ADF Code Corner

Oracle JDeveloper OTN Harvest 09 / 2011

twitter.com/adfcodecorner

Abstract:

The Oracle JDeveloper forum is in the Top 5 of the most active forums on the Oracle Technology Network (OTN). The number of questions and answers published on the forum is steadily increasing with the growing interest in and adoption of the Oracle Application Development Framework (ADF).

The ADF Code Corner "Oracle JDeveloper OTN Harvest" series is a monthly summary of selected topics posted on the OTN Oracle JDeveloper forum. It is an effort to turn knowledge exchange into an interesting read for developers who enjoy harvesting little nuggets of wisdom.

http://blogs.oracle.com/jdevotnharvest/

Author: Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
30-SEP-2011

Oracle ADF Code Corner OTN Harvest is a monthly blog series that publishes how-to tips and information around Oracle JDeveloper and Oracle ADF.

Disclaimer: ADF Code Corner OTN Harvest is a blogging effort according to the Oracle blogging policies. It is not an official Oracle publication. All samples and code snippets are provided "as is" with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

If you have questions, please post them to the Oracle OTN JDeveloper forum: http://forums.oracle.com/forums/forum.jspa?forumID=83

September 2011 Issue – Table of Contents

Oracle Learning Library

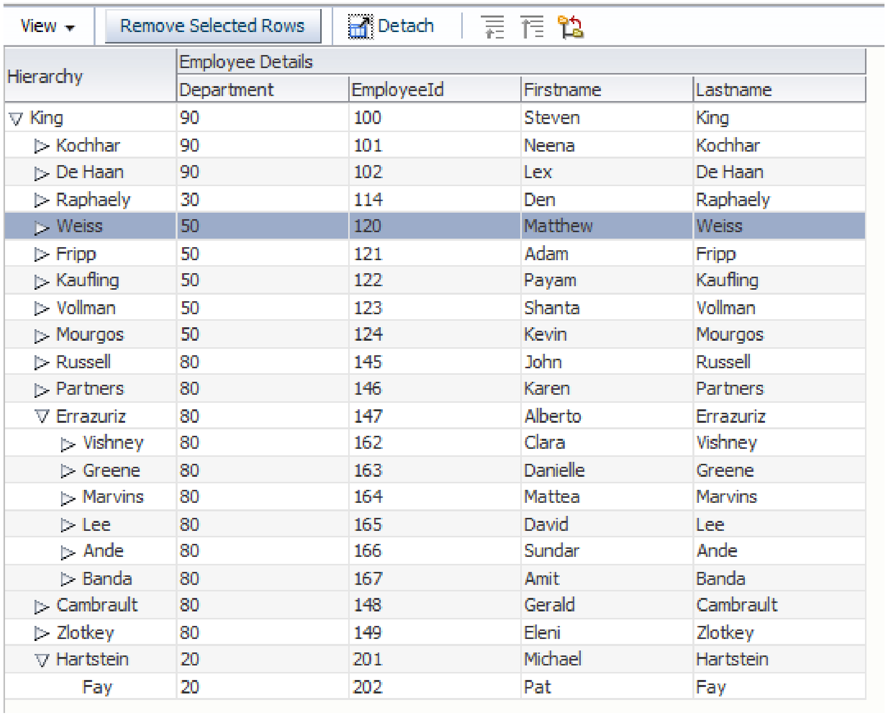
Building enterprise applications often involves different technologies and products that developers need to integrate. Before you can integrate however, there is some learning that has to happen first. The Oracle library is a repository of tutorials for you to search and work through. It is an effort to combine tutorials published by the different product teams into a single place. Check it out!

http://apex.oracle.com/pls/apex/f?p=44785:1:3334232246654331

Synchronize edit form with self-referencing VO in tree table

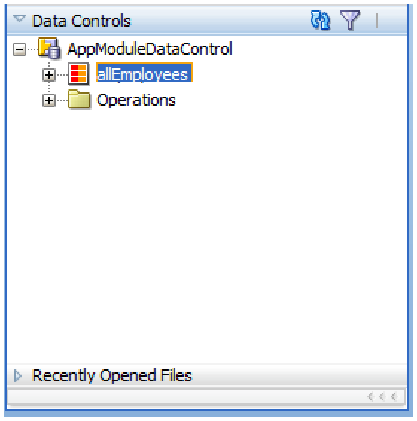
Sample 32 on ADF Code Corner shows how to build a hierarchical tree table representation from a self referencing View Object: http://www.oracle.com/technetwork/developer-tools/adf/learnmore/index-101235.html#CodeCornerSamples

A frequent question is how to build an edit form for the table data and how to synchronize it with the selected row in the tree table. In example 32 published on ADF Code Corner, the tree table component shows employee data of the HR schema structured by the reporting lines.

