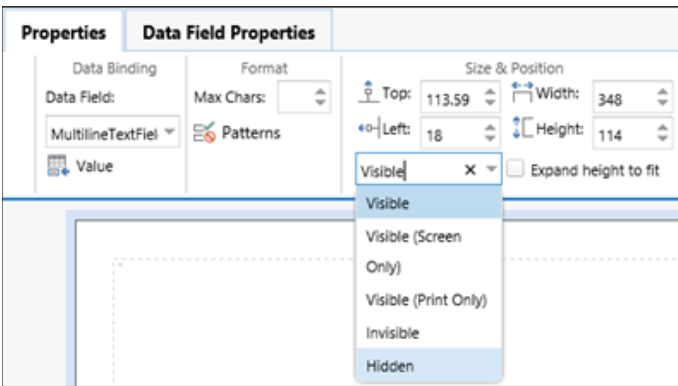
Hide information

## Step 2. Prepare template

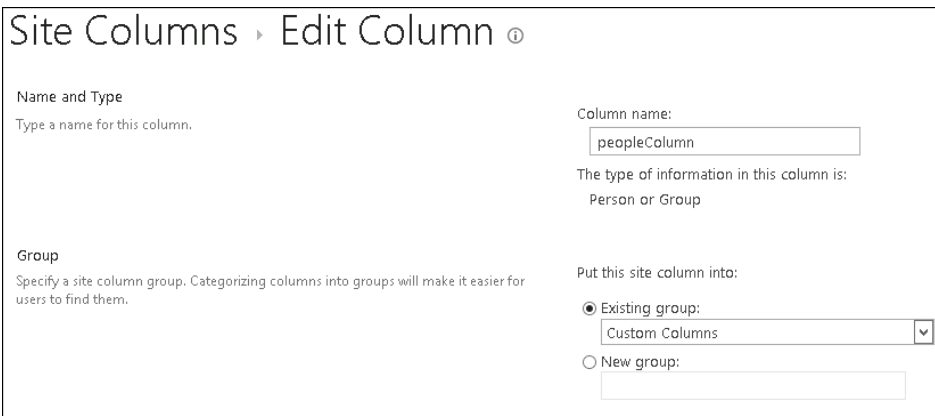


This template has “**SubmitButton1**”, “**Region**” dropdown field, which will be used to select region. “**User**” dropdown field will be populated with users from selected region; “**Email**” text field will store e-mail address for selected user. “**MultilineTextField5**” is used to store data from user profiles.

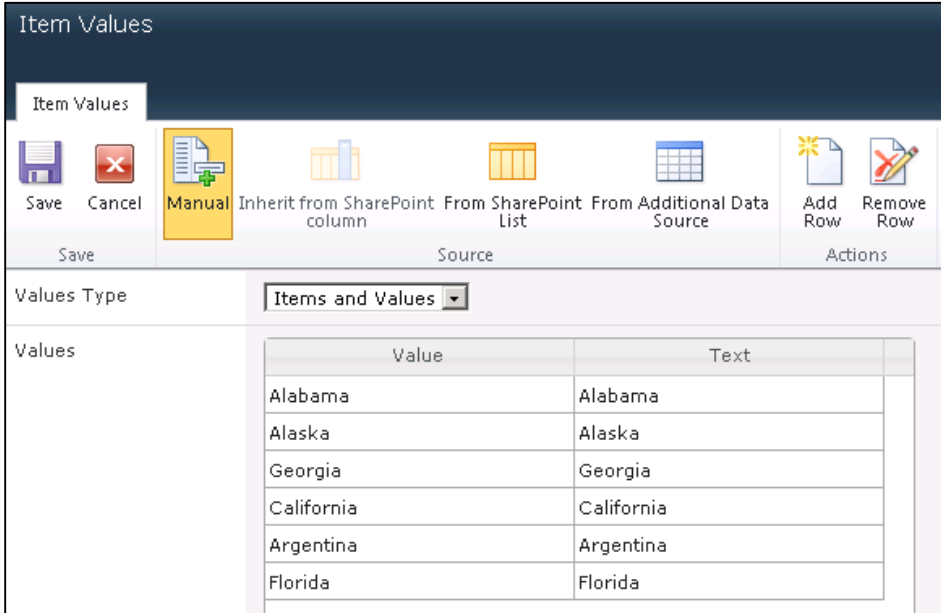
**NOTE:** you can make MultilineTextField5 hidden by selecting “**Hidden**” in “**Size & Position**” section of Properties tab



