PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

How to pick and execute workflow from PDF form

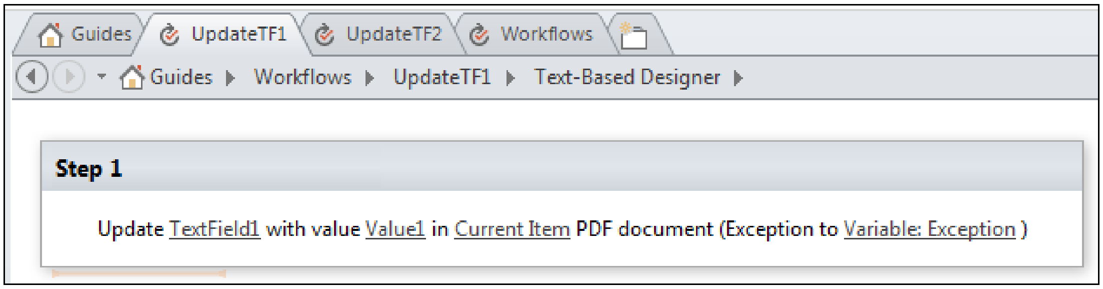
Contents

This guide will show how to pick and automatically execute SharePoint workflow from inside of PDF form right after PDF form was submitted.

Used workflows

For this guide in SharePoint Designer were made two simple workflows – each will populate field inside PDF form:

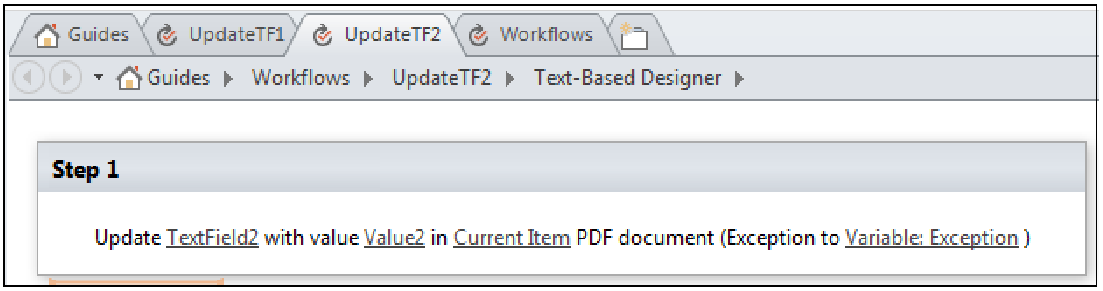
Workflow “UpdateTF1” – it will update TextField1 inside PDF form with text “Value1”:



Step 1

Update TextField1 with value Value1 in Current Item PDF document (Exception to Variable: Exception)

Workflow “UpdateTF2” – it will update TextField2 inside PDF form with text “Value2”:

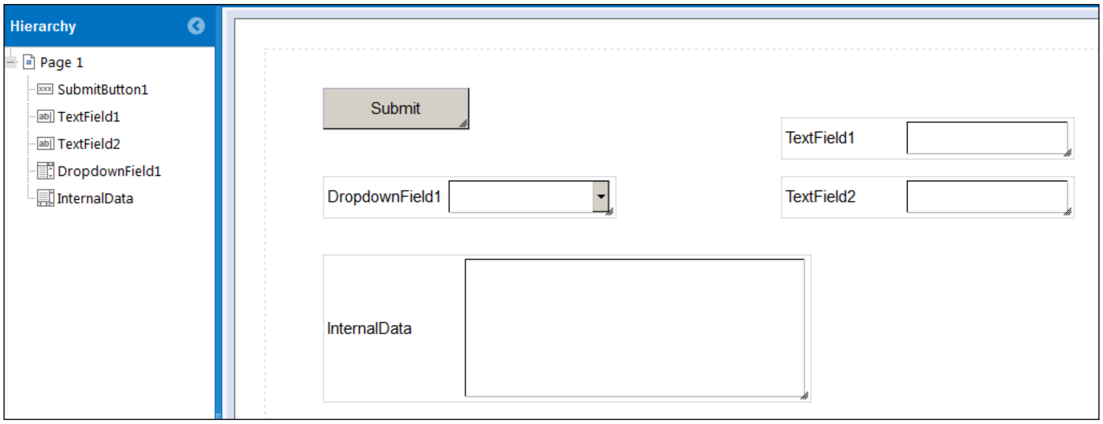


Step 1

Update TextField2 with value Value2 in Current Item PDF document (Exception to Variable: Exception)

