AIM

MD.070 Application Extension Technical Design

Oracle Application Framework Extension

Delete

Author:	Arone.zhang
Creation Date:	April 24, 1999
Last Updated:	March 5, 2009
Document Ref:	MD070_OAExtension_TB007
Version:	1.0

Approvals:

<Approver 1>

Arone.zhang

<Approver 2>



Copy No.

Document Control

Change Record

Date	Author	Version	Change Reference
24-Apr-99	Arone.zhang	1.0	No Previous Document

Reviewers

Name	Position
Arone.zhang	Project Leader

Distribution

Copy No.	Name	Location
1	Library Master	Project Library
2		Project Manager
3		
4		

Note To Holders:

If you receive an <u>electronic copy</u> of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a <u>hard copy</u> of this document, please write your name on the front cover, for document control purposes.

Contents

Document Control	ii
课程介绍	1
概述	1
课程的目标	2
步骤1:在 EmployeeEOImpl 类中实现 Delete	2
步骤2: 实现一个 Delete Switcher	3
步骤3:实现 Delete 动作	6
步骤4(可选): 创建和实现 Status 列	12
Open and Closed Issues for this Deliverable	17
Open Issues	17
Closed Issues	17

课程介绍

概述

本课程讲述了如何在 Employees 查询页中添加一个 Delete 图形列,并实现一个包括警告页面的删除动作。下图显示了根据员工状态显示 Delete 列的页面。

前提:已经完成了 Create 的课程,如果还没有完成相关的学习,请参照之前的相关工作

Em	Employees Arone						
Per	<u>sonalize R</u>	legion					
	Search						
	Personaliz	e Region					
ĺ	Please not	te that the searc	h is case insensitive.				
		malauaa Nama	D		0		
		пріоуее мате		4)	1		
	Emp	ployee Number					
		-	Go Clear				
	Create E	mployee					
	Status	Number \triangle	Name	Manager	Position	Update	Delete
	×	1	<u>Barnes, Penelope</u>		President		Î
	×	2	<u>Brown, James</u>	<u>Barnes, Penelope</u>	Vice President		Ĩ
	×	10	<u>Barnes, Penelope10</u>		President		Î
	×	12	<u>Brown, James 12</u>	<u>Barnes, Penelope</u>	Vice President		Î

课程的目标

完成此课程之后,需要掌握如下的课题:

- 配置一个动作图标来执行表单的提交
- 显示和处理警告对话框消息
- 使用 JSP "forward immediate"显示页内嵌的确认消息
- 删除视图对象中的一行
- 处理 Delete 项目动作
- 使用迭代在视图对象行中查找选择的值
- 实现一个表内容的 switcher
- 控制列的排列对齐
- 使用绑定值实现基于行的图形显示

步骤1:在 EmployeeEOImpl 类中实现 Delete

确认 EmployeeEOImpl 类中存在 remove() 方法,如果没有请添加。

/*

} // end remove()

步骤2: 实现一个 Delete Switcher

Switcher 使用在表中来有条件的显示列中的内容。在本步骤中,配置一个基于员工状态来显示合适的 Delete 图像。

任务2.1: 添加一个属性到 EmployeeSummaryVO 视图对象

为了添加一个 Delete 列到结果表中,需要修改 EmployeeSummaryVO 来包含一个 decode 属性列来实现如下的规则:

- 如果员工有效, Delete 垃圾桶禁用
- 如果员工无效, Delete 垃圾桶启用
- 在导航栏中选择 EmployeeSummaryVO,右键从上下文菜单中选择 Edit EmployeeSummaryVO... 编辑 VO
- 在 Query Statement 文本框中,修改查询语句如下,并通过语法验证。

SELECT EmployeeEO.EMPLOYEE_ID, EmployeeEO.FULL_NAME AS EMPLOYEE_NAME, EmployeeEO.EMAIL_ADDRESS AS EMPLOYEE_EMAIL, EmployeeEO1.EMPLOYEE_ID AS MANAGER_ID, EmployeeEO1.FULL_NAME AS MANAGER_NAME, EmployeeEO1.EMAIL_ADDRESS AS MANAGER_EMAIL, flkp.meaning AS POSITION_DISPLAY, decode(nvl(to_char(EmployeeEO.END_DATE), 'N'), 'N', 'DeleteDisabled', 'DeleteEnabled') AS DELETE_SWITCHER FROM FWK_TBX_EMPLOYEES EmployeeEO, FWK_TBX_EMPLOYEES EmployeeEO1, FWK_TBX_LOOKUP_CODES_VL flkp WHERE EmployeeEO.MANAGER_ID = EmployeeEO1.EMPLOYEE_ID (+) AND flkp.lookup_code = EmployeeEO.POSITION_CODE AND flkp.lookup_type = 'FWK_TBX_POSITIONS'

