

Populating LookUp field from PDF form

This guide describes the steps to change LookUp field connected to current PDF form using drop-down field.

Step 1. Prepare a list and a library.

Step 1.1. List “Positions”:

Positions

[+ new item](#) or [edit this list](#)

All Items ...

✓	Title	emplPosition
1 ✖	...	Manager
2 ✖	...	Sales
3 ✖	...	IT Specialist

“Positions” list will serve as a base for LookUp column in the document library. It has three different values for LookUp column in “**emplPosition**” column.

Step 1.2. Document library “Employees”:

[+ new document](#) or drag files here

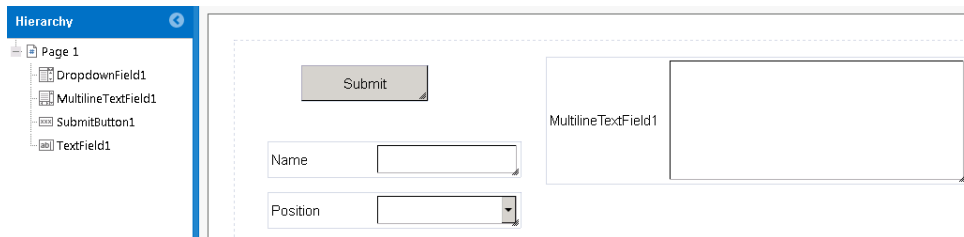
All Documents ...

✓	Name	Modified	Modified By	lookPosition
There are no files in the view "All Documents".				

“Employees” library will be used for template deployment. It already has LookUp column “**lookPosition**” assigned.

Step 2. Prepare a template.

Step 2.1. Template structure:



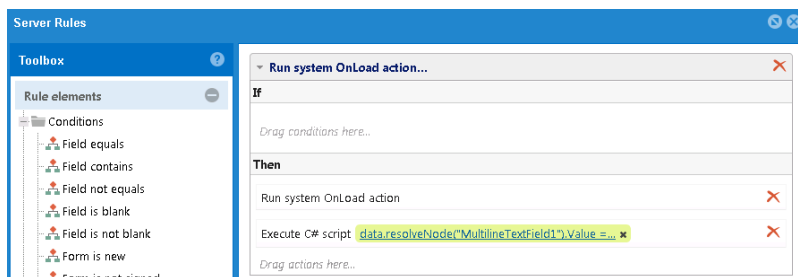
The template has a submit button, a text field, a drop-down field and multiline text field.

Text field “**TextField1**” is for demonstration purposes only;

Dropdown field “**DropdownField1**” will be used for selecting values from “**lookPosition**” column;

Multiline text field “**MultilineTextField1**” will be used for storing values loaded into form from “**lookPosition**” column. It must be hidden, so users can not change values inside it.

Step 3. Add “**Form Load**” script: *Developer* → *Form Load*.



Place “**Execute script**” action under “**Run system OnLoad action**”:

Code:

```
data.resolveNode("MultilineTextField1").Value = "";

SPList list = currentWeb.Lists["Positions"];
SPListItemCollection itemCollection;
itemCollection = list.Items;
foreach (SPListItem item in itemCollection)
{
    data.resolveNode("MultilineTextField1").Value =
data.resolveNode("MultilineTextField1").Value + ";" +
item["emplPosition"].ToString();
};
```

This script will load all values from “**lookPosition**” column in list “**Positions**” to “**MultilineTextField1**”.

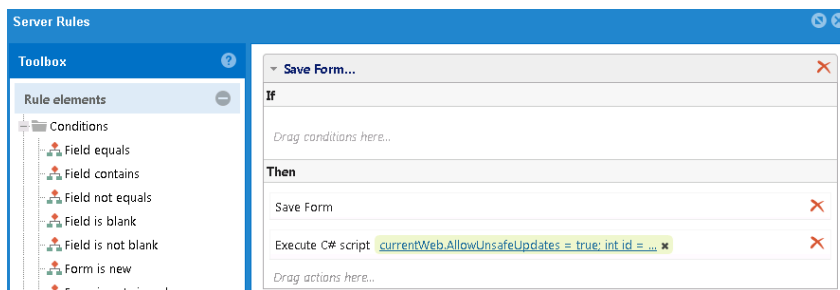
Step 4. Add “On Form Ready Script”: *Developer* → *On Form Ready Script*.

Code:

```
var dropdown = xfa.resolveNode('$form.Root.Default..DropdownField1');
dropdown.clearItems();
var val = xfa.resolveNode("$data..MultilineTextField1").value;
val = val.substring(1, val.length);
var len = val.split(";").length;
for(var i = 0;i< len; i++ ){
    dropdown.addItem(val.split(";")[i],val.split(";")[i]);
}
```

This script will populate “**DropdownField1**” with values from “**MultilineTextField1**”.