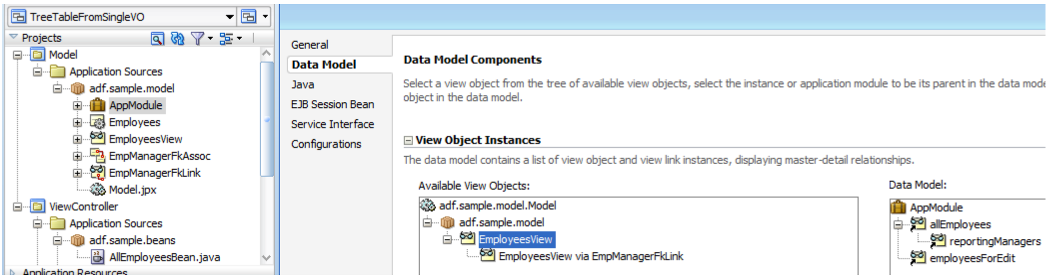


Hierarchy	Employee Details			
	Department	EmployeeId	Firstname	Lastname
▽ King	90	100	Steven	King
▷ Kochhar	90	101	Neena	Kochhar
▷ De Haan	90	102	Lex	De Haan
▷ Raphaely	30	114	Den	Raphaely
▷ Weiss	50	120	Matthew	Weiss
▷ Fripp	50	121	Adam	Fripp
▷ Kaufling	50	122	Payam	Kaufling
▷ Vollman	50	123	Shanta	Vollman
▷ Mourgos	50	124	Kevin	Mourgos
▷ Russell	80	145	John	Russell
▷ Partners	80	146	Karen	Partners
▽ Errazuriz	80	147	Alberto	Errazuriz
▷ Vishney	80	162	Clara	Vishney
▷ Greene	80	163	Danielle	Greene
▷ Marvins	80	164	Mattea	Marvins
▷ Lee	80	165	David	Lee
▷ Ande	80	166	Sundar	Ande
▷ Banda	80	167	Amit	Banda
▷ Cambrault	80	148	Gerald	Cambrault
▷ Zlotkey	80	149	Eleni	Zlotkey
▽ Hartstein	20	201	Michael	Hartstein
Fay	20	202	Pat	Fay

To build and synchronize an edit form with the tree table selection, you need to create an additional View Object instance of the view object used in the tree table. Sample 32 has the tree table built on top of the **allEmployees** view object instance.



To create a second instance, double click the **AppModule** entry in the **Model** project.



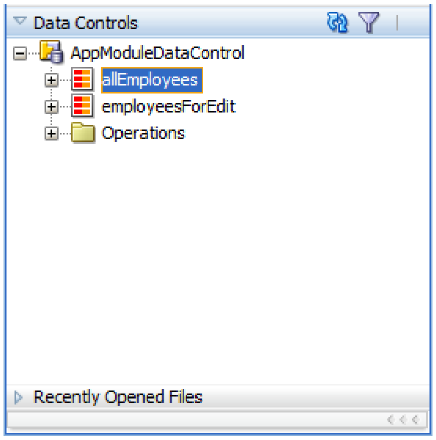
Data Model Components

Select a view object from the tree of available view objects, select the instance or application module to be its parent in the data mode
object in the data model.

View Object Instances

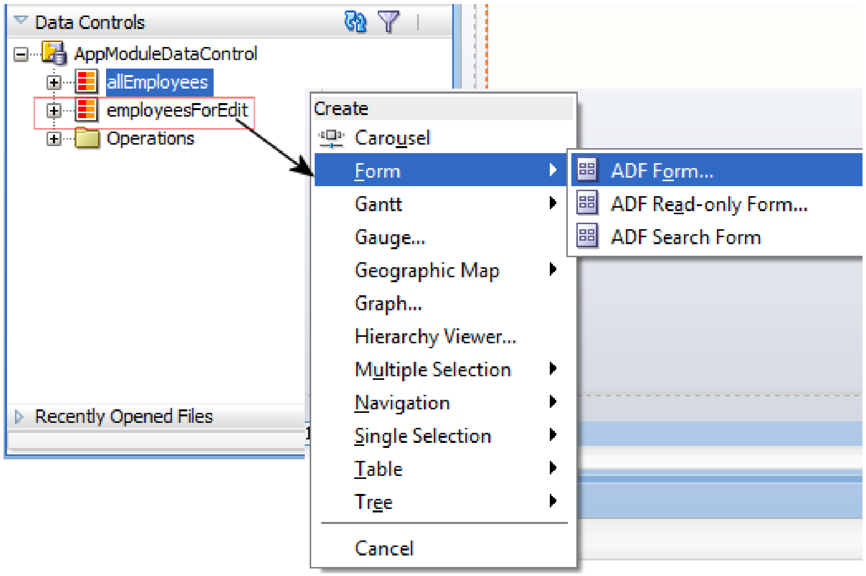
The data model contains a list of view object and view link instances, displaying master-detail relationships.

Select the **EmployeeView** object entry and shuffle it to the **Data Model**. Optionally, right mouse click the new view object instance entry and choose **rename**. In this example, we renamed the instance to **employeesToEdit**, which gets reflected in the DataControl panel as shown below.