任务2.2: 创建 Switcher 区域

创建 switcher bean 来和 DeleteSwitcher 视图属性绑定来决定 Delete 图标的显示。

- 在结构窗口中选择 ResultsTable, 右键从上下文菜单中选择 New > switcher
- 选择新建的区域,并设置如下的属性值:

属性	值
ID	DeleteSwitcher
Region Style	Switcher
Attribute Set	/oracle/apps/fnd/attributesets/Buttons/Delete
View Instance	EmployeeSummaryVO1
View Attribute	DeleteSwitcher

任务2.3: 配置默认 Switcher Case

Switcher 模仿了程序中的开关:为每种可能显示的情况创建一个独立的"case"。Switcher bean 绑定到了视图对象属性,它在页面显示的时候返回 case 的名称。每个 case 的 ID 值必须和 deocde 的返回值相匹配。

配置第一个 case 项目来显示禁用 Delete 图像按钮

- 在结构窗口中选择系统自动创建的 case ,然后从上下文菜单中选择 New > Item
- 选择新建的项目,设置如下的属性值:

属性	值
ID	DeleteDisabled
	(必须与步骤1.1中 decode 的返回值匹配)
Region Style	Image
Image URI	deleteicon_disabled.gif
Additional Text	Delete employee action is disabled.
Height	24
Width	24

除了商标图片外,添加到页面中的图片都必须设置 Height 和 Width 属性值。输入的具体值请查看 Oracle BLAF UI 准则图标库或者 辅助图像库

任务2.4: 创建一个新的 Switcher Case

配置第二个 case 项目来显示启用 Delete 图像按钮

当用户选择 Delete 图标时,提交一个表单以至可以在 processFormRequest() 中处理相关事件。然而,当你为图片设置 Destination URI 属性时,用户的选择是一个 HTTP GET 而非 POST,而我们需要执行一个 POST 动作,为了达到此目的,需要明确配置图片执行一个表单提交动作。

● 在结构窗口中选择 **DeleteSwither**,然后从上下文菜单中选择 **New > Case**

- 选择新的 <case> 节点, 然后从上下文菜单中选择 New > Item
- 选择新建的项目,设置如下的属性值:

属性	值
ID	DeleteEnabled
	(必须与步骤1.1中 decode 的返回值匹配)
Region Style	Image
Image URI	deleteicon_enabled.gif
Additional Text	Delete employee action is enabled.
Height	24
Width	24
Action Type	fireAction
	(为此图片启用表单提交事件)
Event	delete
Submit	True

下面需要为当前行的员工号码和名称定义请求参数,后面将使用这些信息来执行删除操作和显示一个对话框

- 选择 Parameters 属性中的"…" 按钮
- 在 Parameters 窗口中,定义一个参数名为 empNum,参数值为 \${oa.EmployeeSummaryVO1.EmployeeId},这个 SPEL(Simplest Possible Expression Language)语法代表了参数 empNum 的值从 EmployeeSummaryVO1 视图实例中的 EmployeeId 属性取得
- 选择 Add Parameters 按钮添加第二个参数,参数名为 empName,参数值为 \${oa.EmployeeSummaryVO1.EmployeeName},这个 SPEL语法代表了参数 empName 的值从 EmployeeSummaryVO1 视图 实例中的 EmployeeName 属性取得
- 确定创建请求参数

在运行期间,OA Framework自动创建和取得 formParameter(隐藏)中的 empName 和 empNum 参数及参数值。这些值再被添加 到请求中。

...

任务2.5: 格式化 Delete 列

所有的图象列都必须中间对齐。图片列自动中间对齐,而 Switcher列则不是。

为了确保正确的显示 Delete 列,添加如下的代码到 EmployeeResultsCO 控制器中的 processRequest() 方法

import oracle.cabo.ui.data.DictionaryData; import oracle.cabo.ui.data.DataObjectList; import oracle.apps.fnd.framework.webui.beans.table.OATableBean; // This controller is associated with the table. OATableBean table = (OATableBean)webBean; // We need to format the Switcher image column so the image is centered // (this isn't done automatically for Switchers as it is for // plain image columns). We start by getting the table's // column formats. // NOTE!!! You must call the prepareForRendering() method on the table *before* // formatting columns. Furthermore, this call must be sequenced *after* the // table is queried, and *after* you do any control bar manipulation. table.prepareForRendering(pageContext); DataObjectList columnFormats = table.getColumnFormats(); DictionaryData columnFormat = null; int childIndex = pageContext.findChildIndex(table, "DeleteSwitcher"); columnFormat =(DictionaryData)columnFormats.getItem(childIndex); columnFormat.put(COLUMN_DATA_FORMAT_KEY, ICON_BUTTON_FORMAT); 任务2.6: 测试