**NOTE 2:** “**User**” field must be mapped on “**Person or Group**” site column – in this case, it is “**peopleColumn**” column



**Step 3.** “Region” dropdown is populated manually. Choose field → **Properties** → **Item Values**



**Step 4.** Add JavaScript action to “Region” field’s blur action. Choose field and navigate to **PDF Form Tools** → **Properties** → **Actions** → **Blur**



**Script:**

```

this.getField("Email").value = "";
this.getField("User").clearItems();
eval("var yourdata=" +this.getField('MultilineTextField5').value+ ";"");
var dd = this.getField("User");
for(var i = 0;i< yourdata.length; i++){
    if(yourdata[i][1]==this.getField("Region").value){
        dd.insertItemAt(yourdata[i][0],yourdata[i][0],i);
    }
}
this.getField("User").value = "";

```

This script populates “User” dropdown field with data from “MultilineTextField5” field depending on “Result” dropdown field value.

**Step 5. Add Form Load rules.** Navigate to **PDF Form Tools → Developer → Form Load**



**Script:**

```

SPField peopleField = currentWeb.Fields.GetField("peopleColumn"); //column name must be specified
string fieldSettings = peopleField.SchemaXml;
//find which group must be extracted
var regex = System.Text.RegularExpressions.Regex.Match(fieldSettings,
"UserSelectionScope=\\\\"(.^\\\\"*)");
var result = regex.ToString();
regex = System.Text.RegularExpressions.Regex.Match(result, "[0-9]+");
result = regex.ToString();
List<List<string>> users = new List<List<string>>();
foreach (SPUser user in currentWeb.SiteUsers){
    using (SPSite site = new SPSite("http://sp2010app")) //sp ste url must be specified
    {
        SPServiceContext serviceContext = SPServiceContext.GetContext(site);
        // Create a user profile that uses the default user profile subtype.
        UserProfileManager userProfileMgr = new UserProfileManager(serviceContext);
        UserProfile userProfile = userProfileMgr.GetUserProfile(user.LoginName);
        List<string> userData = new List<string>();
        //Note - if user will not have region parametr filled then this uer will not be added
        to list
        userData.Add(user.Name); //get user name from column
        userData.Add(user.Email); //get user email from column
        userData.Add(userProfile["Region"].Value.ToString()); //get user region from userprofile
        users.Add(userData);
    }
}
//write list structure in field
JavaScriptSerializer serializer = new JavaScriptSerializer();
data.resolveNode("MultilineTextField5").value = serializer.Serialize(users);

```

This script reads region information from user profile. Name and e-mail are received from “peopleColumn” and all information is stored in MultipleTextField5 as list.

**Step 6.** Add JavaScript action to “User” field’s blur action. Choose field and navigate to **PDF Form Tools → Properties → Actions → Blur**