PDF template

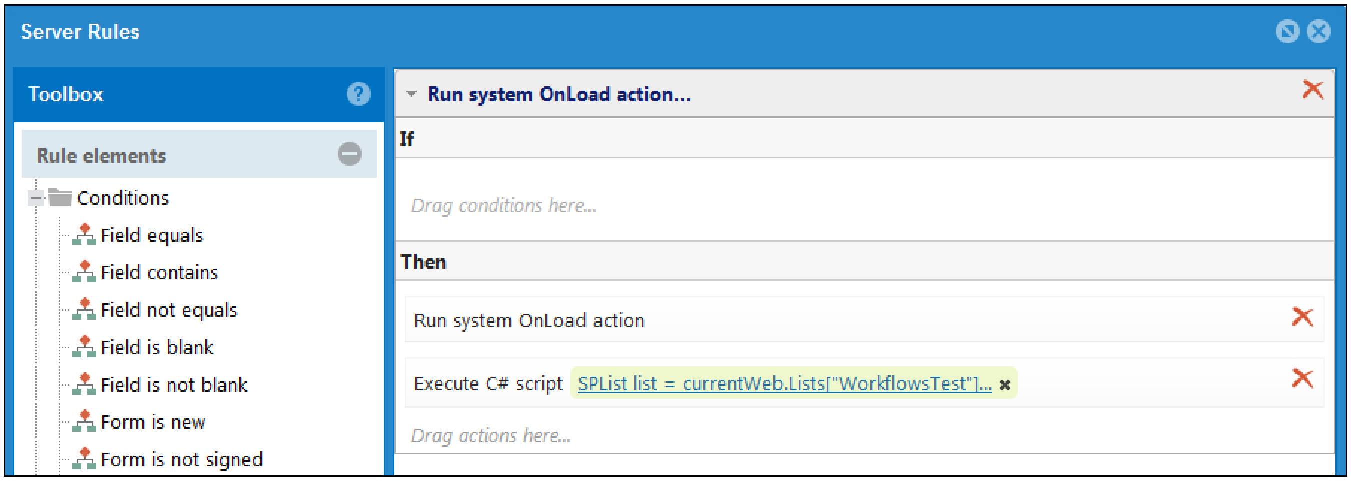
Step 1. Prepare template



This template contains submit button, two text fields wich will be populated through workflows (TextField1 and TextField2), dropdown field which will serve as required workflow selector (DropdownField1) and multiline text field – will be used to store internal form data (InternalData).

Note: multiline text field “InternalData” should be set to invisible so users could not change information stored inside it. Now it is left visible for demonstration purposes.

Step 2. Add Server-side script to “Form Load” event (Developer → Rules → Form Load):



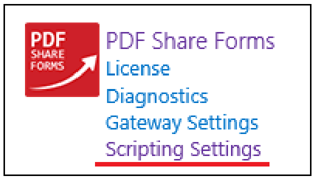
Place “Execute script” action below “Run system OnLoad action”.

Code:

```
SPList list = currentWeb.Lists["WorkflowsLib"];
List <string> WorkflowNames = new List <string>();
SPWorkflowAssociationCollection associationColl = list.WorkflowAssociations;
foreach (SPWorkflowAssociation association in associationColl){
	if (!association.Name.Contains("Previous Version:")){
		WorkflowNames.Add(association.Name);
	}
}
JavaScriptSerializer serializer = new JavaScriptSerializer();
data.resolveNode("InternalData").Value = serializer.Serialize(WorkflowNames);
```

Following code will extract names of all workflows assigned to SharePoint library (in this example called “WorkflowLib”). While extracting it will ingone older version of the same workflows if they were published multiple times. It the end all extracted information will be saved into “InternalData” multiline text field.

