Developer Guide Process Platform - Dynamic Drop Down Form Element

Document version: 7.0

Revision: 23-09-2019

RESULTMAKER

Contents

1 Introduction

The Dynamic Drop Down Form Element (DDFE) is an advanced web form drop down that dynamically loads items from a Data Source assembly installed in the Process Engine.

When a Data Source assembly is installed in the Process Engine, the data source is exposed to and available in the Process Designer when adding a Dynamic Drop Down Form Element to a form page.

At runtime when the end user uses the Dynamic Drop Down Form Element, the data items are retrieved from them installed assembly and presented to the user. It is possible to filter the items based on values in other form fields on the same page and it is possible for the end user to filter the returned items in a filter-as-you-type maner.

1.1 Purpose

The purpose of this document is to provide a rough guideline to the developer on how to develop a data source as a C# assembly and how to consume that data source in the Process Designer at form design time adding a Dynamic Drop Down Form Element to the form page.

1.2 Definitions, Acronyms and Abbreviations

- DDFE – Dynamic Drop Down Form Element – a form element that is added in the Process Designer to a form page at design time by a process consultant.
- Data Source – A C# assembly that implements the IDynamicDropDownDataSource interface and delivers a number of data items each of which contains a number of attributes consisting of a name and value.
- Item – the basic unit of data coming from a data source. Each item contains the attribute names but may differ in attribute values.
- Attribute – A name-value pair describing some property of interest of a data item. All items in a datasource contain the same attribute names, but may differ in attribute values.
- Key - The Id of the dataitem in the datasource. Each key is unique for an item.
- DisplayItem - The text that will be displayed in the dynamic drop down at runtime, where each entry represents a data item.

2 Architecture of the Dynamic Dropdown

This section describes the architecture of the Dynamic Drop Down.

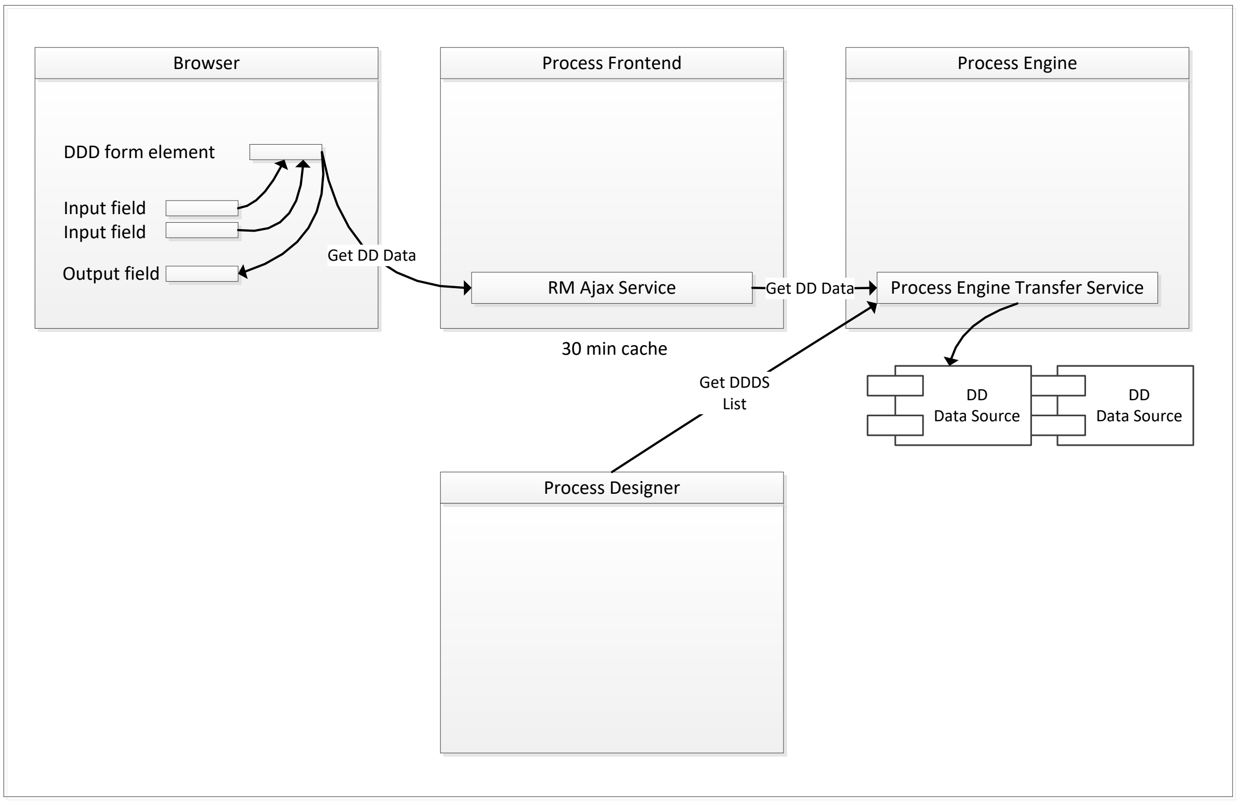


Figure 1 Dynamic Drop Down Architecure

The Dynamic Data Source assembly resides in the bin or Ext folder of the Process Engine. From there it is accessible to the Process Designer, so that the installed Dynamic Data Sources can be listed and selected and mapped to other form elements on the page.

