

Web Dynpro ABAP: UI Element Tree



Applies to:

SAP ECC 6.0. For more information, visit the [Web Dynpro ABAP homepage](#)

Summary

The article tells about displaying data in Tree structure. Recursion Node is used to build the tree structure and supply function is to display data based on the item selection.

Author: J.Jayanthi

Company: Siemens IT Solutions and Services Pvt. Ltd.

Created on: 02 November 2010

Author Bio

J.Jayanthi is a Certified ABAP consultant with HR ABAP knowledge.

Table of Contents

Tree in Web Dynpro ABAP	3
Prerequisites	3
Creating Web Dynpro	3
View	4
Embedding View	13
Creating Web Dynpro Application	13
Code	14
Output	15
Related Content	16
Disclaimer and Liability Notice	17

Tree in Web Dynpro ABAP

ABAP Consultants might have worked with Tree model.

We are going to see how to display data in Tree structure. The objective is to display the employees reporting to a manager in Tree structure.

Prerequisites

View

The view is the smallest unit of a Web Dynpro application visible for the user. The layout elements and dialog elements - for example, tables, text fields, or buttons - required for the application are arranged in a view. The view contains a controller and a controller context in which the application data to be processed is stored in a hierarchical structure. This allows the linking of the graphical elements with the application data.

Window

A window is used to group multiple views and to specify the navigation between the views. A view can only be displayed by the browser if the view is embedded in a window.

Singleton

The property "Singleton" specifies the number of instances that can exist in a dependent context node (that does not belong directly to the root node). If the property "Singleton" is set, exactly one instance of the node exists. Its content changes when the lead selection of the parent node changes. If the property "Singleton" is not set, one instance per parent instance exists. The content of the instances does not change when the lead selection of the parent changes.

Supply Function

We can specify the name of a method which we can use to fill the context node at runtime in supply function.

Creating Web Dynpro

Go to SE80 and select Web Dynpro Comp./Intf. and provide the name (say ZZZ_JAYTEST12) to create. Then enter the description and choose the type as Web Dynpro Component.

View

Click the context tab. Then select Create Node to create Parent Node Manager.

Property	Value
Nodes	
Node Name	MANAGER
Dictionary structure	
Cardinality	0..n
Selection	0..1
Initialization Lead Selection	<input checked="" type="checkbox"/>
Singleton	<input type="checkbox"/>
Supply Function	

Then right click the node and create attribute PERNR.

Property	Value
Attribute	
Attribute Name	PERNR
Type assignment	Type
Type	PA0002-PERNR
Read-only	<input type="checkbox"/>
Default Value	
Null Value	<input type="checkbox"/>
Input Help Mode	Automatic
Determined Input Help	PA0003
Type of Input Help	Check table
Formatting	
Display Leading Zeros	Default Value

Then right click the node Manager and create Recursion node Employee.

The screenshot shows the UI Element Tree with a node 'EMPLOYEE' selected under 'MANAGER'. Below the tree is a table for configuring the recursion node.

Property	Value	Repeated
Recursion Node		
Node Name	EMPLOYEE	
Repeated Node	MAIN.MANAGER	<input type="checkbox"/>

Then create another node Workschedule to display the details under the node Manager with attributes as shown below. (NACHN and VORNA are added from PA0002 table and other attributes are added from PA0007 table).

The screenshot shows the UI Element Tree with a node 'WORKSCHEDULE' selected under 'MANAGER'. Below the tree is a table for configuring the child node.

Property	Value
Nodes	
Node Name	WORKSCHEDULE
Dictionary structure	
Cardinality	1..1
Selection	0..1
Initialization Lead Selection	<input checked="" type="checkbox"/>
Singleton	<input checked="" type="checkbox"/>
Supply Function	GET_DETAILS

Here we enabled singleton property so that there can be one instance of the child node WORKSCHEDULE at runtime.