Step 5. Add “Form Submit” script: *Developer* → *Form Submit*.



Place “**Execute script**” action under “**Save Form**” action.

NOTE:

First you have to save the form, because it must be created as a library item before any actions with LookUp field. That’s why “Execute script” action must be placed under “Save Form” action.

Please see the next page for the code.

Code:

```
currentWeb.AllowUnsafeUpdates = true;
int id = 0;
SPList listPos = currentWeb.Lists["Positions"];
SPListItemCollection itemsPos = listPos.GetItems("emplPosInternalName");
foreach (SPListItem itemPos in itemsPos)
{
    if (itemPos["emplPosition"].ToString() ==
data.resolveNode("DropdownField1").Value)
    {
        id = itemPos.ID;
    }
}
SPList listEmpl = currentWeb.Lists["Employees"];
string fname = form.FileName;
SPListItem itemEmpl = listEmpl.GetItemByUniqueId(form.DocumentID);
itemEmpl["lookPosition"] = new SPFieldLookupValue(id,
data.resolveNode("DropdownField1").Value);
itemEmpl.Update();
currentWeb.AllowUnsafeUpdates = false;
```

This code will populate LookUp field assigned to a form with value from **"DropdownField1"**.

NOTE:

To select a value in lookup field we need two parameters:

ID and a name of value we want to select (in our example they are: "Manager", "Sales" and "IT Specialist").

Step 6. Obtain ID of selected value:

```
currentWeb.AllowUnsafeUpdates = true;
int id = 0;
SPList listPos = currentWeb.Lists["Positions"];
SPListItemCollection itemsPos = listPos.GetItems("emplPosInternalName");
foreach (SPListItem itemPos in itemsPos)
{
    if (itemPos["emplPosition"].ToString() ==
data.resolveNode("DropdownField1").Value)
    {
        id = itemPos.ID;
    }
}
```

This part of the script will get ID of value from **"emplPosition"** column of **"Positions"** list that will be selected by user in **"DropdownField1"**.

NOTE:

In order to get this value, we need to address to internal column name, but not the name we see in the list view.

```
SPListItemCollection itemsPos = listPos.GetItems("emplPosInternalName");
```

This is internal name of column **“emplPosition”**, we can access it by completing the following steps:

1) Go to the list settings: *List* → *List Settings*.

In our case it is “Positions” list:

Columns

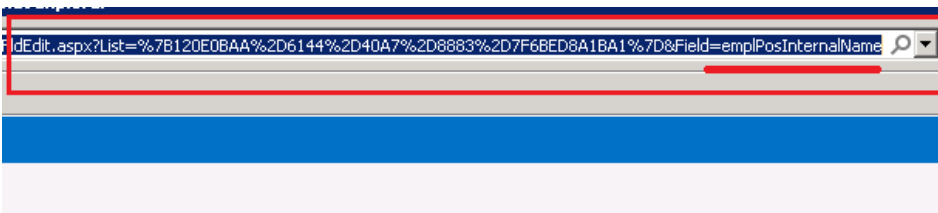
A column stores information about each item in the list. The following columns are currently available in this list:

Column (click to edit)	Type	Required
Title	Single line of text	✓
emplPosition	Single line of text	
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

▪ [Create column](#)

▪ [Add from existing site columns](#)

2) Select the needed column (“emplPosition”) in “Columns” section:



My Supernice SharePoint

 EDIT LINKS

Settings ▸ Edit Column

Name and Type

Type a name for this column.

Column name:

The type of information in this column is:

- Single line of text
- Multiple lines of text
- Choice (menu to choose from)
- Number (1, 1.0, 100)
- Currency (\$, ¥, €)
- Date and Time

3) Check the URL of page – its last part will be an internal name of the current column (“emplPosInternalName”):

http://prima/_layouts/15/start.aspx#/_layouts/15/FldEdit.aspx?List=%7B120E0BAA%2D6144%2D40A7%2D8883%2D7F6BED8A1BA1%7D&Field=emplPosInternalName

Step 7. Obtain the field name and populate data to LookUp field (lookPosition):

```
SPList listEmpl = currentWeb.Lists["Employees"];  
string fname = form.FileName;  
SPListItem itemEmpl = listEmpl.GetItemByUniqueId(form.DocumentID);  
itemEmpl["lookPosition"] = new SPFieldLookupValue(id,  
data.resolveNode("DropdownField1").Value);  
itemEmpl.Update();  
currentWeb.AllowUnsafeUpdates = false;
```