In the browser the Dynamic Dropdown Form element is presented and the data is fetched via JQuery from the Process Frontends RM Ajax Service. The RM Ajax Service again fetches the data from the Process Engines Transfer Service.

The Get DD Data operations to the RM Ajax Service can be filtered be values from other fields on the page. In addition, the attributes on the end-user selected item in the Dynamic Drop Down Element can be mapped to fields on the page.

Beware that the RM Ajax Service at the Process Frontend level has a caching mechanism. The length of the cache is controlled in the Frontends `CacheMinutesForDropDownWebService` configuration key in the Resultmaker.OC.Frontend section of the Resulmaker.OC.FrontEnd.config.

3 Creating a Dynamic Drop Down Data Source

This section describes how to create a Dynamic Drop Down Data Source. This example is based on Visual Studio 2010, but it should work fine in Visual Studio 2008.

Start by creating a new class library.

3.1 Data Source References

Make sure that you reference the proper dlls in your class library project. You should take the below mentioned dll from the Process Platform server in the folder c:\inetpub\wwwroot\OC\bin\ or similar, depending on the installation of the Process Platform.

3.1.1 Required Reference Dlls

Resultmaker.OC.Shared

This must be the version that matches the Process Platform that is installed on the server, where the data source is to be installed on.
The assembly can be copied from C:\inetpub\wwwroot\OC4\bin.

3.1.2 Interface Class

IDynamicDropDownDataSource

Make sure that the class inherits from
Resultmaker.OC.Shared.Datasource.IDynamicDropDownDataSource.

```
using Resultmaker.OC.Shared;
using Resultmaker.OC.Shared.Datasource;

namespace DynamicDropdownDataSourceExampleTest
{
    /// <summary>
    /// The datasource must always derive from IDynamicDropDownDataSource
    /// </summary>
    public class TestFoodDataSource : Resultmaker.OC.Shared.Datasource.IDynamicDropDownDataSource
    {
     ......
    }
```

Figure 2.1.1 Initializing the data source.

This requires you to implement the following required members and methods.

- Properties
  - DataSourceName
- Methods
  - DataSourceDisplayItem[] GetDisplayItems()
  - DataSourceDisplayItem[] GetDisplayItems(IDictionary<string, string> parameter)
  - IEnumerable<string> GetAttributeNames()
  - IDictionary<string, string> GetItemAttributes(string key)
  - IEnumerable<string> GetInputParameterNames()

3.1.3 Obsolete attribute class reference

[ProcessPlatformPlugin]

This attribute was once used to mark a class as a plugin consumable by the engine. These days it is obsolete and should be removed from classes still implementing it.

3.2 IDynamicDropDownDataSource

This section describes the mandatory members and methods that the IDynamicDropDownDataSource requires to be implemented.

3.2.1 Properties

String DataSourceName

This property is used to represent the name of the data source, and is used for presentation when the data source is chosen on the Dynamic Drop Down Form Element in the Process Designer at design time.

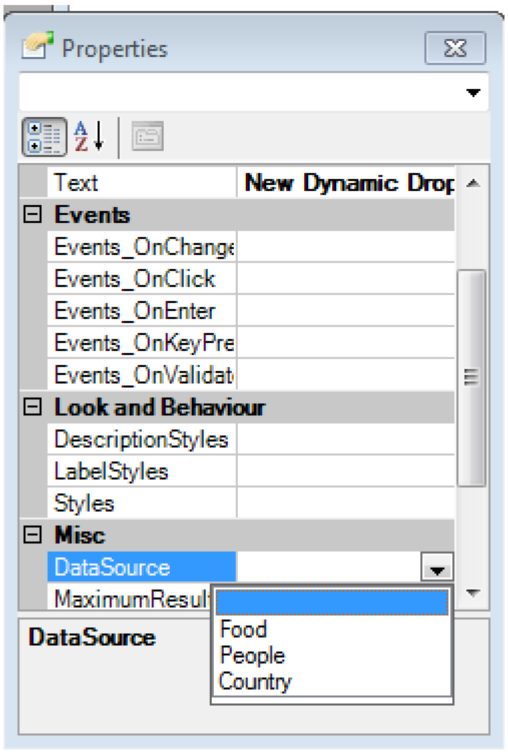


Figure 1 Selecting on of the installed data sources

3.2.2 Methods

3.2.2.1 GetDisplayItems()

This method is used to show a list of items in the DDFE for the user to choose from at runtime. The method returns an array of DatasourceDisplayItem (details in section 2.3) each of which represents an item in the data source. A DataSourceDisplayItem contains a data item key and display text. The text value is what is presented to the user see, see Figure 2.

