Supply Function Method Usage in Web Dynpro Application

Applies to:
SAP ECC 6 and above.

Summary:
This document will help in learning the Supply Function Method usage in Web Dynpro Application. This will explain its usage with one simple application in easy steps.

Author: Vineethkumar Yogimath
Company: Infosys
Created on: 29 September 2010

Author Bio
Vineethkumar Yogimath is working with Infosys for the past 2 years in SAP ABAP technology. His experience is in SAP with variety of work in ABAP. Mainly on Reports, FMs, Interfaces, BAPIs, SAP Scripts, performance analysis etc. As a part of small development he also learned WebDynpro.
# Table of Contents

Background .......................................................................................................................... 3
Introduction ............................................................................................................................ 3
Creating a Web Dynpro ABAP Application that Demonstrate the Supply Function Method .................................................. 4
Steps: .................................................................................................................................................. 4
Related Content ..................................................................................................................... 21
Disclaimer and Liability Notice .......................................................................................... 22
**Background**

Before going to the main objective of this document to show the usage of Supply Function method in WebDynpro application, we need to understand some of the properties which are useful in using the Supply Function method.

**Cardinality Property of context node**: This property defines the context node, number of elements of this node available at runtime. The first part gives minimum number of elements and second part gives the maximum number of elements that a node can contain
- **1..1**: only one element.
- **0..1**: at maximum one element at runtime.
- **1..n**: n number of elements can be there, but one should be there.
- **0..n**: n number of elements can be there with no condition of one should be there.

**Singleton Property**: It is the one, which affects the relationship between child node and parent node. The contents of the child node are changed every time, as and when the lead selection in the parent node changes. So the Singleton child node must be repopulated every time.

For each newly created context node, lead selection is initialized automatically, but this setting can also be deactivated. However, lead selection must generally be initialized for every context node.

An important difference of **singleton node** with **non singleton node** is, when the node is declared as singleton there will be only one instance of the node irrespective of the number of elements found in the parent node's collection. This is not case in non singleton node. The latter takes as many instances as there are number of elements in the parent node's element collection.

The WebDynpro programming model provides the concept of Supply Functions to fill the context nodes at runtime, which can be defined by assigning a method name to the node property Supply Function in the controller at design time.

Supply Function is usually called when the context node is not filled yet or the context node has been invalidated in previous step. Supply function and context node have **1..1** relationship. i.e., they are specific to individual context nodes.

**Introduction**

**Supply Function**: Thus they are used to repopulate child nodes when the lead selection in the parent node changes.

Points on Supply Function:
1) It is used to populate entire node collection.
2) It should not be used to manipulate an existing node collection.
3) Any context node can have supply function defined for it.
4) The existence of a supply function is defined declaratively.
5) It is called automatically by the Web Dynpro Runtime when an attempt is made to read an empty node collection. This situation occurs for following condition.
   - The node is never populated before.
   - The lead selection of the parent node is changed.
   - Application coding has explicitly invalidated the element collection.

Note: The Supply Function of the node should never access any context node that is also populated by Supply Function. Otherwise it will cause runtime exception.
Creating a Web Dynpro ABAP Application that Demonstrate the Supply Function Method

In this application, the header details are selected from VBAK table and displayed in ALV for the inputs given by the user for the two fields VKORG (Sales Organisation) and AUART (Document type). The Supply function method populates the corresponding Item details in ALV table when user selects a record of the displayed header table.

Below are the steps to meet the objective of this document of using Supply Function method in WebDynpro Application in the case of user selecting the lead selection of parent node.

Steps:

1) Go to transaction SE80 and create the new WebDynpro Application by selecting it in the drop down list.

2) Double click on the COMPONENT CONTROLLER, to create context nodes/attributes, which are used in anywhere in the WebDynpro Application.
3) Create a node/attribute as shown below.

4) A popup will display on the screen, fill the attribute name (VKORG), type and then click on 'Additional Attribute' to add another attribute AUART. These two attributes will be used as inputs provided by the user.
5) In a similar way, create the node HEADER and fill the dictionary name and other properties. Click on the 'Add Attribute from Structure'.
6) Select few fields from VBAK structure, which are attributes of the node HEADER.

```

<table>
<thead>
<tr>
<th>Component</th>
<th>Key</th>
<th>RTy</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Decim</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANUT</td>
<td>✓</td>
<td>✓</td>
<td>MANUT</td>
<td>CLNT</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>VELLM</td>
<td>✓</td>
<td>✓</td>
<td>VELLM V</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>ERPAT</td>
<td>✓</td>
<td>✓</td>
<td>ERPAT</td>
<td>DATS</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>ERZET</td>
<td>✓</td>
<td>✓</td>
<td>ERZET</td>
<td>TIMS</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>ERNAM</td>
<td>✓</td>
<td>✓</td>
<td>ERNAM</td>
<td>CHAR</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>ANSET</td>
<td>✓</td>
<td>✓</td>
<td>ANSET</td>
<td>DATS</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>BNDT</td>
<td>✓</td>
<td>✓</td>
<td>BNDT</td>
<td>DATS</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>AUCAT</td>
<td>✓</td>
<td>✓</td>
<td>AUCAT</td>
<td>DATS</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>MVCYP</td>
<td>✓</td>
<td>✓</td>
<td>MVCYP</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>TRIVG</td>
<td>✓</td>
<td>✓</td>
<td>TRIVG</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>KU1ART</td>
<td>✓</td>
<td>✓</td>
<td>KU1ART</td>
<td>CHAR</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>AU1GERU</td>
<td>✓</td>
<td>✓</td>
<td>AU1GERU</td>
<td>CHAR</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>NW01T</td>
<td>✓</td>
<td>✓</td>
<td>NW01T</td>
<td>DATS</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>SUB1I</td>
<td>✓</td>
<td>✓</td>
<td>SUB1I</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>LIFSK</td>
<td>✓</td>
<td>✓</td>
<td>LIFSK</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>FAKSK</td>
<td>✓</td>
<td>✓</td>
<td>FAKSK</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>NETWR</td>
<td>✓</td>
<td>✓</td>
<td>NETWR</td>
<td>CURR</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>WAREK</td>
<td>✓</td>
<td>✓</td>
<td>WAREK</td>
<td>CUKY</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>VKORG</td>
<td>✓</td>
<td>✓</td>
<td>VKORG</td>
<td>CHAR</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>VTWEG</td>
<td>✓</td>
<td>✓</td>
<td>VTWEG</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>SPART</td>
<td>✓</td>
<td>✓</td>
<td>SPART</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>VKGEP</td>
<td>✓</td>
<td>✓</td>
<td>VKGEP</td>
<td>CHAR</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>VKHUR</td>
<td>✓</td>
<td>✓</td>
<td>VKHUR</td>
<td>CHAR</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>SEEBER</td>
<td>✓</td>
<td>✓</td>
<td>SEEBER</td>
<td>CHAR</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
```