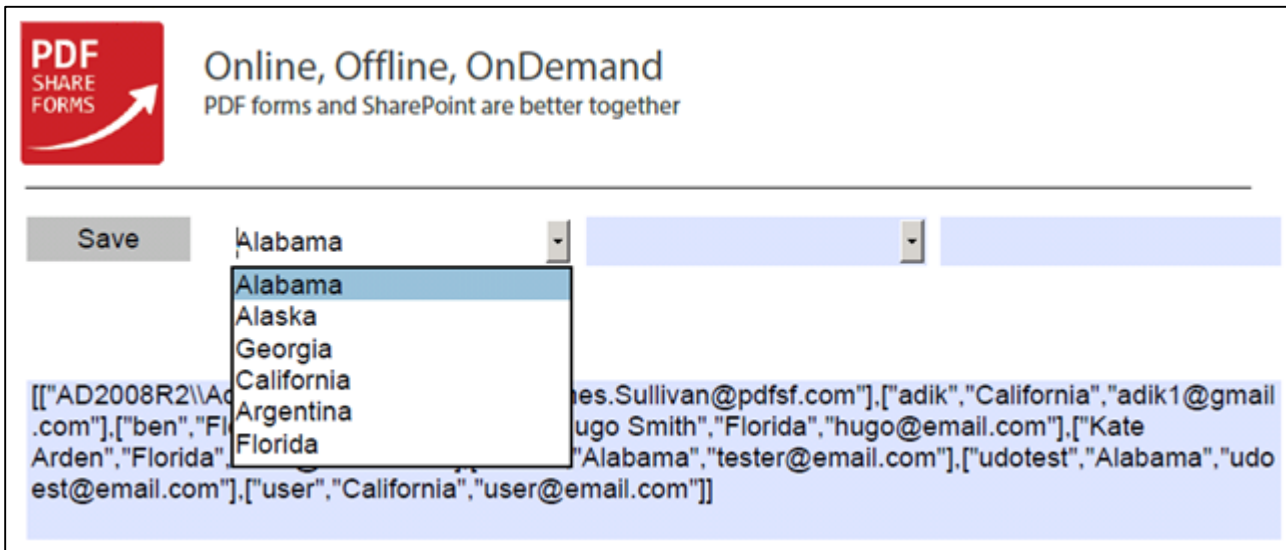


**Script:**

```
var temp = this.getField("User").value;
eval("var yourdata=" +this.getField('MultilineTextField5').value+ " ");
for(var i = 0;i< yourdata.length; i++ ){
if(yourdata[i][0] == this.getField("User").value){
if(yourdata[i][2] == ""){
this.getField("email").value = "This user does not have e-mail";} // if selected user does not
have e-mail - text field will store this message
else{
this.getField("Email").value = yourdata[i][2];}}}
```

This script will populate “Email” text field with user’s e-mail address from “MultilineTextField5”.

**Step 7.** Runtime. Choose region





### Step 7.1 "User" dropdown field is populated with values depending on region

Save Florida

ben  
Hugo Smith  
Kate Arden

```
[[["AD2008R2\Administrator","Alabama","James.Sullivan@pdfsf.com"],["adik","California","adik1@gmail.com"],["ben","Florida","ben@gmail.com"],["Hugo Smith","Florida","hugo@email.com"],["Kate Arden","Florida","kate@email.com"],["tester","Alabama","tester@email.com"],["udotest","Alabama","udotest@email.com"],["user","California","user@email.com"]]]
```

### Step 7.2 "Email" field is populated with selected user's e-mail address

Save Florida Hugo Smith hugo@email.com

```
[[["AD2008R2\Administrator","Alabama","James.Sullivan@pdfsf.com"],["adik","California","adik1@gmail.com"],["ben","Florida","ben@gmail.com"],["Hugo Smith","Florida","hugo@email.com"],["Kate Arden","Florida","kate@email.com"],["tester","Alabama","tester@email.com"],["udotest","Alabama","udotest@email.com"],["user","California","user@email.com"]]]
```



## Second approach

**Step 1.** Create list that will contain user information. In this example is used “**Region**” list which contains user name, region and e-mail address

	peopleColumn	Region	Email
	ben	Alabama	ben@gmail.com
	Hugo Smith	Florida	hugo@email.com
	Kate Arden	Florida	kate@email.com
	tester	California	tester@email.com
	user	Argentina	user@email.com
	adik	Florida	adik1@gmail.com