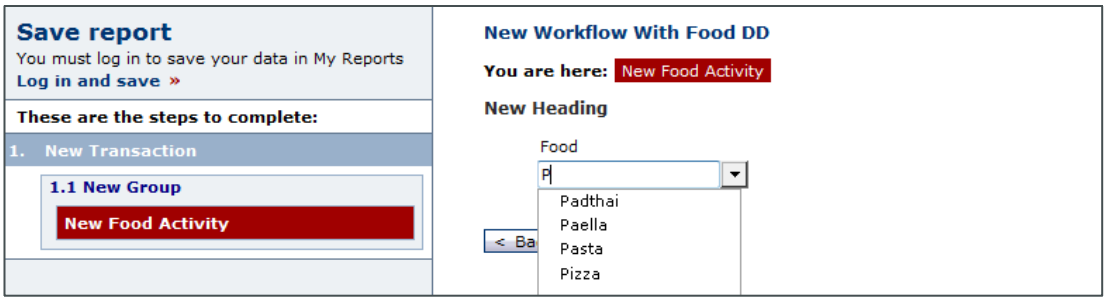

Save report
You must log in to save your data in My Reports
Log in and save »
These are the steps to complete:
1. New Transaction
1.1 New Group
New Food Activity
New Workflow With Food DD
You are here: New Food Activity
New Heading
Food
P
Padthai
Paella
Pasta
Pizza
< Ba

Figure 2. A Dynamic Drop Down From Element in action showing the display texts

3.2.2.2 GetDisplayItems(IDictionary<string, string> parameter)

This method is used to return data elements that are shown as a list of display items in the DDFE for the user to choose from at run-time. The method returns an array of DatasourceDisplayItem (details in section 2.3) each of which represents an item in the data source. This method allows for input parameters, so that information on the form page can be sent to the data source and used as a filtering value.

3.2.2.3 GetAttributeNames()

This method is used to get all the attribute names, which the data source contains and is able to return. (For example in the food data source, the attribute names can be name, price, and continent). The developer should changes attribute names according to the data source data.

3.2.2.4 GetItemAttributes (string key)

This method is used to return all attributes for a specific data item (including its name and value) filtered by key. **Note that the data type has been changed from an int to a string in version 6.0.5300 in order to be able to accommodate more advanced data sources**.

For the Food data source, the item attributes are Name, Price and Region.

3.2.2.5 GetInputParameterNames()

This method is used to return all the defined input parameter names, so that these parameters can be used in the GetDisplayItems(IDictionary<string,string> param). These input parameter name lists are part of the binding properties in the Process Designer, where the parameter values can be mapped to form elements. In this way form element values can work as filtering values for the data source return values.

3.2.3 Interface change

As of process platform 6.2.3000, the DataSourceName property no longer has a setter in the interface. Any DataSources implemented prior to this change should have the setter removed. Note that if an implementation has a setter that throws an exception when used (as is quite common as the setter was never used) it must be deployed using automatic registration in the Ext folder as detailed later.

3.3 DataSourceDisplayItem

The DataSourceDisplayItem represents an item in the data source, a row so to speak.

Properties

String Key

This is the Id of the data item which will be used to identify which data item is selected. **Note that the data type has been changed from an int to a string in version 6.0.5300 in order to be able to accommodate more advanced data sources**.

String DisplayItem

This property contains the text to display to the user when representing the item in the dynamic drop down in the front end.

4 Creating a Dynamic Drop Down Form Element

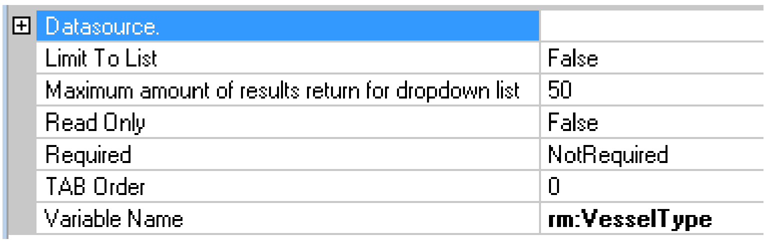
This section describes how to set up and add a Dynamic Drop Down Form Element to the form in the Process Designer.

4.1 Dynamic Drop Down Properties

4.1.1 DataSource

This is the name of the data source which will be used for the Dynamic Drop Down Form Element.

Note: If an existing datasource name is selected, a list of attribute name and input parameter name for that datasource will appear automatically. These attribute names and input parameter names can be mapped to element name if so desired. When the user selects an item, its attribute/parameter values will be transferred to the mapped element name on form submission. (see example 3.3).



⊞ Datasource.	
Limit To List	False
Maximum amount of results return for dropdown list	50
Read Only	False
Required	NotRequired
TAB Order	0
Variable Name	**rm:VesselType**

Figure 3 The Dynamic Drop Down Data Source

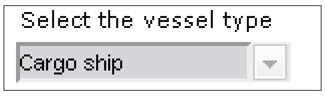
4.1.2 Maximum amount of results returned for the drop down

This is the maximum of display items to display at a time on the frontend. The default value is 50. If you have a list of data items that exceed the specified value, the Dynamic Drop Down Form Element will display the first 50 data items. When the user types in a search text the drop down will change the list to match the search filter with the next 50 items.