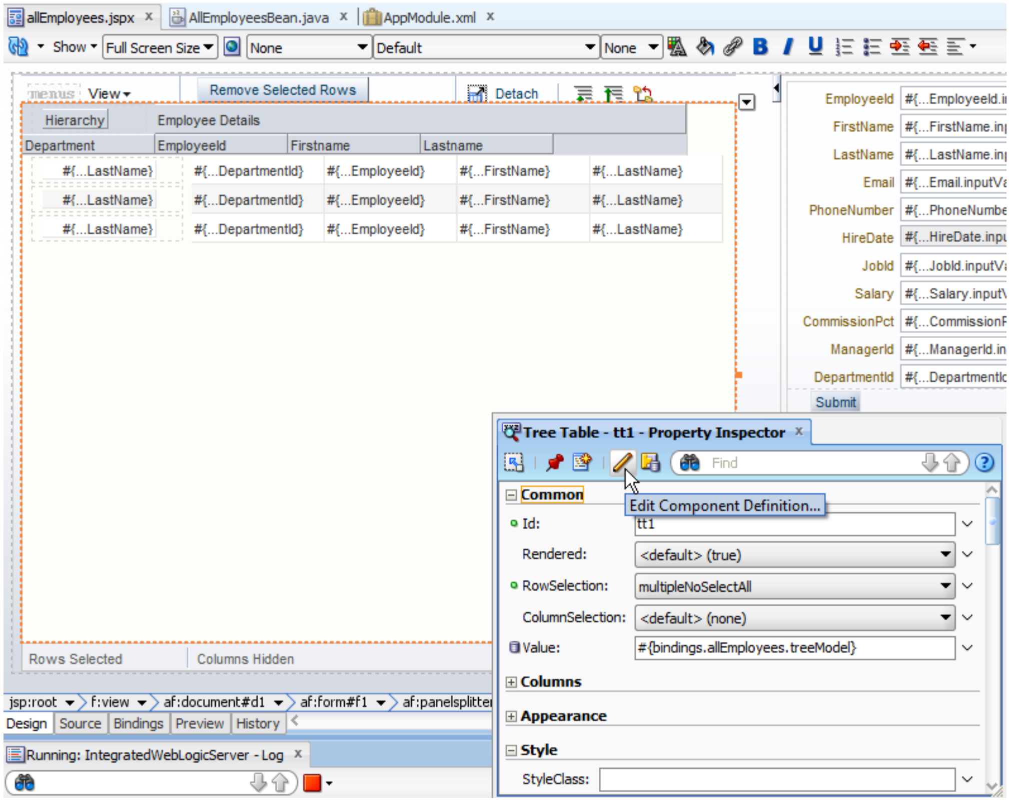
Expand the **ViewController** project and open the **allEmployees.jspx** page.

Drag the **employeesForEdit** collection from the DataControl panel into the page and drop it as an ADF form.



This implicitly creates a new iterator binding, **employeesForEditIterator**.

In the visual editor, select the tree table and open the Property Inspector. In the PropertyInspector, select the pencil icon as shown in the image below:



Id: tt1

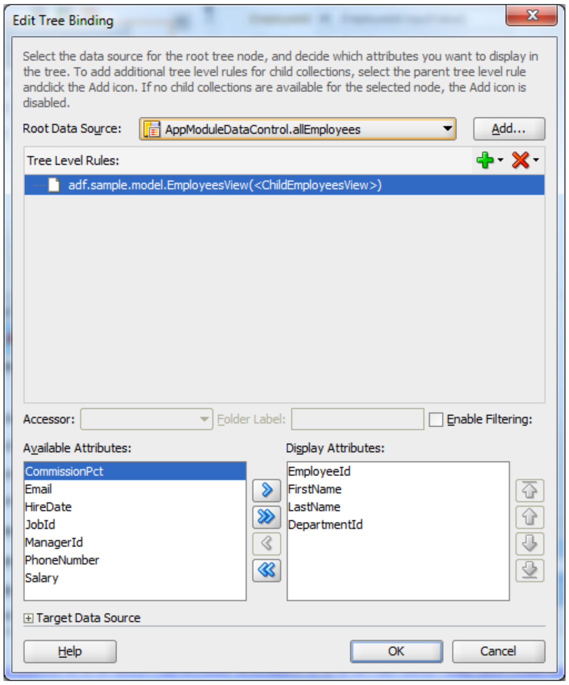
Rendered: <default> (true)

RowSelection: multipleNoSelectAll

ColumnSelection: <default> (none)

Value: #{bindings.allEmployees.treeModel}

The tree binding dialog shows the following entry for the self referencing view object.



Edit Tree Binding

Select the data source for the root tree node, and decide which attributes you want to display in the tree. To add additional tree level rules for child collections, select the parent tree level rule andclick the Add icon. If no child collections are available for the selected node, the Add icon is disabled.

Root Data Source: AppModuleDataControl.allEmployees

Tree Level Rules: adf.sample.model.EmployeesView(<ChildEmployeesView>)

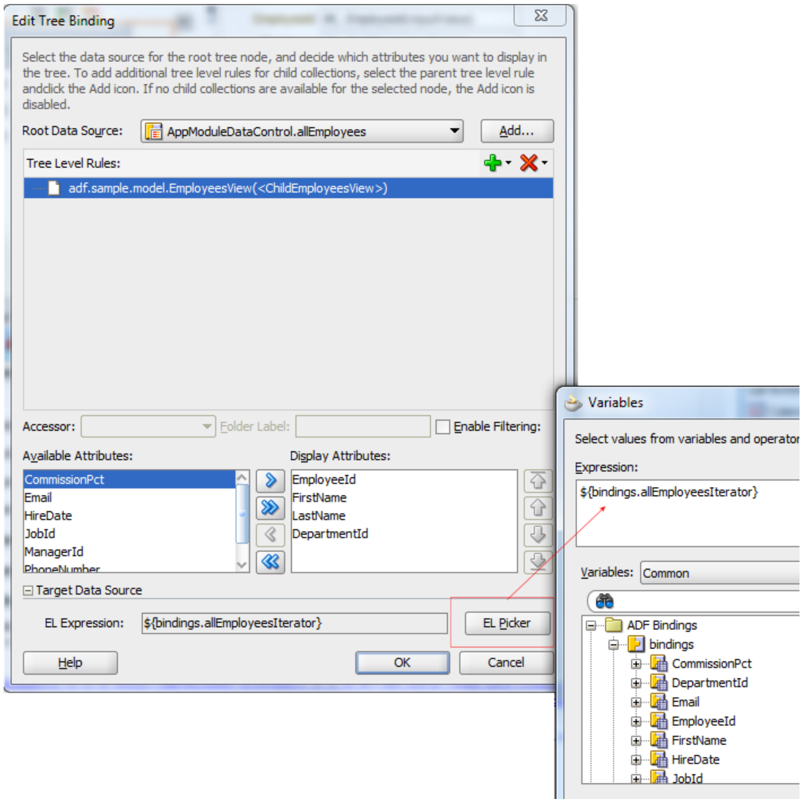
Accessor: Folder Label: □ Enable Filtering:

Available Attributes: CommissionPct, Email, HireDate, JobId, ManagerId, PhoneNumber, Salary

Display Attributes: EmployeeId, FirstName, LastName, DepartmentId

⊞ Target Data Source

Expand the **Target Data Source** node and configure it to have its value pointing to the **employeesForEditIterator** (which is the iterator the edit form is bound to).