Please note that following assemblies and namespaces should be added in Central Administration → PDF Share Forms → Scripting Settings:



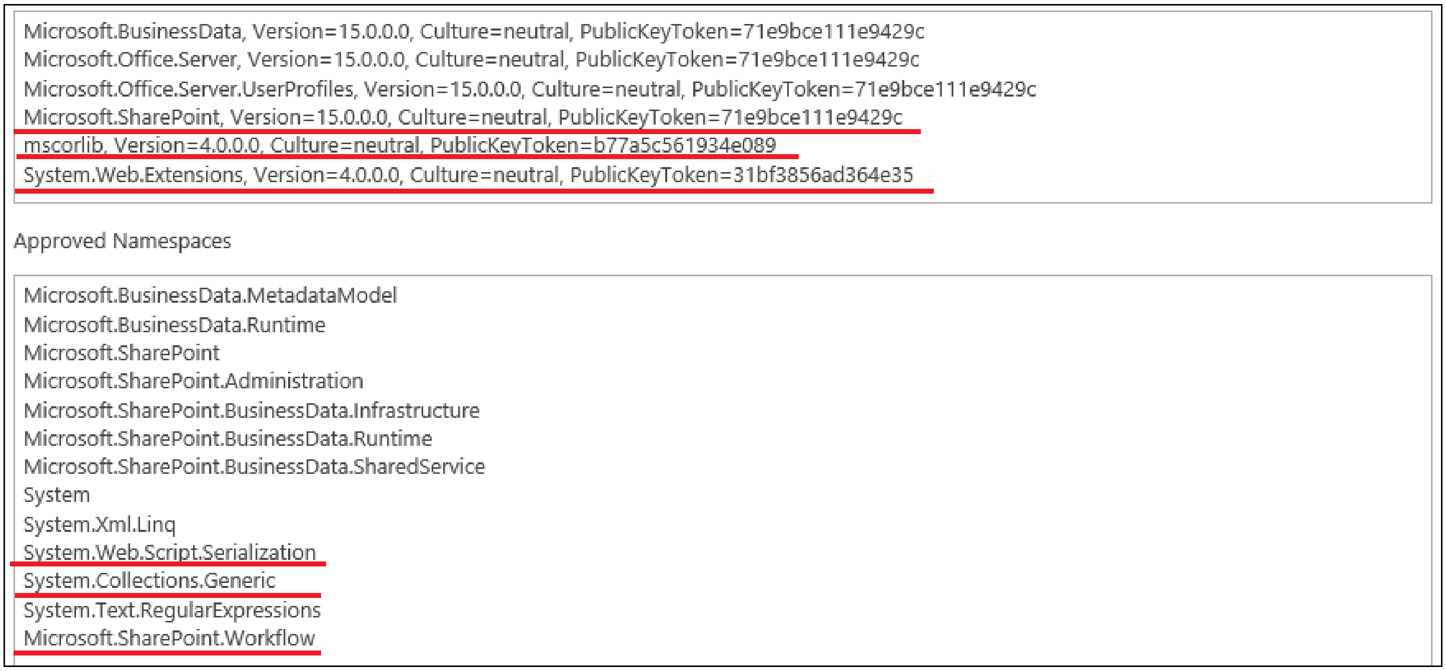
In order to add required asseblies or namespaces you can pick required from dropdown selectors or just write them into Approved Namesapces and approved Asseblies manually.

Assemblies to add:

- Microsoft.SherePoint
- System.Web.Extensios
- mscorlib

Namespaces to add:

- Microsoft.SherePoint.Workflow
- System.Web.Script.Serialization
- System.Collection.Generic



Microsoft.BusinessData, Version=15.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
Microsoft.Office.Server, Version=15.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
Microsoft.Office.Server.UserProfiles, Version=15.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
Microsoft.SharePoint, Version=15.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c
mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089
System.Web.Extensions, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35

Approved Namespaces

Microsoft.BusinessData.MetadataModel
Microsoft.BusinessData.Runtime
Microsoft.SharePoint
Microsoft.SharePoint.Administration
Microsoft.SharePoint.BusinessData.Infrastructure
Microsoft.SharePoint.BusinessData.Runtime
Microsoft.SharePoint.BusinessData.SharedService
System
System.Xml.Linq
System.Web.Script.Serialization
System.Collections.Generic
System.Text.RegularExpressions
Microsoft.SharePoint.Workflow

