

Providing a Suggestion List for Object Value Selector Using IWDOVSSuggester



Applies to:

Applies to SAP NetWeaver Composition Environment 7.1 EHP 1. For more information, visit the [Web Dynpro Java homepage](#).

Summary

This article explains about providing a suggestion List for Object Value Selector using IWDOVSSuggester.

Author: FAZAL AHAMED

Company: HCL Technologies Ltd.

Created on: 23 September 2009

Author Bio

Fazal Ahamed is working as Java developer in HCL Technologies Limited on SAP NetWeaver Technology (Web Dynpro Java, CAF, Guided Procedures and BPM).

Table of Contents

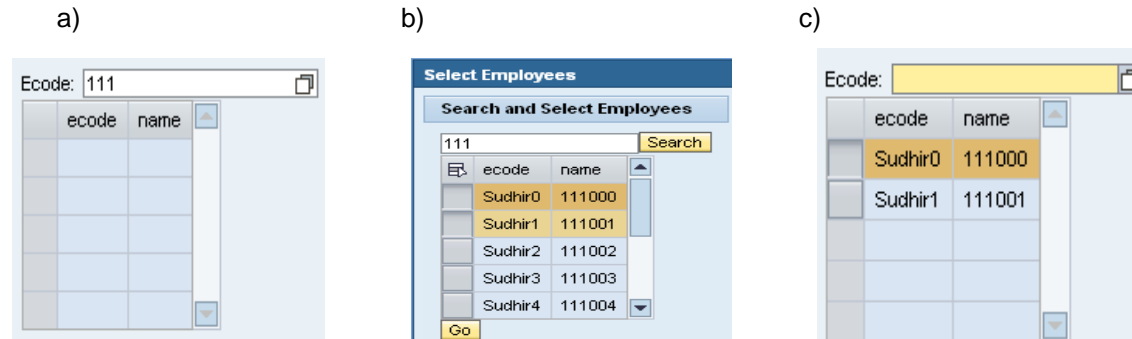
Creating a Custom Value Help	3
Adding Suggestions List to OVS using IWDOVSSuggester	3
Output.....	6
Providing the suggestion list when the user enter some character	6
Note:	8
Related Content.....	9
Disclaimer and Liability Notice.....	10

Creating a Custom Value Help

<http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/401975e3-e2ee-2b10-e191-a8140aabe9d6&overridelayout=true>

Following the above article, I have created a custom Value Help and imported the Employee Model for performing employees search.

Output of the employee value help is shown below.



Adding Suggestions List to OVS using IWDOVSSuggester

We can provide the suggestion list for the input field which triggers the OVS Value Help. For this we need to set the **“suggestValues”** property of the input field to True.

To provide Suggestion list for an OVS, we need to implement IWDOVSSuggester class.

The IWDOVSSuggester interface has two methods as shown below:

Method Name	Description	Return Type
<pre>suggest(IWDOVSControl ovsControl, String filter, IWDSuggestionList suggestions)</pre>	<p>Determines the List of Suggestions based on a given <code>filter</code>.</p> <p>@param ovsControl – the Information about the OVS</p> <p>@param filter – the filter value</p> <p>@param suggestions – the list of suggestions to be filled</p>	void
<pre>finalize(IWDOVSControl ovsControl, String identifier) {</pre>	<p>Finalizes the selection.</p> <p>@param ovsControl – the information about OVS</p> <p>@param identifier – the selected value (one of the <code> identifier </code> values that were put into the suggestion list) or <code> null </code> if nothing is selected.</p>	void

IWDSuggestionList is an Interface which gives a list of suggestion which **IWDOVSSuggester** has to fill.

Summary of methods present in IWDSuggestionList interface:

Method Name	Description	Return Type
addSuggestion(String key, String description, String identifier)	<p>Adds a suggestion to the list.</p> <p>@param key – key to display in the suggestion</p> <p>@param description – the description to display in the suggestion</p> <p>@param identifier – the value by which the IWDOVSSuggester can identify the selection.</p>	void
getLimit()	Returns the maximum number of suggestions that this list will accept.	int

Note: Please ensure that you have changed the **suggestValues** Property of the Input Field to **True**.

Copy the below code in wdDoInit method of component controller:

```
public void wdDoInit()
{
    ///@begin wdDoInit()

        WdValueServices
            .addOVSExtension("Employee Value Help",
                new IWDAttributeInfo[] { wdContext.nodeEmployeeNum()
                    .getNodeInfo().getAttribute(
                        IEmployeeNumElement.FOCDE) },
                new MyOVSPProvider(wdComponentAPI, wdThis
                    .wdGetSuggesterCompUsageComponentUsage()),
                new MyOVSSuggester());

    ///@end
}
```

Copy the below code in begin others section of component controller.

```

public class MyOVSSuggester implements IWDOVSSuggester {
    public void suggest(IWDOVSControl ovsControl, String filter,
        IWDSuggestionList suggestions) {

        wdThis.findEmployeeByPattern(filter);
        int empSize = wdContext.nodeEmployee().size();
        if (empSize > 0) {
            for (int i = 0; i < empSize; i++) {
                IEmployeeElement element = wdContext.nodeEmployee()
                    .getEmployeeElementAt(i);
                Integer index = element.index();
                suggestions.addSuggestion(element.getEcode(), element
                    .getName(), index.toString());
            }
        } else {
            suggestions = null;
        }
    }

    public void finalize(IWDOVSControl ovsControl, String identifier) {
        if (identifier != null) {
            int index = Integer.parseInt(identifier);
            IEmployeeElement result = wdContext.nodeEmployee()
                .getEmployeeElementAt(index);
            IEmployeeResult_1Element element = wdContext
                .nodeEmployeeResult_1().createEmployeeResult_1Element();
            element.setEcode(result.getEcode());
            element.setName(result.getName());
            wdContext.nodeEmployeeResult_1().addElement(element);
        }
    }
}
//@@@end
}