+ Add new item

**NOTE:** this template contains repeatable values in “Region” column.

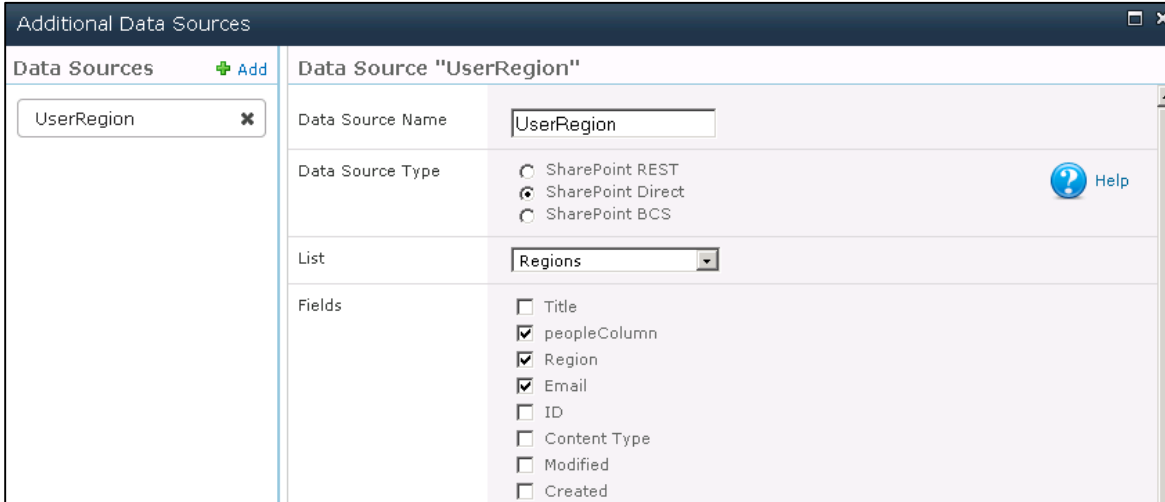
**Step 2.** Prepare template

This template has “**SubmitButton1**”, “**Region**” dropdown field, which will be used to select region. “**User**” dropdown field will be populated with users from selected region; “**Email**” text field will store e-mail address for selected user. “**MultilineTextField5**” is used to store data from list.

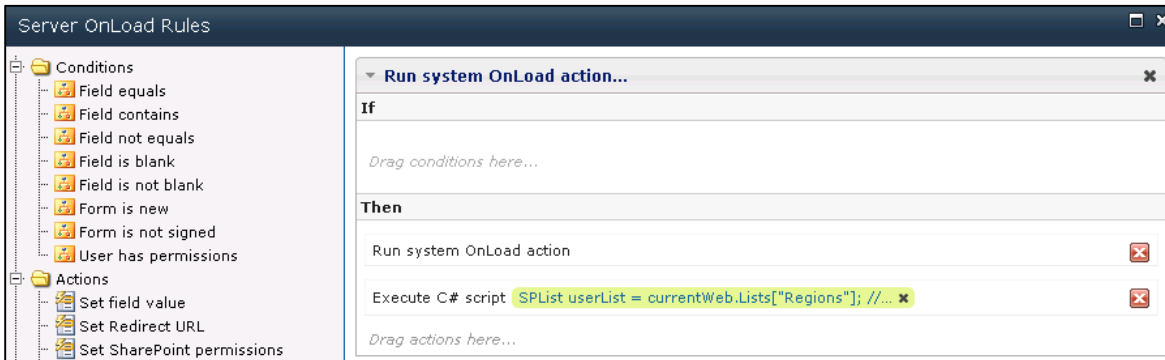
**NOTE:** you can make MultilineTextField5 hidden by selecting “**Hidden**” in “**Size & Position**” section of Properties tab



**Step 3.** Create new data source. Navigate to **PDF Form Tools → Developer → Data Sources**



**Step 4.** Add Form Load rules. Navigate to **PDF Form Tools → Developer → Form Load**



**Script:**

```

SPList userList = currentWeb.Lists["Regions"]; //list name must be specified for proper case
SPQuery query = new SPQuery();
query.Query = "<OrderBy><FieldRef Name='Region' /></OrderBy>"; //column name must be specified
//gets distinct values and writes in data table
DataTable dt = userList.GetItems(query).GetDataTable();
DataView dtview = new DataView(dt);
DataTable distinct = dtview.ToTable(true, "Region");
List<string> columns = new List<string>();
foreach(DataRow row in distinct.Rows){
foreach(DataColumn column in distinct.Columns){
columns.Add(row[column].ToString());}}
//serialize data
JavaScriptSerializer serializer = new JavaScriptSerializer();
data.resolveNode("MultilineTextField5").value = serializer.Serialize(columns);

```

This script takes distinct values from list and store in MultilineTextField5 as list.

**Step 5.** Add “Global JavaScript”. Navigate to **PDF Forms Tools → Developer → Global JavaScript**

```
var val = this.getField("Region").value;
eval("var yourdata=" + this.getField("MultilineTextField5").value + ";");
//Populate data from list to dropdown field
var dropdown = this.getField("Region");
var i = 0;
for(i = 0;i< yourdata.length; i++ ){
    dropdown.insertItemAt(yourdata[i],yourdata[i],i);
}
this.getField("Region").value = val;
```

This script populate “Region” dropdown field with data from “MultilineTextField5”.