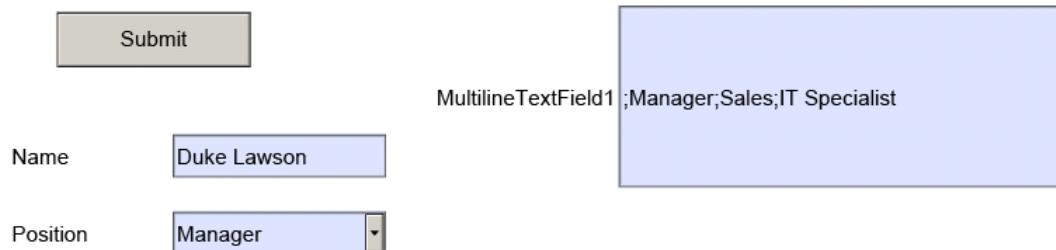
In this part of script we find current PDF file and add data from “**DropdownField1**” to LookUp field assigned to this PDF.

```
itemEmpl["lookPosition"] = new SPFieldLookupValue(id,  
data.resolveNode("DropdownField1").Value);  
itemEmpl.Update();
```

These two lines will add value to LookUp field and commit changes to SharePoint library.

Step 8. Deploy a form and run it.

Step 8.1. New form:



The screenshot shows a form with a 'Submit' button at the top left. Below it are three visible fields: 'Name' with the value 'Duke Lawson', 'Position' with a dropdown menu showing 'Manager', and a large, light blue rectangular area labeled 'MultilineTextfield1' containing the text 'Manager,Sales;IT Specialist'. The 'MultilineTextfield1' label is positioned to the left of the text area.

User has created new form from content type and filled it: wrote some name and last name inside “TextField1” and selected “Manager” in “DropdownField1”.

NOTE:

“**MultilineTextfield1**” must be hidden. It is visible only for demonstration purposes.




Step 8.2. Save the form and check the library:

Employees

 [new document](#) or drag files here

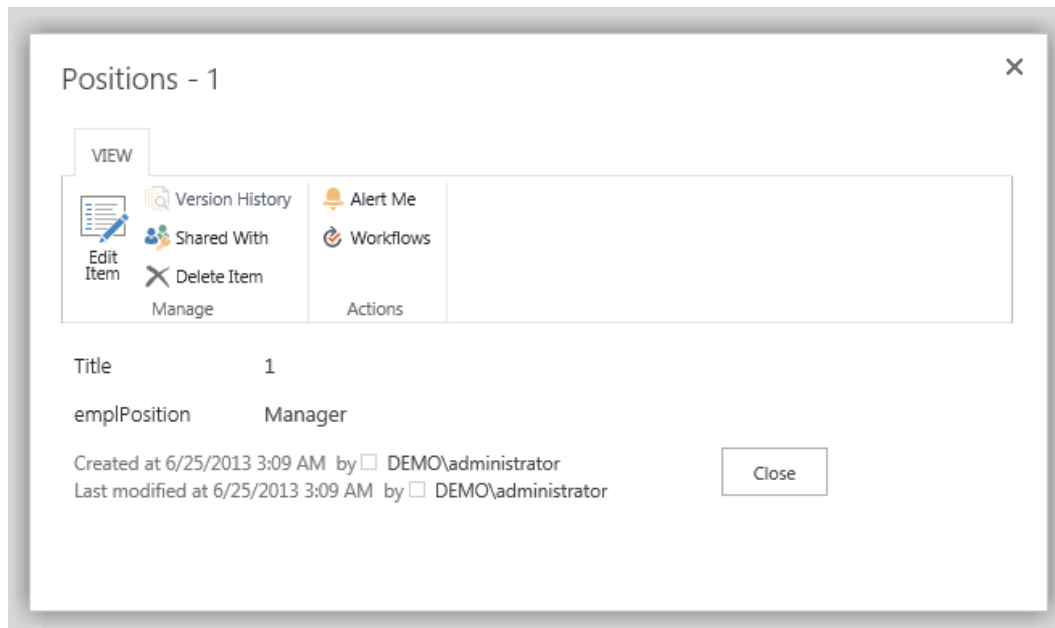
All Documents ...

Find a file 

<input checked="" type="checkbox"/>		Name	Modified	Modified By	lookPosition
<input type="checkbox"/>		EmployeeData_50 	48 minutes ago	<input type="checkbox"/> DEMO\administrator	Manager

User has saved the form and as can be seen, PDF has been created and “**lookPosition**” LookUp field has been populated with value from drop-down.

This entry acts as usual LookUp field value and can be previewed by clicking on it:



The screenshot shows a preview window titled "Positions - 1" with a close button (X) in the top right corner. Below the title is a "VIEW" tab and a menu of actions: "Edit Item", "Delete Item", "Version History", "Alert Me", "Shared With", and "Workflows". The main content area displays the following information:

Title	1
emplPosition	Manager

Created at 6/25/2013 3:09 AM by DEMO\administrator
Last modified at 6/25/2013 3:09 AM by DEMO\administrator

A "Close" button is located at the bottom right of the preview window.