Double click the method GET_DETAILS and code will be automatically generated with comment. Keep the required code as below and write logic for filling child node.

```

method GET_DETAILS .
*   if necessary, get static attributes of parent element
  DATA ls_parent_attributes TYPE wd_this->element_manager.
  parent_element->get_static_attributes(
    IMPORTING
      static_attributes = ls_parent_attributes ).
DATA ls_workschedule TYPE wd_this->Element_workschedule.
* From Pa0002
select single vorna nachn from pa0002
into (ls_workschedule-vorna, ls_workschedule-nachn)
where pernr eq ls_parent_attributes-pernr
and begda le sy-datum
and endda ge sy-datum.
* From Pa0007
select single schkz mostd wostd from pa0007
into (ls_workschedule-schkz,
ls_workschedule-mostd,
ls_workschedule-wostd)
where pernr eq ls_parent_attributes-pernr
and begda le sy-datum
and endda ge sy-datum.
ls_workschedule-pernr = ls_parent_attributes-pernr.
* bind a single element
node->bind_structure(
  new_item      = ls_workschedule
  set_initial_elements = abap_true ).
endmethod.

```

We are going to design layout as follows.

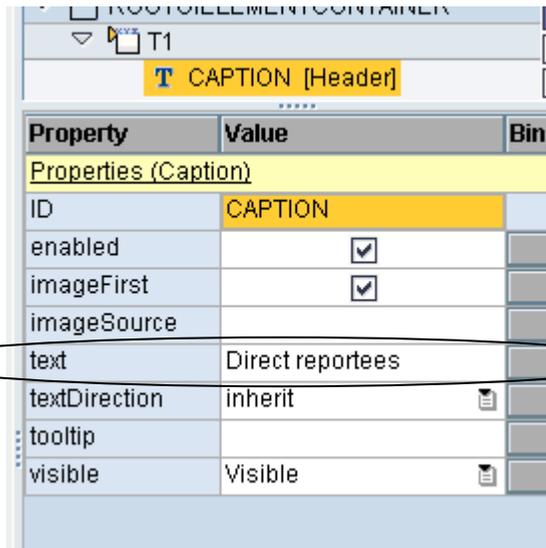
There will be 2 trays. In first tray, tree structure will be displayed. In second tray, details of employees will be displayed. Transparent container will be used to separate the trays.

Then in layout, set the properties for ROOTUIELEMENTCONTAINER as follows.

Property	Value	Bin...
Properties (TransparentContainer)		
ID	ROOTUIELEMENTCONTAINER	
Layout	FlowLayout	
accessibilityDesc		
defaultButtonId		
enabled	<input checked="" type="checkbox"/>	
height		
isLayoutContainer	<input checked="" type="checkbox"/>	
scrollingMode	none	
tooltip		
visible	Visible	
width		
Layout (FlowLayout)		
wrapping	<input checked="" type="checkbox"/>	

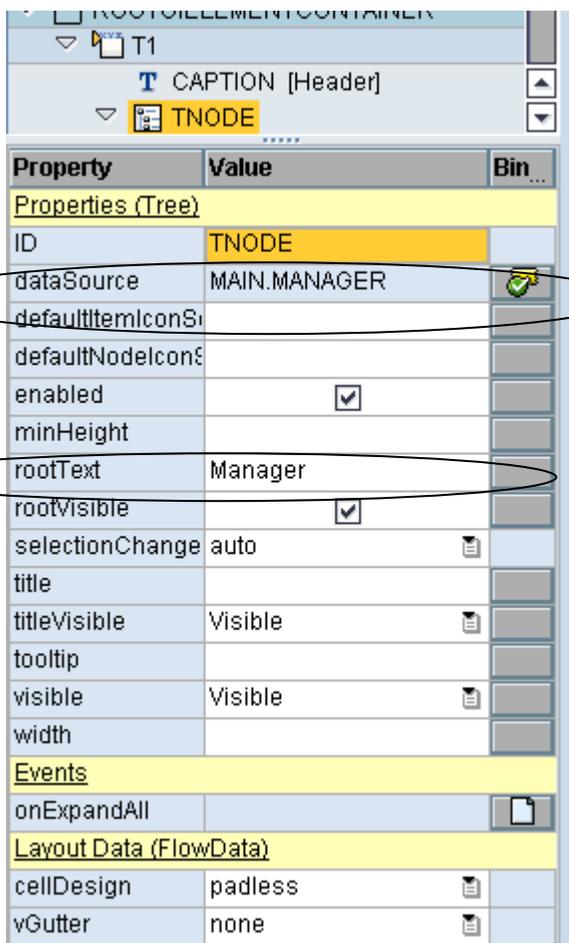
Right click the ROOTUIELEMENTCONTAINER and then choose Insert element. Create a Tray T1 with Flow Layout in properties.

Change the caption of the tray as required.