步骤3: 实现 Delete 动作

此步骤的任务是添加代码来处理 Delete 动作。当查询出员工时,可以看到可删除的记录显示。当用户按下 Delete 图标时,一个警告信息将显示如下。如果选择 OK 按钮继续删除操作,一个确认消息将显示在 Employee 页的顶部。

🔨 Warning				
Are you sure you want to delete this employee (Barnes, Penelope10)?				
	No Yes			

_							
E	E Confirmation						
	Employee (Barnes, Penelope10) has been deleted.						
En	ployees	s Arone					
Per	<u>sonalize R</u>	egion					
	Search						
	Personaliz	e Region					
	Please not	te that the search	n is case insensitive.				
Employee Name 🛛 B							
Employee Number		oloyee Number		*			
		2	Go Clear				
	(Create E	mployee)					
	Status	Number Δ	Name	Manager	Position	Update	Delete
	~	1	<u>Barnes, Penelope</u>		President		Ŵ
	~	2	<u>Brown, James</u>	<u>Barnes, Penelope</u>	Vice President		Û
	×	12	<u>Brown, James 12</u>	<u>Barnes, Penelope</u>	Vice President		Ŵ

任务3.1: 添加 deleteEmployee() 方法到 EmployeeAM

添加一个方法到 EmployeeAMImpl 类中获得参数 employeeNumber 来查找 EmployeeSummaryVO 视图对象中匹配的行。

提示:下面的代码演示了通过迭代方式查到需要删除的对象。然而,在11.5.10版本中 oracle.apps.fnd.framework.server.OAViewObjectImpl提供了一个新的方便的方法,能够通过提供的值快速找到需要删除的值。而 外查看Javadoc •••

import oracle.jbo.domain.Number; import oracle.jbo.RowSetIterator; * Deletes an employee. public void deleteEmployee(String employeeNumber) // First, we need to find the selected employee in our VO. // When we find it, we call remove() on the row which in turn // calls remove on the associated EmployeeEOImpl object. int empToDelete = Integer.parseInt(employeeNumber); OAViewObject vo = (OAViewObject)getEmployeeSummaryVO1(); EmployeeSummaryVORowImpl row = null; // This tells us the number of rows that have been fetched in the // row set, and will not pull additional rows in like some of the // other "get count" methods. int fetchedRowCount = vo.getFetchedRowCount(); // We use a separate iterator -- even though we could step through the // rows without it -- because we don't want to affect row currency. RowSetIterator deleteIter = vo.createRowSetIterator("deleteIter"); if (fetchedRowCount > 0) deleteIter.setRangeStart(0); deleteIter.setRangeSize(fetchedRowCount); for (int i = 0; i < fetchedRowCount; i++)</pre> row = (EmployeeSummaryVORowImpl)deleteIter.getRowAtRangeIndex(i); // For performance reasons, we generate ViewRowImpls for all

// View Objects. When we need to obtain an attribute value, // we use the named accessors instead of a generic String lookup. // Number primaryKey = (Number)row.getAttribute("EmployeeId"); Number primaryKey = row.getEmployeeId(); if (primaryKey.compareTo(empToDelete) == 0) { // This performs the actual delete. row.remove(); getTransaction().commit(); break; // only one possible selected row in this case } } // Always close the iterator when you're done. deleteIter.closeRowSetIterator(); // end deleteEmployee