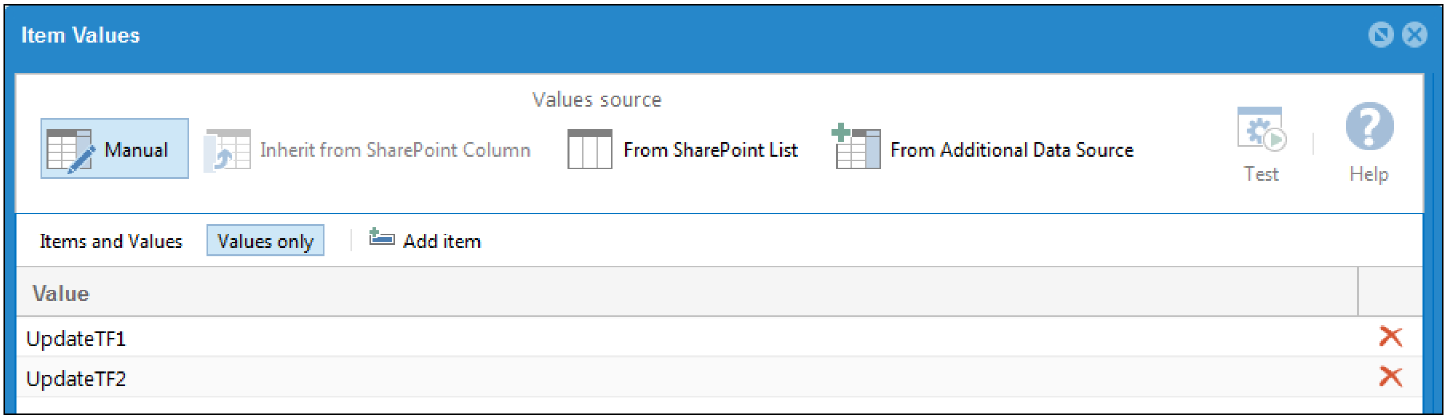
Step 3. Add Client-side script to “Form Ready” event (Developer > Client Scripts > Global Events > Form Ready):

```
var val = xfa.resolveNode("$data..DropdownField1").value;
eval("var yourdata=" + xfa.resolveNode("$data..InternalData").value + ";");
var dropdown = xfa.resolveNode("$form.Root.Default..DropdownField1")
for(var i=0; i < yourdata.length; i++) {
 dropdown.addItem(yourdata[i], yourdata[i]);
}
xfa.resolveNode("$data..DropdownField1").value = val;
```

This code will populate “DropdownField1” dropdown with values saved inside “InternalData” multiline text field.

Step 2-3. (alternative) Adding workflows' names manually

In case if it is not necessary to update "DropdownField1" with available workflows dynamically each time when form is opened, it is also possible to set predefined values for dropdown field in PDF Designer. Select dropdown field and go to Properties → Item Values:

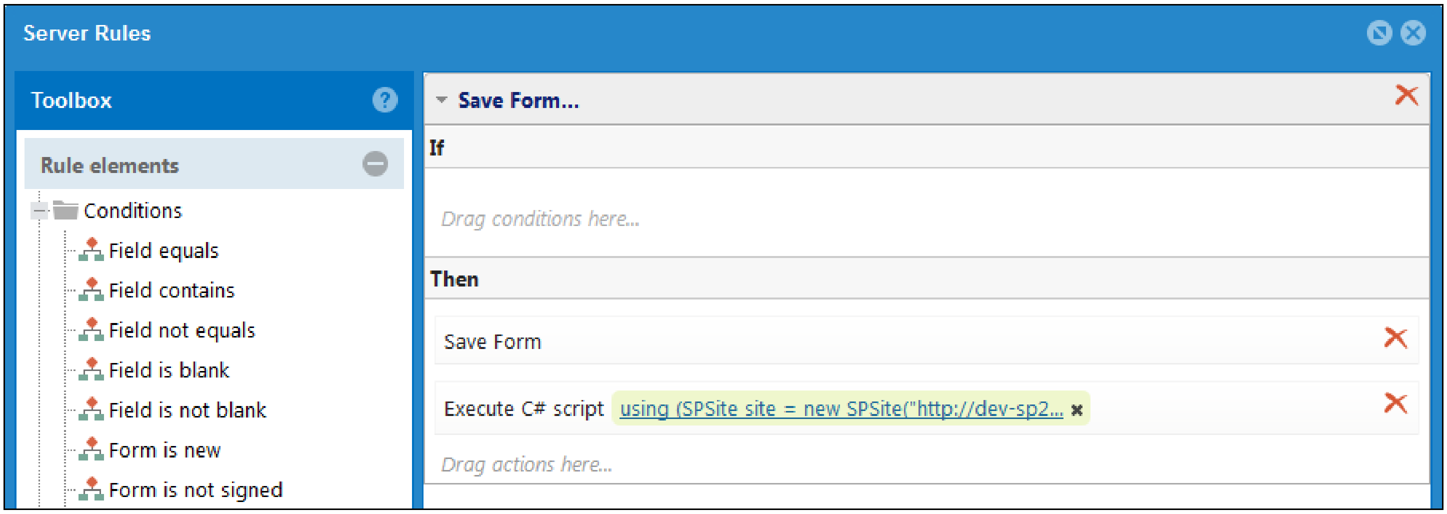


UpdateTF1

UpdateTF2

Here you can manually populate dropdown field with necessary values. In our case it will be "UpdateTF1" and "UpdateTF2".

Step 4. Add Server-side script to "Form Submit" event (Developer → Rules → Form Submit):



Save Form...

If

Then

Save Form

Execute C# script using (SPSite site = new SPSite("http://dev-sp2...

Place "Execute script" action below "Save Form".

Code:

```
using (SPSite site = new SPSite("http://yoursite.com/sitecollection/"))
{
    SPFile file = currentWeb.GetFile(form.DocumentID);
    SPListItem listItem = file.Item;
    SPList list = currentWeb.Lists["WorkflowsLib"];
    List <string> WorkflowNames = new List <string>();
      SPWorkflowAssociationCollection associationColl = list.WorkflowAssociations;
      foreach (SPWorkflowAssociation association in associationColl){
             if (association.Name == data.resolveNode("DropdownField1").Value){
                   association.AssociationData = string.Empty;
                   site.WorkflowManager.StartWorkflow(listItem, association,
association.AssociationData);
             }
    }
}
```

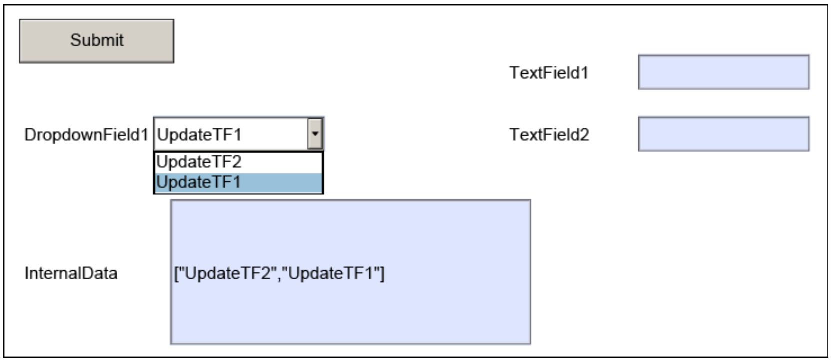
This code will take value selected in “DropdownField1”, then will use it to find right SharePoint workflow to start in library “WorkflowsLib”. In this example will be used current PDF form as target item for workflow execution.

Step 5. Save and deploy PDF template

In this example template will be deployed to SharePoint library “WorkflowsLib”.

Runtime

Step 1. Create new PDF form from deployed template



TextField1

DropdownField1 UpdateTF1
UpdateTF2
UpdateTF1

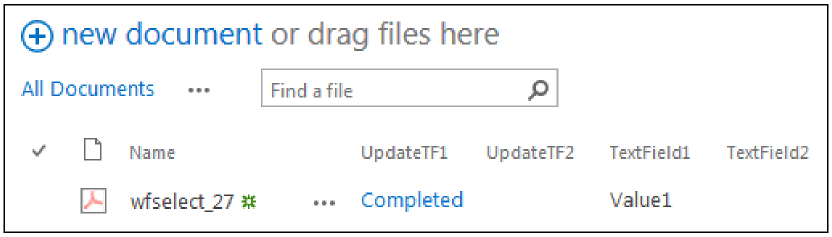
TextField2

InternalData ["UpdateTF2","UpdateTF1"]

As it can be seen, “DropdownField1” was populated with workflow names and “InternalData” has values extracted during "Form Load" event. For this time will be selected “UpdateTF1” inside “DropdownField1”.