**Step 6.** Add JavaScript action to “Region” field’s blur action. Choose field and navigate to **PDF Form Tools → Properties → Actions → Blur**



**Script:**

```
this.getField("Email").value = "";
this.getField("User").clearItems();
var datalist = DataSources['UserRegion']; // data sourcemust be specified
var region = this.getField("Region").value;
var dd = this.getField("User");
var i = 0;
for(i = 0;i< DataSources['UserRegion'].length; i++){
if(datalist[i]['Region'] == region){
dd.insertItemAt(datalist[i]['peopleColumn']);}} //datasource column must be specified
this.getField("User").value = "";
```

This script populates “User” dropdown field from datasource depending on value selected in “Region” dropdown field.

**Step 7.** Add JavaScript action to “User” field’s blur action. Choose field and navigate to **PDF Form Tools → Properties → Actions → Blur**

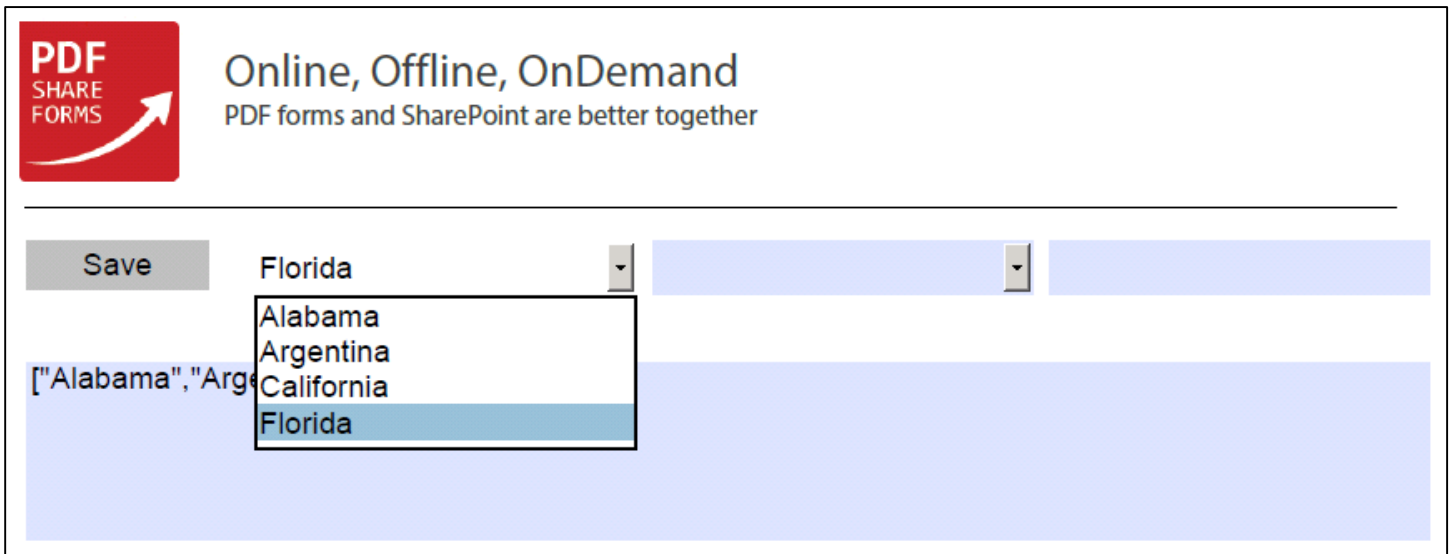


**Script:**

```
var datalist = DataSources['UserRegion']; // data source must be specified
var user = this.getField("User").value;
for(var i = 0; i < DataSources['UserRegion'].length; i++) {
  if(datalist[i]['peopleColumn'] == user) {
    this.getField("Email").value = datalist[i]['Email']; //datasource column must be specified
  }
}
```

This script will populate “Email” text field with user’s e-mail address from datasource.

**Step 8.** Result



“Region” dropdown is populated with distinct values from “Region” list.