When the specified value is increased, this will increase the time taken to display the search so balancing the maximum to a good amount is advised.

4.1.3 Read Only

This will render the Dynamic Drop Down Form Element read only and it will not be possible to select a value with the arrow or type in a value. The field will however present the value from the corresponding Workflow Variable specified by name in the Variable Name property.


Select the vessel type

Cargo ship

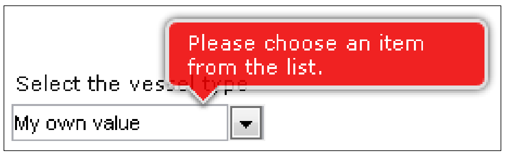
Figure 2 Read Only rendering of a prefilled field.

4.1.4 Required

Setting this property to True will enforce a validation that requires the field to have a value.

4.1.5 Limit To List

With the property it is possible to set whether the user should be limited to the data items in the drop down. So if set to True, the end user is not allowed to type in their own values that do not match an item in the data source. And if set to False it is allowed to type in any value. An example of the validation of an illegal value is seen in Figure 3.


Please choose an item from the list.

Select the vessel type

My own value

Figure 3 Limit To List validation

4.1.6 Variable Name

This property specifies the name of the Workflow Variable that can prefill the element.

4.2 Adding Dynamic Drop Down Form Element to form

1. Run the Process Designer.
2. Create a new form.
3. Add a Dynamic Drop Down Form Element to the form.

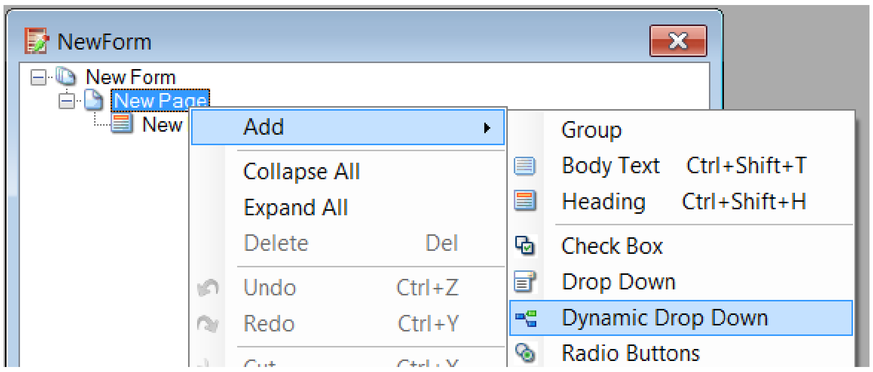
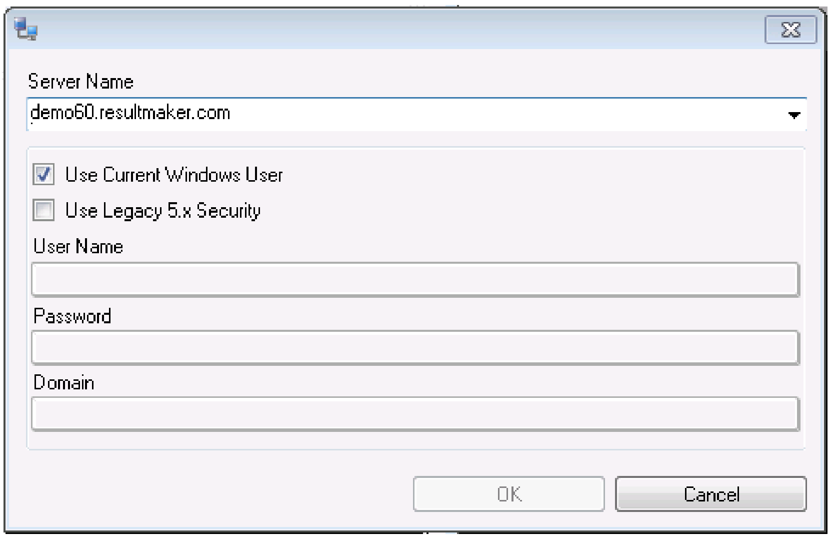


Figure 4 Adding a Dynamic Drop Down Form Element

4. Click Data Source property in the properties window and select the drop down.
5. Select server to use for data source.