7) Create a node ITEM under node HEADER as shown below where the HEADER is the source of supply for the node ITEM.

```

Web Dynpro Explorer: Change Controller for ZWD_SUPPLY_TEST
```

© 2010 SAP AG
8) Fill the details for node ITEM and choose few fields from the VBAP structure similarly as we did for node. In our case, the supply function method is get_itemdata.

After that screen looks like as below.
9) The next step is to create a view. In our case we make use of default MAIN view. Go to the context tab.

10) In this step context mapping happens. Map the specific context nodes from COMPONENT CONTROLLER with the view context. Just drag and drop from the right side to the left side view context or right click on the view context and choose 'Copy Nodes of Different Context'.

11) In the pop up display, choose controller name using F4 help and set the context elements to be copied as shown below.

12) Go to the LAYOUT tab and right click on the ROOTELEMENTCONTAINER and click insert element.
13) A pop up will appear, fill the details for first input field VKORG.

14) Create on the similar line the second input field AUART and captions for both these input fields.

15) Then create a button for action and in the properties of the button, type on the TEXT property to get caption.

In the properties, click on the create button of the EVENT property. Fill the details in the pop up display.
16) Create the element for HEADER with type as TABLE.

Create the binding for the fields in the HEADER table with context node HEADER.
17) Similarly repeat the steps for binding the ITEM node with the view.
18) After saving the MAIN view, click on the component name and add the ALV component ‘ALV_TAB’ as shown.
19) Now go to the CONTROLLER USAGE and then INTERFACECONTROLLER_USAGE.

20) Click on the CONTROLLER USAGE button and drag & drop the HEADER node into the DATA node to create mapping with ALV component added.
21) Now click on the WINDOWS and here the step is to embed the Table view of the ALV_TABLE over the View.

22) Then choose the ‘ALV_TAB’ component after clicking on the F4 help for ‘View TO BE Embedded’.

23) After saving, create the application and enter the details as shown below.
24) Coding for the methods, which are created.
The ‘parent_element’ is a reference to the lead selection element of the parent node. When user selects any different element, the value of the ‘parent_element’ changes. The Supply function is called to fill new elements into the table Item based on the selected element from the Header table.

25) Activate the whole application if there are no errors. Check the output below for the input given.

The Header table is bounded to context node HEADER, and item table to the node ITEM.

When the user inputs and clicks on ‘Fetchdata’, the method ‘Onactiongetdata’ fills the header node from the VBAK table. The lead selection of the context node has been created on the first element of the node. All the line items for the Sales Document selected are populated on the second table Item from the table VBAP. This is done by the Supply function ‘get_itemdata’.

When the user selects another record from the HEADER table. The content of the Item table changes automatically.
### SalesOrg 5600, DocType ZSB

<table>
<thead>
<tr>
<th>Sales Document</th>
<th>Created on</th>
<th>Time</th>
<th>Sales Doc. Type</th>
<th>Sales Org.</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000884</td>
<td>30.05.2007</td>
<td>16:02:23</td>
<td>ZSB</td>
<td>5600</td>
</tr>
<tr>
<td>40000885</td>
<td>01.06.2007</td>
<td>09:36:11</td>
<td>ZSB</td>
<td>5600</td>
</tr>
<tr>
<td>40000964</td>
<td>14.07.2007</td>
<td>07:30:36</td>
<td>ZSB</td>
<td>5600</td>
</tr>
<tr>
<td>40000965</td>
<td>14.07.2007</td>
<td>08:36:37</td>
<td>ZSB</td>
<td>5600</td>
</tr>
<tr>
<td>40000966</td>
<td>14.07.2007</td>
<td>08:42:08</td>
<td>ZSB</td>
<td>5600</td>
</tr>
</tbody>
</table>

### ItemDetails

<table>
<thead>
<tr>
<th>Sales Document</th>
<th>Item</th>
<th>Material</th>
<th>Material Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000885</td>
<td>00010</td>
<td>D2831Z/A</td>
<td>TNG00</td>
<td>TRAINING UNITS-INT</td>
</tr>
<tr>
<td>40000885</td>
<td>00020</td>
<td>D2831Z/A</td>
<td>TNG00</td>
<td>TRAINING UNITS-INT</td>
</tr>
</tbody>
</table>
Related Content

sap.sdn.com
help.sap.com
saptechnical.com
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.