```

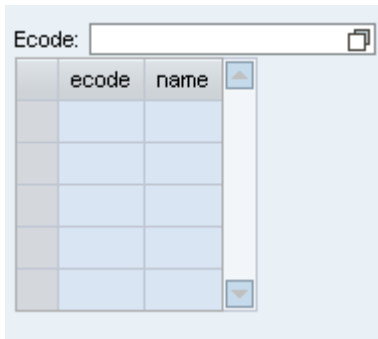
In the above code, the class **MyOVSSuggester** implements **IWDOVSSuggester** interface and overrides the methods **suggest** and **finalize** present in the interface.

Method **suggest** is called when the user types some character in the Input Field and based on the filter value (characters entered by the user), Employee Model returns the Employee list which are populated in the IWDSuggestionList using **addSuggestion** method.

This Suggestion list is displayed in the input field where the user can select any value from the suggestion list. If the user select any value from the suggestion list, the selected value is populated to the Table in the **finalize** method using **Identifier** parameter.

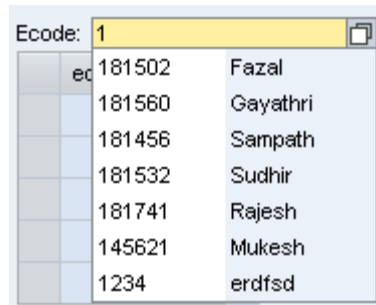
Output

Finally deploy and run the application to get the below Output.

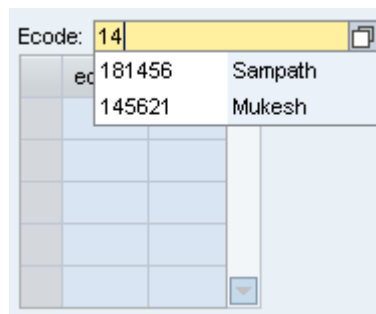


Providing the suggestion list when the user enter some character

- When the User enters some character in the input field, method **suggest** is called.
- The method parameter **filter** holds the string entered by the user.
- Based on the filter value, employee model returns the employee list and this list are added to the **IWDSuggestionList** using **addSuggestion** method.
- IWDSuggestionList holds the employee List and **identifier** for each employee element.
- **Identifier** attribute holds the unique value for each employee element present in the suggestion list.

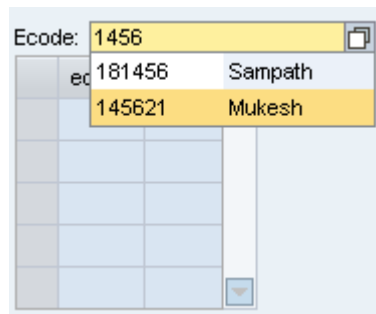


- If the user doesn't select any value from suggestion list and gives some other search string.
- Method **finalize** is called where Identifier is null and again the suggest method is called to populate the suggestion list for new search string.



Populating the selected value from the suggestion list to Table

- On selecting any one of values from the suggestion list, **finalize** method is called.



- Using Identifier parameter, the selected element is populated to the table.

Ecode:

ecode	name
145621	Mukesh

Note:

New Suggestion Support in CE 7.1 EhP1 Potentially Increases Number of Roundtrips!



Beware the increased number of server roundtrips for suggestion-enabled (`suggestValues = true`) Input Fields in the following case:



- The InputField suggestion is based on the **OVS valuehelp**:
see **IWDOVSSuggester**, **IWDSuggestionList**
- In this case the client repeatedly calls the server to query and send back a list of matching suggestions on-demand.

There is also an **increased number of server roundtrips** when the InputField suggestion is based on **on-demand valuesets or static valuesets** defined in dictionary simple types **with MORE than 200 values**:

- A matching “chunk” valueset with 200 values is retrieved in a **first roundtrip**.
- In **additional roundtrips** new valuesets matching the user input are retrieved from the server successively.
- see **WDSimpleValueSupplier**



[Best Practices for optimizing Web Dynpro Java application Performance](#)

The number of values added to suggestion list can be got from **getLimit** method of interface **IWDSuggestionList**. If the user is adding more elements to this list, then the suggestion list may not show the complete suggest values and the user may think that the object is not there in Backend.

Related Content

1. <http://help.sap.com/javadocs/nwce/current/wdr/com.sap.wdr.com/sap/tc/webdynpro/progmodel/api/IWDOVSSuggester.html>
2. <http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/com.sap.km.cm.docs/media/streamingmedia/events/sap-teched-2008/Virtual%20SAP%20TechEd%2008/composition-and-application-development/COMP204%20Best%20Practices%20for%20Optimizing%20Web%20Dynpro%20Java%20Application%20Performance/COMP204.pdf>
3. [http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/e0e682bd-b59e-2b10-cd82-fa175f502fe6;jsessionid=\(J2EE3414800\)ID0311666150DB10781511139449274109End?overridelayout=true](http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/e0e682bd-b59e-2b10-cd82-fa175f502fe6;jsessionid=(J2EE3414800)ID0311666150DB10781511139449274109End?overridelayout=true)
4. <http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/401975e3-e2ee-2b10-e191-a8140aabe9d6&overridelayout=true>
5. <https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/c0c6132f-4ea3-2b10-4d8f-e68d2e96113d>
6. For more information, visit the [Web Dynpro Java homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.