Edit Tree Binding

Select the data source for the root tree node, and decide which attributes you want to display in the tree. To add additional tree level rules for child collections, select the parent tree level rule andclick the Add icon. If no child collections are available for the selected node, the Add icon is disabled.

Root Data Source: AppModuleDataControl.allEmployees

Tree Level Rules: adf.sample.model.EmployeesView(<ChildEmployeesView>)

Accessor: Folder Label: □ Enable Filtering:

Available Attributes: CommissionPct, Email, HireDate, JobId, ManagerId, PhoneNumber

Display Attributes: EmployeeId, FirstName, LastName, DepartmentId

⊟ Target Data Source

EL Expression: ${bindings.allEmployeesIterator}

Variables

Select values from variables and operator

Expression: ${bindings.allEmployeesIterator}

Variables: Common

Configure the **PartialTrigger** property of the edit form **panelFormLayout** component to point to the **treeTable** component id so that the form is partially refreshed when the selection in the tree table has changed. To show row data updates in the tree table, configure the **PartialTrigger** property of the `af:treeTable` component and have it pointing to the edit form submit button.

In summary: The secret to synchronizing an edit form with the selection in a tree or tree table built from a recursive view object query is in that you need to create a second instance of the view object in the data model. You cannot just use the iterator that the tree table is bound to.

Launching popup dialogs maximized

Pressing the ADF Faces panel collection – `af:panelCollection` – detach button shows the tree or table content it contains in a popup dialog that shows maximized. The same behavior is often asked for by customers for their own `af:dialog` popups. This – of course – then is expected also to work with different client-side screen resolutions.

Martin Deh from Oracle published a JavaScript based solution for this on the Oracle Architecrure team (A-team) blog (http://blogs.oracle.com/ATEAM_WEBCENTER/). The JS function below is copied from Martin's blog for documentation purpose.

```
function openPopup(popupId, panelWindowId){
  return function(event){
   var agent = AdfAgent.AGENT;
   var windowWidth = agent.getWindowWidth();
   var windowHeight = agent.getWindowHeight();
   var popup = AdfPage.PAGE.findComponentByAbsoluteId(popupId);
   var panelWindow = popup.findComponent(panelWindowId);
   panelWindow.setContentWidth(Math.max(100, windowWidth-60));
   panelWindow.setContentHeight(Math.max(100, windowHeight-80));
   if(popup != null)
     popup.show();
   }
}
```

Note how this function uses a JavaScript callback so it can pass additional arguments – the popup Id and the panelWindow id - when invoked from an `af:clientListener` added e.g. to a button.

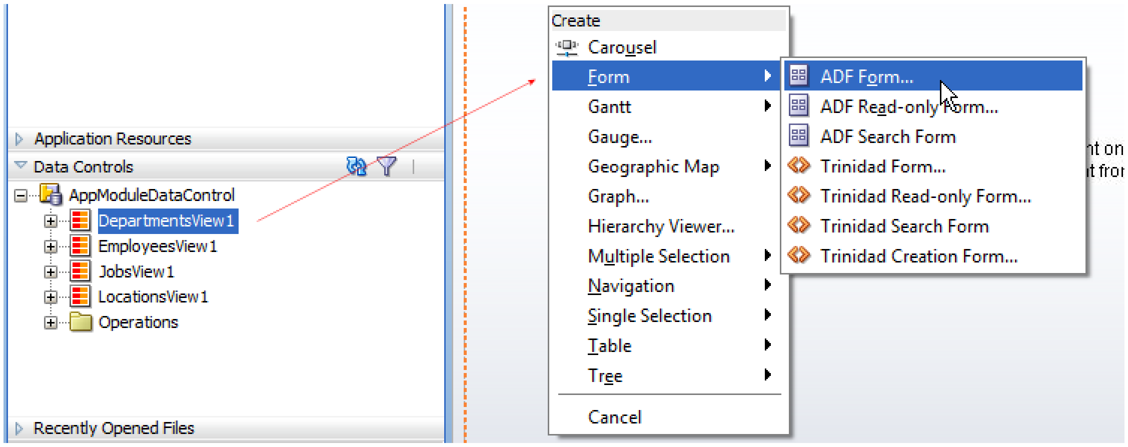
```
af:clientListener type="action"
                   method="openPopup('popupId','panelWindowId')"
```

For a detailed explanation and screenshots, read Martin's full blog entry at:

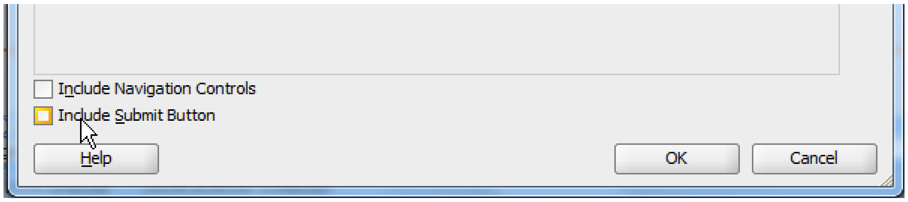
http://blogs.oracle.com/ATEAM_WEBCENTER/entry/dynamic_resizing_for_popup_dialogs