Property	Value	Bin
Properties (Caption)		
ID	CAPTION	
enabled	<input checked="" type="checkbox"/>	
imageFirst	<input checked="" type="checkbox"/>	
imageSource		
text	Direct reportees	
textDirection	inherit	
tooltip		
visible	Visible	

Create tree node TNODE inside Tray T1.



Property	Value	Bin...
Properties (Tree)		
ID	TNODE	
dataSource	MAIN.MANAGER	<input checked="" type="checkbox"/>
defaultItemIconSize		
defaultNodeIconSize		
enabled	<input checked="" type="checkbox"/>	
minHeight		
rootText	Manager	
rootVisible	<input checked="" type="checkbox"/>	
selectionChange	auto	
title		
titleVisible	Visible	
tooltip		
visible	Visible	
width		
Events		
onExpandAll		
Layout Data (FlowData)		
cellDesign	padless	
vGutter	none	

Inside Tree, create Treeltem type with id TREEITEM.

Property	Value	Bin...
Properties (TreeltemType)		
ID	TITEM	
dataSource	MAIN.MANAGER	
design	standard	
iconSource		
ignoreAction	<input type="checkbox"/>	
text	MAIN.MANAGER.PERNR	
textDirection	inherit	
tooltip		
Events		
onAction		

Note that for data source, main node is chosen and for text PERNR which appears under node Manager directly. This text will make sure the employees number will be displayed at runtime.

Now create transparent container TC with height 100PX.

Property	Value	Bin...
Properties (TransparentContainer)		
ID	TC	
Layout	FlowLayout	
accessibilityDesc		
defaultButtonId		
enabled	<input checked="" type="checkbox"/>	
height	100px	
isLayoutContain	<input checked="" type="checkbox"/>	
scrollingMode	none	
tooltip		
visible	Visible	
width		
Layout (FlowLayout)		
wrapping	<input checked="" type="checkbox"/>	
Layout Data (FlowData)		
cellDesign	padless	
vGutter	none	

Create another Tray T2 with Input fields and labels as below.

▼ 📄 **T2**

- T** CAPTION_1 [Header]
- T** L1
- REC PERNR
- T** L4
- REC SCHKZ

Property	Value	Bin...
Properties (Tray)		
ID	T2	
Layout	MatrixLayout 📄	
accessibilityDes		
defaultButtonId		
design	transparent 📄	
enabled	<input checked="" type="checkbox"/>	
expanded	<input checked="" type="checkbox"/>	
hasContentPad	<input checked="" type="checkbox"/>	
height		
scrollingMode	none 📄	
tooltip		
visible	Visible 📄	
width		
Events		
onToggle		📄
Layout (MatrixLayout)		
stretchedHorizo	<input type="checkbox"/>	
stretchedVertica	<input type="checkbox"/>	

Property	Value	Bin...
Properties (Label)		
ID	L1	
Layout Data	MatrixHeadData	
design	standard	
enabled	<input checked="" type="checkbox"/>	
labelFor	PERNR	
text	Employee Number	
textDirection	inherit	
tooltip		
visible	Visible	
width		
wrapping	<input type="checkbox"/>	
Layout Data (MatrixHeadData)		
cellBackground	transparent	
cellDesign	rPad	
colSpan	1	
height		
hAlign	beginOfLine	
vAlign	baseline	

Change the text for each label as desired. Bind the value of Input field by right click.

The screenshot shows the 'Properties (InputField)' window in SAP Web Dynpro. The 'value' property is highlighted with a red oval and set to 'MAIN.MANAGER.WORKS'. Other visible properties include 'D' (PERNR), 'Layout Data' (MatrixData), 'alignment' (auto), 'enabled' (checked), 'length' (0), 'passwordField' (unchecked), 'readOnly' (unchecked), 'state' (Normal Item), 'textDirection' (inherit), 'tooltip', 'visible' (Visible), and 'width'. The 'Events' section shows 'onEnter' and the 'Layout Data (MatrixData)' section shows 'cellBackground' (transparent).

Property	Value	Bin...
Properties (InputField)		
D	PERNR	
Layout Data	MatrixData	
alignment	auto	
enabled	<input checked="" type="checkbox"/>	
explanation		
length	0	
passwordField	<input type="checkbox"/>	
readOnly	<input type="checkbox"/>	
state	Normal Item	
textDirection	inherit	
tooltip		
value	MAIN.MANAGER.WORKS	
visible	Visible	
width		
Events		
onEnter		
Layout Data (MatrixData)		
cellBackground	transparent	

Finally layout will appear as below.