Server Name
demo60.resultmaker.com

[✓] Use Current Windows User
[ ] Use Legacy 5.x Security

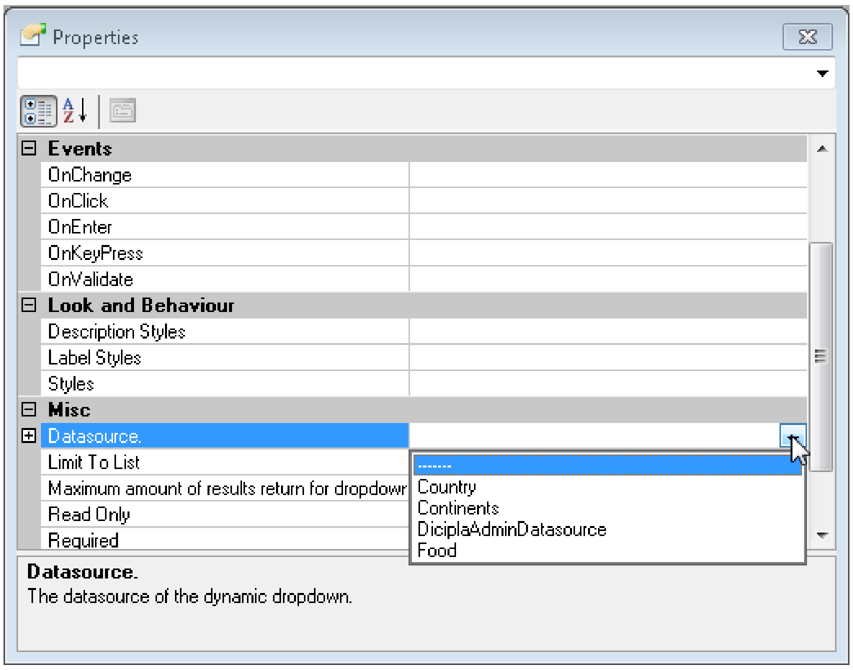
User Name

Password

Domain

Figure 5 Choosing the server for the data source

6. Select one of the installed data sources in the list.



Properties

Events
- OnChange
- OnClick
- OnEnter
- OnKeyPress
- OnValidate

Look and Behaviour
- Description Styles
- Label Styles
- Styles

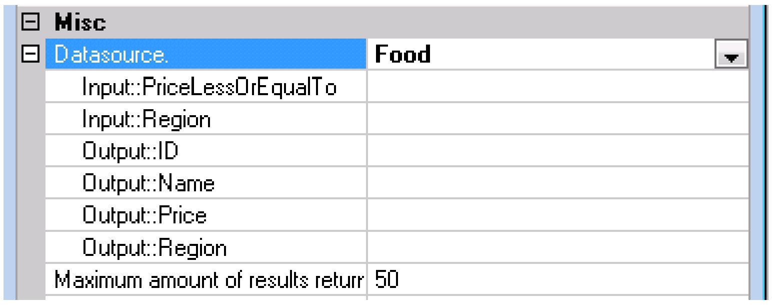
Misc
- Datasource.
- Limit To List
- Maximum amount of results return for dropdowr
- Read Only
- Required

Country
Continents
DiciplaAdminDatasource
Food

Datasource.
The datasource of the dynamic dropdown.

Figure 6 Selecting the DataSource by Data Source name

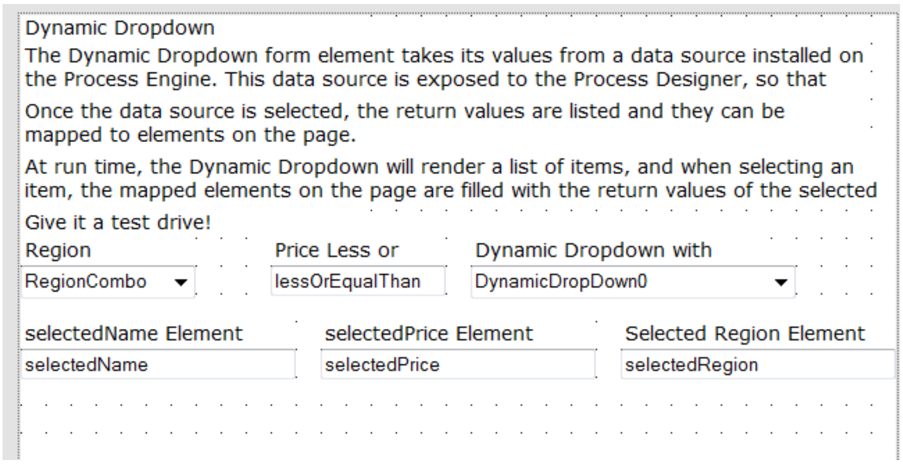
Notice the data source Output attributes and Input parameters in Figure 7.



Misc	
Datasource.	**Food**
Input::PriceLessOrEqualTo	
Input::Region	
Output::ID	
Output::Name	
Output::Price	
Output::Region	
Maximum amount of results returr	50

Figure 7 Data Source input parameters and output attributes

7. Now create text fields with the Internal ID selectedName, selectedPrice, selectedRegion that are to be used for mapping to the Data Source, and give them a relevant value in the Text property.


Dynamic Dropdown

The Dynamic Dropdown form element takes its values from a data source installed on the Process Engine. This data source is exposed to the Process Designer, so that

Once the data source is selected, the return values are listed and they can be mapped to elements on the page.