Creating navigation buttons for an existing form

When you drag a collection from the DataControls panel and drop it as an ADF form on a page …



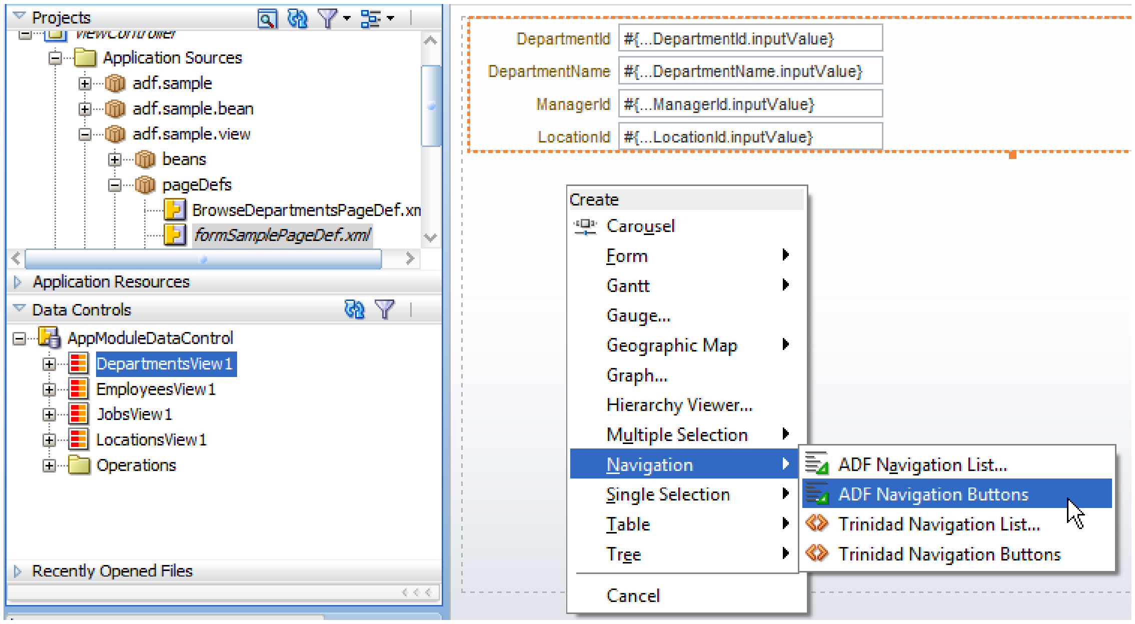
… you check the **Include Submit Button** option to create navigation buttons for the form.



□ Include Navigation Controls
□ Include Submit Button

However, if you forget to select this option, how can you create navigation buttons without deleting and re-creating the form?

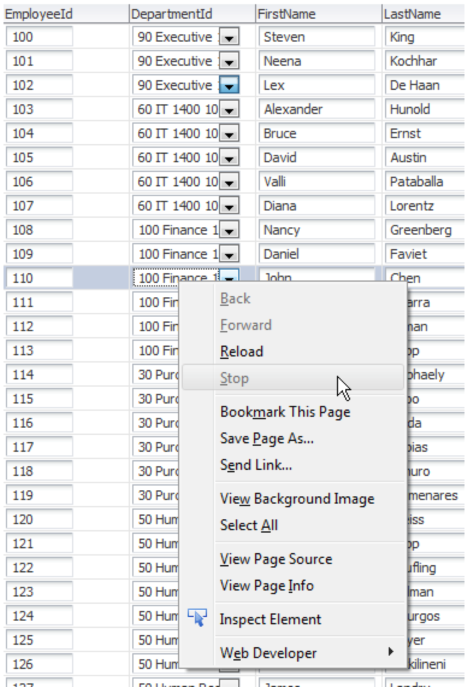
You can create navigation buttons by again dragging the collection from the DataControls panel. This time however, you choose **Navigation | ADF Navigation Buttons** from the context menu.



DepartmentId	#{...DepartmentId.inputValue}
DepartmentName	#{...DepartmentName.inputValue}
ManagerId	#{...ManagerId.inputValue}
LocationId	#{...LocationId.inputValue}

How to disable browser context menus

Browsers have context menus, which may not be ideal if the menu options make no sense in the context of an application. For example, the image below shows the context menu FF opens when the right mouse button is pressed on an ADF table.



EmployeeId	DepartmentId	FirstName	LastName
100	90 Executive	Steven	King
101	90 Executive	Neena	Kochhar
102	90 Executive	Lex	De Haan
103	60 IT 1400 10	Alexander	Hunold
104	60 IT 1400 10	Bruce	Ernst
105	60 IT 1400 10	David	Austin
106	60 IT 1400 10	Valli	Pataballa
107	60 IT 1400 10	Diana	Lorentz
108	100 Finance 1	Nancy	Greenberg
109	100 Finance 1	Daniel	Faviet
110	100 Finance 1	John	Chen
111	100 Fir		arra
112	100 Fir		nan
113	100 Fir		op
114	30 Purc		phaely
115	30 Purc		bo
116	30 Purc		da
117	30 Purc		bias
118	30 Purc		uro
119	30 Purc		menares
120	50 Hum		iss
121	50 Hum		op
122	50 Hum		ıfling
123	50 Hum		man
124	50 Hum		urgos
125	50 Hum		yer
126	50 Hum		kilineni

On the OTN forum, Timo Hahn (see his blog: http://tompeez.wordpress.com/) provided a solution of how to suppress this browser behavior using JavaScript.