Note: multiline text field “InternalData” should be set to invisible so users could not change information stored inside it. Now it is left visible for demonstration purposes.

Step 2. Save form and check the library

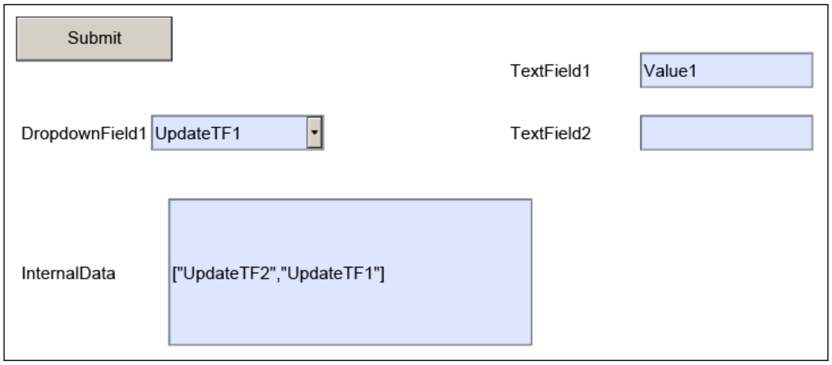


✓	Name		UpdateTF1	UpdateTF2	TextField1	TextField2
	wfselect_27 ✱	•••	Completed		Value1	

Selected workflow was executed and populated “TexField1” with text “Value1”

Step 3. Selecting different workflow

Open already existing PDF form which was used in steps above:



TextField1: Value1

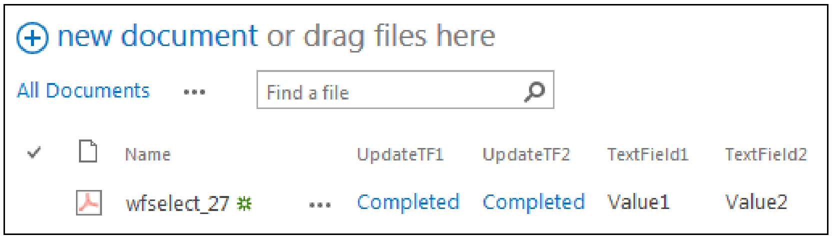
DropdownField1: UpdateTF1

TextField2:

InternalData: ["UpdateTF2","UpdateTF1"]

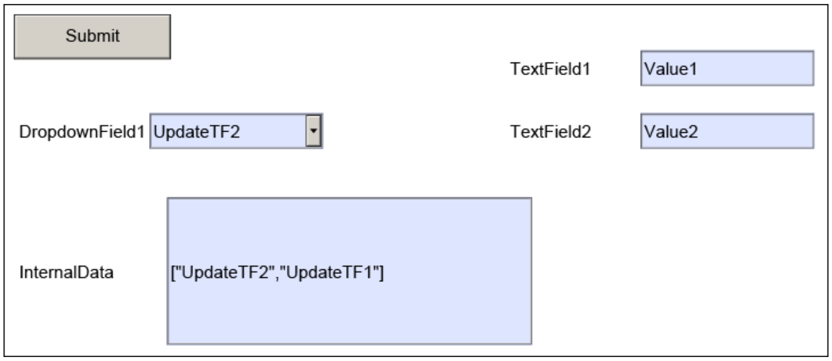
Select “UpdateTF2” in “DropdownField1” ans submit for again.

Step 4. Check the library



✓	Name		UpdateTF1	UpdateTF2	TextField1	TextField2
	wfselect_27 ✱	•••	Completed	Completed	Value1	Value2

Now “TextField2” is also populated with “Value2” and workflow “UpdateTF2” set to “Completed”.



TextField1: Value1

DropdownField1: UpdateTF2

TextField2: Value2

InternalData: ["UpdateTF2","UpdateTF1"]