The screenshot shows the SAP Web Dynpro layout preview. The 'Layout' tab is active, showing a tree structure with 'VORNA' highlighted. The 'Employee Details' section contains input fields for 'Employee Number', 'Work Schedule', 'Last Name', 'First Name', 'Weekly Hours', and 'Monthly Hours', all with the value 'MAIN.MANAGER.WORKS'. The 'Direct reportees' section is also visible.

Tree Structure:

- CONTEXT_MENUS
- ROOTUIELEMENTCONTAINER
 - T1
 - CAPTION [Header]
 - TNODE
 - TITEM
 - TC
 - T2
 - CAPTION_1 [Header]
 - L1
 - PERNR
 - L4
 - SCHKZ
 - L2
 - NACHN
 - L3
 - VORNA**
 - L5
 - MOSTD
 - L6
 - WOSTD

For the labels of Employee number, Last name and Weekly Hours, define the layout as Matrix Head Data and for rest others, define it as Matrix Data. The reason behind this, Matrix Head data will make it as first in the row.

Make sure for all the input fields, Value in the properties is binded using the right click for context binding. Select the method WDD0INIT (For Address Button) in methods tab.



Use Web Dynpro code wizard to generate code automatically.

Step a:

Choose the radio button Read context for the node MAKT.

Read Context

Node/Attribute:

Keep the below generated code and delete the rest which is not required.

```
DATA lo_nd_manager TYPE REF TO if_wd_context_node.
DATA lo_el_manager TYPE REF TO if_wd_context_element.
DATA ls_manager TYPE wd_this->element_manager.
* navigate from <CONTEXT> to <MANAGER> via lead selection
lo_nd_manager = wd_context->get_child_node( name =
wd_this->wdctx_manager ).
```

Step b:

Then select the data by the below logic.

```
types : begin of ty_emp,
        pernr type pa0002-pernr,
        end of ty_emp.

data : t_emp type standard table of ty_emp,
        wa_emp type ty_emp,
        t_tab type standard table of SWHACTOR,
        wa_tab type swhactor,
        lt_tab type standard table of SWHACTOR,
        lwa_tab type swhactor.

* Find reporting employees position
CALL FUNCTION 'RH_STRUC_GET'
  EXPORTING
    act_otype = 'S'
    act_objid = '50072696'
    act_wegid = 'B002' "Evaluation path
  TABLES
    RESULT_TAB = t_tab.
```

```

loop at t_tab into wa_tab.
refresh lt_tab.
* Find the holder of the position(repotees)
CALL FUNCTION 'RH_STRUC_GET'
  EXPORTING
    act_otype      = 'S'
    act_objid      = wa_tab-objid
    act_wegid      = 'A008'"Evaluation Path
  TABLES
    RESULT_TAB    = lt_tab.
clear lwa_tab.
read table lt_tab into lwa_tab index 1.
if sy-subrc eq 0.
  wa_emp-pernr = lwa_tab-objid.
  append wa_emp to t_emp.
endif.
endloop.

```

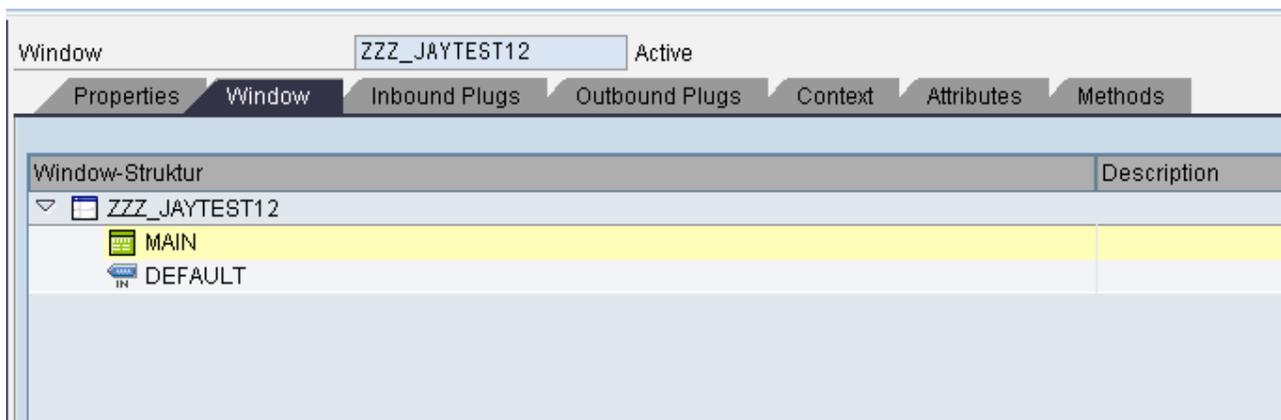
Step c:

Then bind the table.

```
lo_nd_manager->bind_table( t_emp ).
```

Embedding View

Right click and embed the view Main.



Creating Web Dynpro Application

Create Web Dynpro Application by right clicking the Webdynpro (ZZZ_JAYTEST12).

Right click the Web Dynpro component and activate.

Code

Supply Function method Get_details.

```

method GET_DETAILS .
* if necessary, get static attributes of parent element
DATA ls_parent_attributes TYPE wd_this->element_manager.
parent_element->get_static_attributes(
IMPORTING
static_attributes = ls_parent_attributes ).
DATA ls_workschedule TYPE wd_this->Element_workschedule.
* From Pa0002
select single vorna nachn from pa0002
into (ls_workschedule-vorna, ls_workschedule-nachn)
where pernr eq ls_parent_attributes-pernr
and begda le sy-datum
and endda ge sy-datum.
* From Pa0007
select single schkz mostd wostd from pa0007
into (ls_workschedule-schkz,
ls_workschedule-mostd,
ls_workschedule-wostd)
where pernr eq ls_parent_attributes-pernr
and begda le sy-datum
and endda ge sy-datum.
ls_workschedule-pernr = ls_parent_attributes-pernr.
* bind a single element
node->bind_structure(
new_item = ls_workschedule
set_initial_elements = abap_true ).
endmethod.

method WDDUINIT .
DATA lo_nd_manager TYPE REF TO if_wd_context_node.
DATA lo_el_manager TYPE REF TO if_wd_context_element.
DATA ls_manager TYPE wd_this->element_manager.
* navigate from <CONTEXT> to <MANAGER> via lead selection
lo_nd_manager = wd_context->get_child_node( name =
wd_this->wdctx_manager ).

types : begin of ty_emp,
pernr type pa0002-pernr,
end of ty_emp.
data : t_emp type standard table of ty_emp,
wa_emp type ty_emp,
t_tab type standard table of SWHACTOR,
wa_tab type swhactor,
lt_tab type standard table of SWHACTOR,
lwa_tab type swhactor.

```

```

* Find reporting employees position
CALL FUNCTION 'RH_STRUC_GET'
  EXPORTING
    act_otype      = 'S'
    act_objid      = '50072696'
    act_wegid      = 'B002'"Evaluation path"
  TABLES
    RESULT_TAB    = t_tab.
loop at t_tab into wa_tab.
refresh lt_tab.
* Find the holder of the position(repotees)
CALL FUNCTION 'RH_STRUC_GET'
  EXPORTING
    act_otype      = 'S'
    act_objid      = wa_tab-objid
    act_wegid      = 'A008'"Evaluation Path"
  TABLES
    RESULT_TAB    = lt_tab.
clear lwa_tab.
read table lt_tab into lwa_tab index 1.
if sy-subrc eq 0.
  wa_emp-pernr = lwa_tab-objid.
  append wa_emp to t_emp.
endif.
endloop.
lo_nd_manager->bind_table( t_emp ).
endmethod.

```

Output

Direct reportees

▼ Manager

- 90000030
- 90000028
- 90000031
- 90000027
- 90000046

Employee Details

Employee Number: <input type="text" value="90000030"/>	Work Schedule: <input type="text" value="B002"/>
Last Name: <input type="text" value="Training"/>	First Name: <input type="text" value="Jaytest3"/>
Weekly Hours: <input type="text" value="37,00"/>	Monthly Hours: <input type="text" value="160,21"/>

Depending upon the employee selected, the Employee details will be displayed. Since we build the Recursion node, the number of reportees is getting decided at runtime and the child nodes will be built in runtime.

Related Content

[Web Dynpro ABAP - Supply Function and ALV Tree](#)

For more information, visit the [Web Dynpro ABAP 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.