First, you drag an `af:clientListener` component from the ADF Faces component palette onto the table

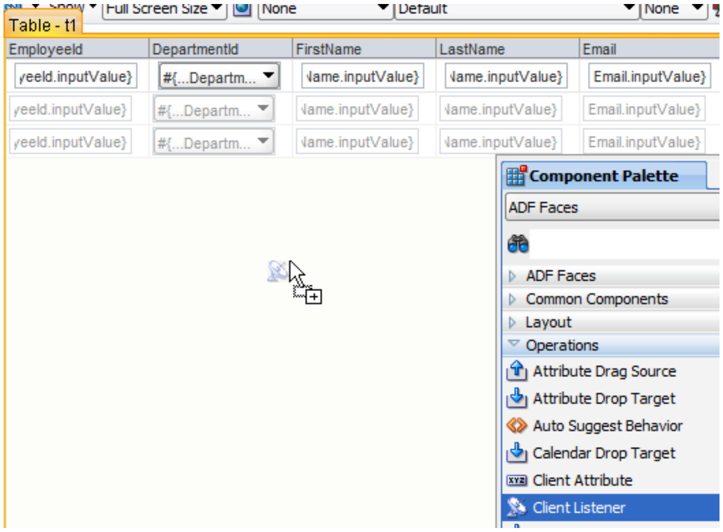
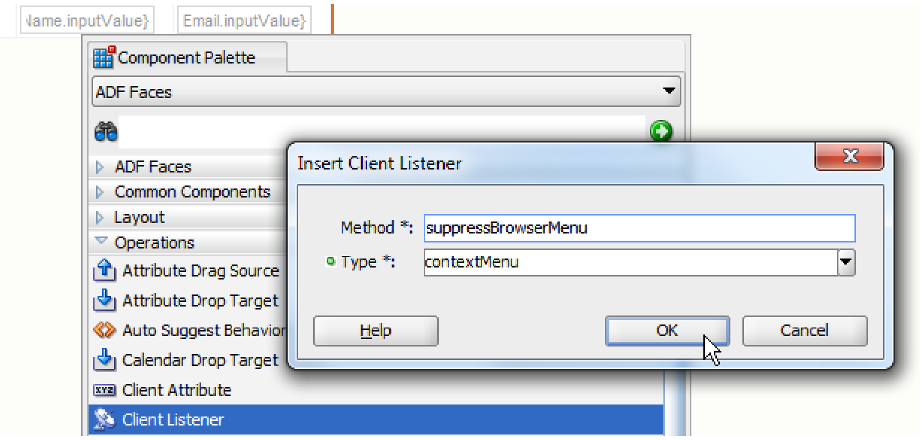


Table - t1

EmployeeId	DepartmentId	FirstName	LastName	Email
yeeId.inputValue}	#{...Departm...	Name.inputValue}	Name.inputValue}	Email.inputValue}
yeeId.inputValue}	#{...Departm...	Name.inputValue}	Name.inputValue}	Email.inputValue}
yeeId.inputValue}	#{...Departm...	Name.inputValue}	Name.inputValue}	Email.inputValue}

In the opened client listener configuration dialog, provide the name of a JavaScript function to call and choose **contextMenu**as the type.



Method *: suppressBrowserMenu

Type *: contextMenu

In ADF Faces, you define the JavaScript function **suppressBrowserMenu** (as used in this example) using the `af:resource` tag. The JavaScript code snippet can be read from a JS library (an external file) or directly added to the page (shown below).

```
<af:resource type="javascript">
   function suppressBrowserMenu(ctxEvent){
     ctxEvent.cancel();
   }
</af:resource>
```

Note: Think big! Enterprise applications should be designed for reuse and as such I recommend adding JavaScript snippets like this in a custom JS library that you reference e.g. from a page template. JavaScript that is stored and loaded from external files is cached in browsers and thus doesn't cost extra.

How-to launch print dialog when showing printable page

The code shown below, when referenced from the **beforePhase** property of the **f:view** component, automatically opens the browser print dialog if an ADF Faces page renders as printable in response to command item invoking the **af:showPrintablePageBehavior** tag.

```
public void beforePhaseMethod(PhaseEvent phaseEvent) {
  if (phaseEvent.getPhaseId() == PhaseId.RENDER_RESPONSE){
    FacesContext fctx = FacesContext.getCurrentInstance();
    AdfFacesContext adfFacesContext =
                       AdfFacesContext.getCurrentInstance();
    if(adfFacesContext.getOutputMode()== OutputMode.PRINTABLE){
       ExtendedRenderKitService erks = null;
       erks = Service.getRenderKitService(
                         fctx,
                         ExtendedRenderKitService.class);
       erks.addScript(fctx, "window.print();");
```

```
            }
        }
}
```

The managed bean method is referenced from the f:view component as follows

```
<f:view beforePhase="#{SampleBean.beforePhaseMethod}">
  ...
</f:view>
```

With this listener and code, when the af:showPrintableBehavior tag is used on a command item to show a printable page, the browser print dialog is automatically opened.

While the same code also works for page fragments, the f:view tag is only available for JSPX documents. In this case you either set the **beforePhase** property on the JSPX document hosting the page fragment, or define a global phase listener (faces-config.xml) that then works for all pages in an application.