Step 8.1 "User" dropdown field is populated with values depending on region

The screenshot shows a form header with the PDF SHARE FORMS logo and the text "Online, Offline, OnDemand" and "PDF forms and SharePoint are better together". Below the header is a horizontal bar containing a "Save" button, a dropdown menu set to "Florida", and another dropdown menu set to "Kate Arden". A third dropdown menu is partially visible to the right. Below this bar is a large light blue text area containing the text ["Alabama","Argentina","California","Florida"]. A dropdown menu is open below the "Kate Arden" dropdown, showing three options: "adik", "Kate Arden" (highlighted), and "Hugo Smith".

Step 8.2 "Email" field is populated with selected user's e-mail address

The screenshot shows the same form header as in Step 8.1. Below the header is a horizontal bar containing a "Save" button, a dropdown menu set to "Florida", a dropdown menu set to "Kate Arden", and a text field containing "kate@email.com". Below this bar is a large light blue text area containing the text ["Alabama","Argentina","California","Florida"].



## Additional information

Adobe Reader has functionality that provides auto suggestion in dropdown fields. Following steps must be done to achieve auto suggestions in forms.

**Step 1.** Check “Allow Custom Text Entry” for dropdown field. Choose field → **PDF Form Tools** → **Properties**

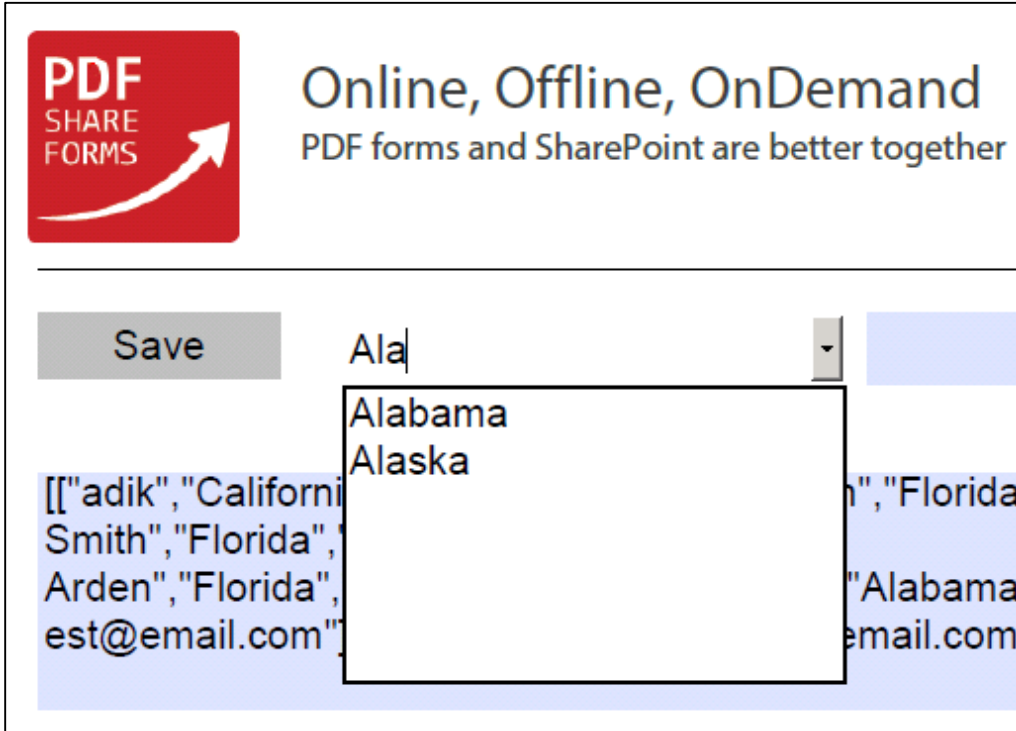
The screenshot shows the PDF Form Tools interface. The 'Properties' tab is active, displaying various settings for the selected field. The 'Allow Custom Text Entry' checkbox is checked. The 'Hierarchy' pane on the left shows the field structure, with 'Region' selected. The main preview area shows a form with a 'Save' button and a 'Region' dropdown menu.

**Step 2.** Runtime. Start typing text

The screenshot shows the form at runtime. The 'Region' dropdown menu is open, displaying a list of suggestions. The suggestions include 'Alab', 'California', 'Smith', 'Florida', 'Arden', 'Alabama', and 'est@email.com'. The 'Save' button is visible to the left of the dropdown.



**Step 3.** Remove character



All values which start from "Ala" are shown.