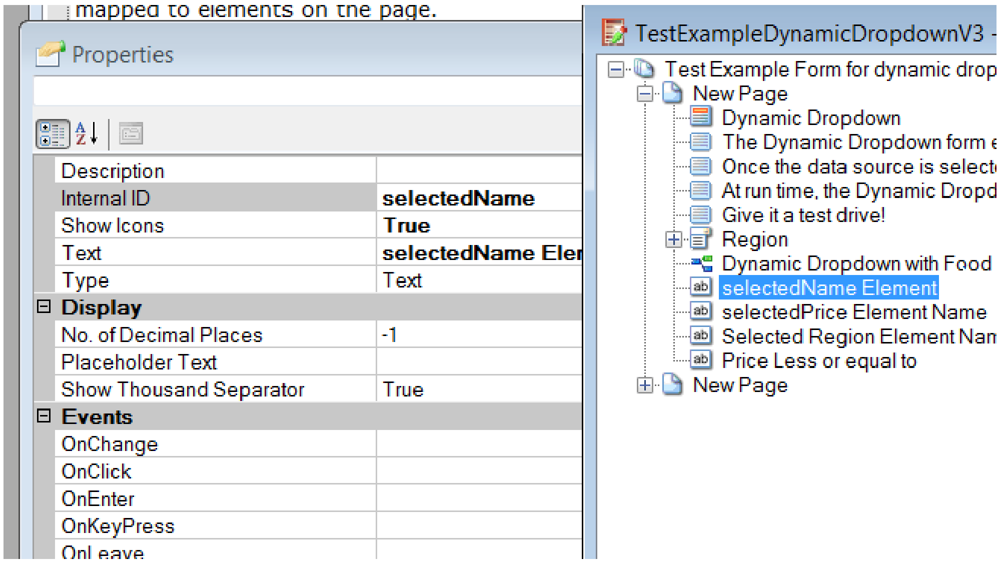
At run time, the Dynamic Dropdown will render a list of items, and when selecting an item, the mapped elements on the page are filled with the return values of the selected

Give it a test drive!

Region	Price Less or	Dynamic Dropdown with
RegionCombo	lessOrEqualThan	DynamicDropDown0

selectedName Element	selectedPrice Element	Selected Region Element
selectedName	selectedPrice	selectedRegion

Figure 8 Form page in layout mode with text elements for mapping


mapped to elements on the page.

Properties

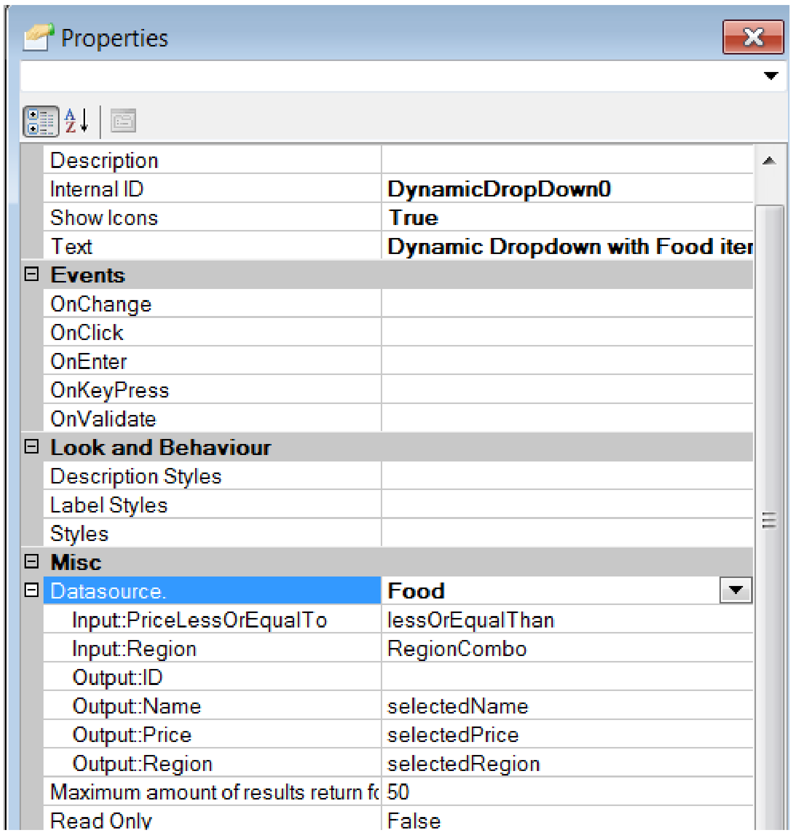
Description	
Internal ID	**selectedName**
Show Icons	**True**
Text	**selectedName Eler**
Type	Text
⊟ **Display**	
No. of Decimal Places	-1
Placeholder Text	
Show Thousand Separator	True
⊟ **Events**	
OnChange	
OnClick	
OnEnter	
OnKeyPress	
OnLeave	