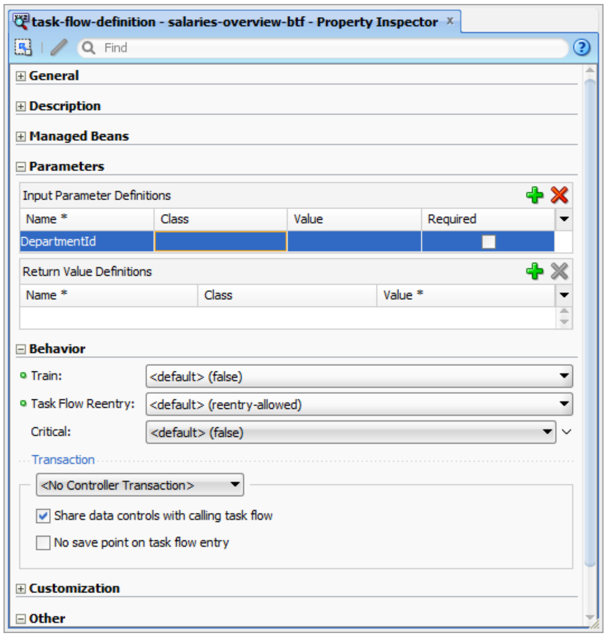
Note: I did write about this solution in the January 2011 edition of OTN Harvest. To this time however I used an internal flag as I did not know better. The code used in this post only uses public APIs and thus is a solution that lasts.

SQL error during statement preparation

Interesting finding regarding the *javax.faces.el.EvaluationException: oracle.jbo.SQLStmtException: JBO-27122: SQL error during statement preparation* problem has been posted to the OTN forum by Tammy Osborn. Tammy found that this had to do with statement caching size in WebLogic Server, which by default is set to 10. Setting this value to 0 solved the problem.

Defining default values for task flow input parameters

Bounded task flows cannot have default value defined in metadata for their input parameters.



task-flow-definition - salaries-overview-btf - Property Inspector

Find

⊞ General

⊞ Description

⊞ Managed Beans

⊟ Parameters

Input Parameter Definitions

Name *	Class	Value	Required
DepartmentId			☐

Return Value Definitions

Name *	Class	Value *

⊟ Behavior

Train: <default> (false)

Task Flow Reentry: <default> (reentry-allowed)

Critical: <default> (false)

Transaction

<No Controller Transaction>

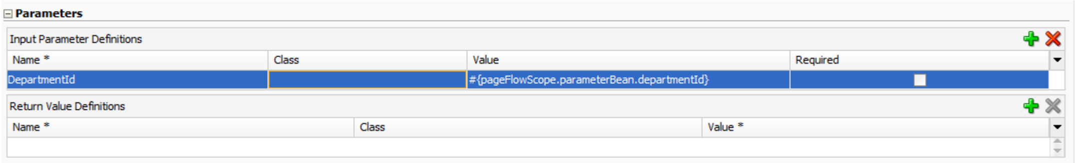
☑ Share data controls with calling task flow

☐ No save point on task flow entry

⊞ Customization

⊟ Other

However, you can bind the **Value** property to a managed bean property, with the managed bean being in **pageFlowScope**. The default value of the bean property variable would then automatically become the default value of the task flow input parameter.



⊟ **Parameters**

Input Parameter Definitions

Name *	Class	Value	Required
DepartmentId		#{pageFlowScope.parameterBean.departmentId}	□

Return Value Definitions

Name *	Class	Value *

Note that if the input parameter is configured to be required, the parameter needs to be part of the task flow call.

ADF Code Corner

OTN Harvest Spotlight

- Andrejus Baranovskis

Andrejus Baranovskis is CEO and founder of Red Samurai Consulting and one of the "extreme" early adopters of Oracle JDeveloper 11g and Oracle ADF. Andrejus Baranovskis is a known as an OracleADF expert and travels the world visiting his clients.

" Long live ADF" – Andrejus Baranovskis

ACC: **Tell us about Red Samurai Consulting and your role in this company?**

Andrejus: Red Samurai Consulting focuses on high quality, expert level services around Oracle Fusion Middleware. We are a small team of experts specialized in Oracle Fusion technology and our goal is to maintain high quality in everything we do.

I'm the founder of Red Samurai Consulting. The idea to become specialized in consulting I got while working for IBM Global Business Services. I enjoy running my own company and helping my customers to successfully implement their projects.

Typically I go onsite, design the technical architecture for a project, implement frameworks, deliver advanced trainings, resolve technical issues and performance tune existing systems.

I enjoy sharing my ADF, WebCenter and UCM knowledge with the community through my blog (http://andrejusb.blogpost.com).

Beside of this, I frequently participate in the ADF Enterprise Methodology Group (EMG) - http://groups.google.com/group/adf-methodology - and speak at Oracle conferences.

The latest focus of Red Samurai Consulting has been in implementing its own product – a lightweight Social Enterprise Suite (SES), on top of Oracle WebCenter Suite, to better cater to customer business needs.

ACC: **Why did you name your company - Red Samurai Consulting?**

Andrejus: Red is the color of Oracle. We are exclusively specialized in Oracle technologies and thus red became our color too. A Samurai is always dedicated in delivering top quality results and never disappoint the customer. The combination of these two qualities allows us to ensure excellent service to our customers.

ACC: **What is your IT background?**

Andrejus: I have Masters in Computer Science and have been working in IT industry from the early days of my college.

Initially I did work with Oracle JDeveloper 9*i* and implemented Java EE systems based on Oracle GIS products. It was during this time that the ADF framework came up and I saw a clear strategy at Oracle towards Oracle ADF and invested lot of my private time in learning Oracle ADF framework.

After working with Java EE technology I truly enjoyed learning Oracle ADF. Oracle ADF is doing all of the ground work for me so I can focus on what is really important and complex tasks.

ACC: **You started with customer projects using Oracle ADF and JDeveloper 11g when it was in Technology Preview and not in production. What made you so confident that it was the right technology for you and your customers?**

Andrejus: Oracle always had a clear strategy for Oracle ADF. When Oracle JDeveloper 11g had been in Technology Preview, it was clear that Oracle JDeveloper 11g and Oracle ADF would become the key development platform for Oracle Fusion Applications.

While attending Oracle conferences, I met with Oracle Product Managers and everyone was very enthusiastic about Oracle ADF. This enthusiasm motivated me to be confident in ADF from day one.

I was also lucky to work with a customer, who shared my view and who was willing to adopt Oracle ADF as a framework to migrate away from Oracle Forms. It was a risk, but we were following outlined Oracle strategy and learning new technology. In retrospect, it was the right decision.

ACC: **How do you currently use Oracle JDeveloper and ADF?**

Andrejus: With the adoption of the Oracle Fusion Middleware (which in large parts is based on the Oracle ADF framework), customers use products built with Oracle ADF. This means we are not only working with Oracle JDeveloper and ADF, but also use JDeveloper and ADF skills to integrate different Oracle Fusion products together.

For example, we are using Oracle JDeveloper and ADF to extend Oracle WebCenter Task Flows, to develop our own custom ADF Task Flows and integrate into WebCenter portal. We are working with SOA and BPM projects, where most of the business logic and interfaces are implemented with ADF.

ACC: **How do you see the market for Oracle ADF related jobs?**

Andrejus: With Oracle announcing Oracle Fusion Middleware, each year, the Oracle ADF job market grows significantly.

When Oracle ADF 11g went production, only former Oracle Forms customers did hire Oracle ADF developers. Now the situation is changed and I see more acceptances for Oracle ADF among Java EE customers. Today there are more ADF jobs available in the market than skilled ADF developers.

ACC: **If you had to hire new developers for Red Samurai Consulting, how would you go finding them and what skills would you look for?**

Andrejus: We have a very good contact with the local University. Probably the first option would be to go and talk with the professors to have them recommending students.

However, at Red Samurai, we prefer to grow our own talents. We would look for people with Java EE mindset and strong motivation to deliver good results. If we need to find someone with ADF experience - we would search inside the ADF Community.

Along with technical skills, it is also important to be a good team player, so we will be looking at those skills as well.

ACC: **So far, what has been your biggest challenge in building Java EE application with Oracle ADF?**

Andrejus: When working in large distributed teams, it is always a challenge to control project quality and ensure every developer is following same standard for naming conventions, development patterns, etc.

In specific situations, it becomes hard to manage dependencies between different auto generated objects. However, we expect this to improve with latest release of JDeveloper 11g R2, where we can track dependencies between ADF BC objects through dependency diagram.

ACC: **Which feature of ADF was the greatest benefit to your projects?**

Andrejus: I think it was the MDS Seeded Customizations, because it allowed us to implement our own product (lightweight Social Enterprise Suite) on top of Oracle ADF/WebCenter framework.

Our product can be deployed at different customer sites, with each customer having their own customizations applied, based on specific requirements. With MDS, we are able to maintain our own code source without impacting any changes applied for the customer.

ADF task flows lifts Oracle ADF to higher levels, allowing better support for user requirements implementation and higher reusability.

ACC: **Away from the on line help, what have been your most valuable sources of ADF knowledge?**

Andrejus: Oracle conferences are really valuable to me - Oracle OpenWorld, ODTUG, local Oracle User Group events. Conferences are best way to get latest up to date information and chat with Oracle Product Management about new features and product strategy.

ACC: **Are you in any way actively involved in the ADF Community?**

Andrejus: Starting from 2006, I have been posting one or two sample applications on my blog weekly - http://andrejusb.blogspot.com

The sample applications are designed to describe solution for certain question or problem and provide detailed step-by-step description instructions.

I enjoy sharing my Oracle technical knowledge through the blog, because it's a two way channel as it allows me to learn more and be in touch with blog readers through blog reader comments and emails.

I'm also a frequent speaker on various Oracle conferences (Oracle Open World, ODTUG, UKOUG, SOA Community, ADF EMG).

One of the first members of ADF Enterprise Methodology Group (ADF EMG) and contributor to ADF EMG sessions content.

In partnership with Oracle University, I'm delivering Oracle celebrity seminar series for Oracle ADF, WebCenter and BPM integration.

ACC: **Many people live by calendar mottos. What would be your quote of the day for Oracle ADF?**

Andrejus: My quote of the day - "Keep It Simple !" This quote is for all Oracle ADF developers because the key knowledge in Oracle ADF is to understand and use the framework and not to create overly complicated solutions. Instead you should use features offered by the ADF framework.

ACC: **Are you attending OOW 2011 in San Francisco? If so, what are your planned activities there and how could people meet you just to say 'hello'?**

Andrejus: I will be attending OOW 2011 in San Francisco, and am looking forward to catching up with friends and colleagues.

I am always excited to talk to new people and make friends. I am interested in talking to developers beginning with Oracle ADF to answer questions with my recommendations on how to avoid common pitfalls.

I will be speaking on ADF EMG Sunday, as well as other sessions - I enjoy talking to people and hear about their adventures, so stop by and say 'hello'.

ACC: **ADF Genie grants you a wish, what would you ask for?**

Andrejus: My wish - "Long live ADF !"

ACC: **Thank you, Andrejus.**

RELATED DOCOMENTATION

☒	
☒	
☒	