TestExampleDynamicDropdownV3 -

- Test Example Form for dynamic drop
  - New Page
    - Dynamic Dropdown
    - The Dynamic Dropdown form e
    - Once the data source is select
    - At run time, the Dynamic Dropd
    - Give it a test drive!
    - Region
    - Dynamic Dropdown with Food
    - selectedName Element
    - selectedPrice Element Name
    - Selected Region Element Nan
    - Price Less or equal to
  - New Page

Figure 9 Properties on a text field for mapping

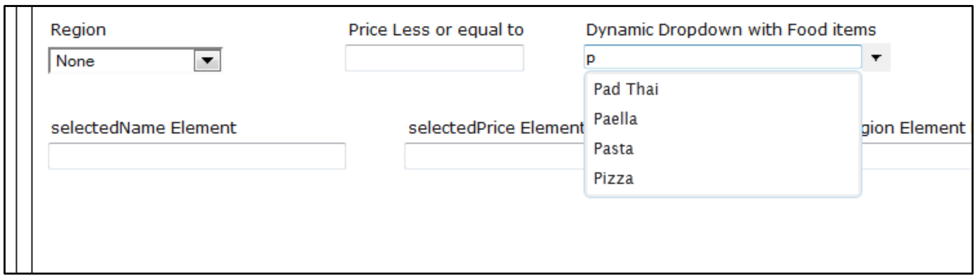
8. Mapping the element names for those attribute names you wish to map.


Properties

Description	
Internal ID	DynamicDropDown0
Show Icons	True
Text	Dynamic Dropdown with Food iter
⊟ Events	
OnChange	
OnClick	
OnEnter	
OnKeyPress	
OnValidate	
⊟ Look and Behaviour	
Description Styles	
Label Styles	
Styles	
⊟ Misc	
⊟ Datasource.	Food
Input::PriceLessOrEqualTo	lessOrEqualThan
Input::Region	RegionCombo
Output::ID	
Output::Name	selectedName
Output::Price	selectedPrice
Output::Region	selectedRegion
Maximum amount of results return fc	50
Read Only	False

Figure 10 Mapping an attribute to a for element
Note the "Input::" prefix is used to indicate that the mapping is an input parameter and the "Output::" prefix for output attributes.

Save the form.
Preview the form in the Process Frontend by pressing F5.
Type "p" in the Dynamic Drop Down Form field.


Region

None

Price Less or equal to

Dynamic Dropdown with Food items

p

Pad Thai
Paella
Pasta
Pizza

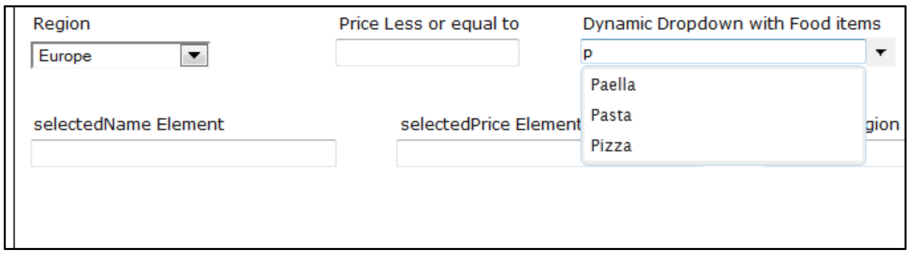
selectedName Element

selectedPrice Element

gion Element

Figure 11 Previewing the Dynamic Drop Down Form Element

You can also filter the results by "Region" and "Price Less or equal to". The Region element is a normal drop down, and here the Internal ID values from the drop down items are used as filtering parameters.


Region

Europe

Price Less or equal to

Dynamic Dropdown with Food items

p

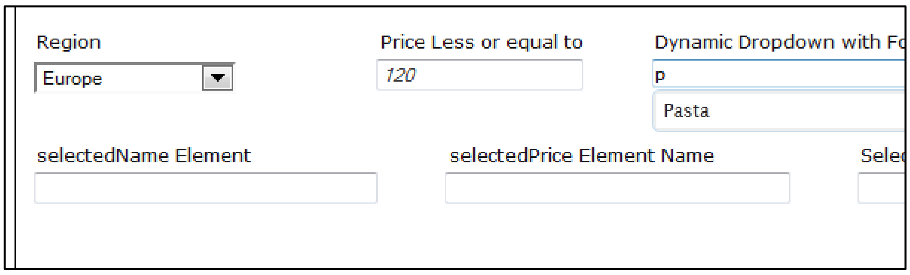
Paella
Pasta
Pizza

selectedName Element

selectedPrice Element

gion

Figure 12 Filtering by the region "Europe" and searching by "p"


Region

Europe

Price Less or equal to

120

Dynamic Dropdown with Fo

p

Pasta

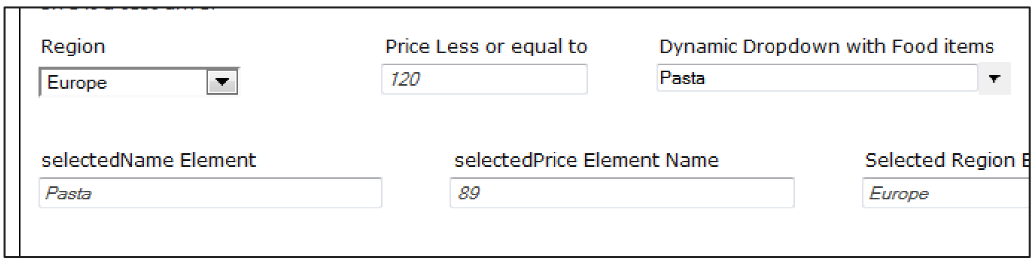
selectedName Element

selectedPrice Element Name

Sele

Figure 13 Filtering by region "Europe" and price less or equal to 120 and searching by the term "p"

Select the item of your choice.
The information is filled in the page accordingly.



Region	Price Less or equal to	Dynamic Dropdown with Food items
Europe	120	Pasta
selectedName Element	selectedPrice Element Name	Selected Region E
Pasta	89	Europe

Figure 14 After selecting an item, the output fields are populated

5 Caching

Beware that the RM Ajax Service at the Process Frontend level has a caching mechanism for the data source data requested. The length of the cache is controlled in the Frontends `CacheMinutesForDropDownWebService` configuration key in the Resultmaker.OC.Frontend section of the Resulmaker.OC.FrontEnd.config. If set to 0, the caching is omitted and lookups are always created.

6 Dynamic Drop Down Data Source deployment

6.1 Deployment location

The Dynamic Drop Down Data Source should be deployed in the Process Engine bin folder for 6.0.5000 and later[1]. For 6.2.3000 and later, it may be deployed in the Ext directory as well, if automatic loading is used.

Prior to 6.2.3000, Dynamic Drop Down Data Sources that are placed in the bin folder are loaded at application start and held for as long as the application lives.

After 6.2.3000, Data Source assemblies must be placed in the Ext directory to be loaded automatically thjs way. Take care not to put any dependencies of the DataSource here, but instead put them in the bin directory. If DataSource assemblies are placed in the bin directory they are no longer loaded automatically, but must instead be declared in the ServiceProviders section of the web.config using StructureMap xml, just like service providers.

If deploying a DataSource implementing the discontinued DataSourceName setter, and if that implementation throws an exception when called (which is a common occurrence since an example did it), the DataSource cannot be deployed in bin but must use automatic registration in Ext.

When a Dynamic Drop Down Data Source assembly is overwritten in the bin folder, the web application recycles as normal. This means that the newly copied assembly is the loaded. This is not the case for assemblies deployed to Ext.

6.2 Legacy deployment

The following only applies to Process Engine 6.0.4000.

You can find the base location in the Process Engine Web.config File, where the modules directory is defined in the Resultmaker.OC element:

<Resultmaker.OC modulesDirectory="C:\Program Files\Common Files\Resultmaker\Online Consultant\Modules">

[1] Process Engine Modules are no longer supported in Process Engine 6.0.5000.

You would then place the data source in the directory: “C:\Program Files\Common Files\Resultmaker\Online Consultant\Modules\MyModuleName\MyModuleName.dll".

After this you need to load the modules. This is done with the Process Engine web service.

1. Log on the the server where the Process Platform is installed.
2. Make an IISReset
    a. Click Start type “IISReset”
3. Open the browser and type in the address: http://localhost/oc4/oc.asmx.
4. Find the LoadNewModules web service method and click it.
5. On the next page click Invoke.
6. If you get an error perform step 2-6 again.

7 Dynamic Drop Down security configuration

The Dynamic Drop Down infrastructure is built on WCF, and has to be configured according to the IIS security setup under which the Process Platform is being installed. This section describes the following security scenarios:

- Windows Authentication
- Anonymous Authentication

When the Process Platform is configured to expose the Frontend under Windows Authentication the following WCF binding configurations for *security mode* and *clientCredentialType* have to be set. Notice that there is a binding for both HTTP and HTTPS.

```
<system.serviceModel>
  <bindings>
    <webHttpBinding>
      <binding name="HttpBinding">
        <security mode="TransportCredentialOnly">
          <transport clientCredentialType="Windows" />
        </security>
      </binding>
      <binding name="HttpsBinding">
        <security mode="Transport">
          <transport clientCredentialType="Windows" />
        </security>
      </binding>
    </webHttpBinding>
  </bindings>
  …
</system.serviceModel>
```

When the Process Platform is configured to expose the Frontend under Anonymous Authentication, the following WCF binding configurations for *security mode* and *clientCredentialType* have to be set.

```
<system.serviceModel>
  <bindings>
```

```
    <webHttpBinding>
      <binding name="HttpBinding">
        <security mode="None">
          <transport clientCredentialType="None" />
        </security>
      </binding>
      <binding name="HttpsBinding">
        <security mode="Transport">
          <transport clientCredentialType="None" />
        </security>
      </binding>
    </webHttpBinding>
  </bindings>
  …
</system.serviceModel>
```

This is part of the default installation of Process Platform