任务3.2: 添加 Delete 处理到 EmployeeResultsCO

添加如下的代码到 EmployeeResultsCO.processFormRequest() 方法中,确定 Delete 图标被选中,则显示警告窗口。

提示: if 语句部分已经在上一课中添加来处理 Create Employee 按钮

import com.sun.java.util.collections.HashMap; import oracle.apps.fnd.framework.webui.OADialogPage; import oracle.apps.fnd.framework.webui.OAWebBeanConstants; import oracle.apps.fnd.common.MessageToken; import oracle.apps.fnd.framework.OAException; ... else if ("delete".equals(pageContext.getParameter(EVENT_PARAM))) // The user has clicked a "Delete" icon so we want to display a "Warning" // dialog asking if she really wants to delete the employee. Note that we // configure the dialog so that pressing the "Yes" button submits to // this page so we can handle the action in this processFormRequest() method. String employeeNumber = pageContext.getParameter("empNum"); String employeeName = pageContext.getParameter("empName"); MessageToken[] tokens = { new MessageToken("EMP_NAME", employeeName)}; OAException mainMessage = new OAException("AK", "FWK_TBX_T_EMP_DELETE_WARN", tokens); // Note that even though we're going to make our Yes/No buttons submit a // form, we still need some non-null value in the constructor's Yes/No // URL parameters for the buttons to render, so we just pass empty // Strings for this. OADialogPage dialogPage = new OADialogPage(OAException.WARNING, mainMessage, null, "", ""); // Always use Message Dictionary for any Strings you want to display. String yes = pageContext.getMessage("AK", "FWK_TBX_T_YES", null); String no = pageContext.getMessage("AK", "FWK_TBX_T_NO", null); // We set this value so the code that handles this button press is // descriptive. dialogPage.setOkButtonItemName("DeleteYesButton"); // The following configures the Yes/No buttons to be submit buttons, // and makes sure that we handle the form submit in the originating // page (the "Employee" summary) so we can handle the "Yes" // button selection in this controller. dialogPage.setOkButtonToPost(true); dialogPage.setNoButtonToPost(true); dialogPage.setPostToCallingPage(true); // Now set our Yes/No labels instead of the default OK/Cancel. dialogPage.setOkButtonLabel(ves);

dialogPage.setNoButtonLabel(no);

// We need to keep hold of the employeeNumber, and the OADialogPage gives us a

// convenient means of doing this. Note that the use of the Hashtable is

// really more appropriate for passing multiple parameters, but we've used

// it here for illustration purposes. See the OADialogPage javadoc for an

// alternative when dealing with a single parameter.

java.util.Hashtable formParams = new java.util.Hashtable(1);

formParams.put("empNum", employeeNumber);

formParams.put("empName", employeeName);

dialogPage.setFormParameters(formParams);

pageContext.redirectToDialogPage(dialogPage);

任务3.3: 添加 Delete 确认处理到 EmployeeResultsCO

添加如下的代码到 EmployeeResultsCO.processFormRequest() 方法中,在警告对话框中确定删除后代表用户希望删除员工。

提示: if 语句部分已经在添加来处理 Delete 图标按钮

import java.io.Serializable;

...

else if (pageContext.getParameter("DeleteYesButton") != null)

// User has confirmed that she wants to delete this employee.

// Invoke a method on the AM to set the current row in the VO and

// call remove() on this row.

String employeeNumber = pageContext.getParameter("empNum");
String employeeName = pageContext.getParameter("empName");
Serializable[] parameters = { employeeNumber };
OAApplicationModule am = pageContext.getApplicationModule(webBean);
am.invokeMethod("deleteEmployee", parameters);
// Now, redisplay the page with a confirmation message at the top. Note
// that the deleteEmployee() method in the AM commits, and our code
// won't get this far if any exceptions are thrown.
MessageToken[] tokens = { new MessageToken("EMP_NAME", employeeName) };
OAException message = new OAException("AK",
 "FWK_TBX_T_EMP_DELETE_CONFIRM", tokens, OAException.CONFIRMATION, null);
pageContext.putDialogMessage(message);

任务3.4: 保存测试

步骤4(可选): 创建和实现 Status 列

上面已经学习了一种有条件显示项目的方式;这个步骤讲述如何使用绑定值的方式来实现同样的任务(使用 oracle.apps.fnd.framework.webui.OADataBoundValueViewObject 来引用 EmployeeSummaryVO 视图对象中的 DECODE 属性)

任务4.1: 添加一个属性到 EmployeeSummaryVO 视图对象

- 如果员工有效,复选标记显示在 Status 列
- 如果员工无效,X图片显示在 Status 列
- 在导航栏中选择 EmployeeSummaryVO,右键从上下文菜单中选择 Edit EmployeeSummaryVO... 编辑 VO
- 在 Query Statement 文本框中,修改查询语句如下,并通过语法验证。

SELECT EmployeeEO.EMPLOYEE ID, EmployeeEO.FULL_NAME AS EMPLOYEE_NAME, EmployeeEO.EMAIL ADDRESS AS EMPLOYEE EMAIL, EmployeeEO1.EMPLOYEE_ID AS MANAGER_ID, EmployeeEO1.FULL NAME AS MANAGER NAME, EmployeeEO1.EMAIL_ADDRESS AS MANAGER_EMAIL, flkp.meaning AS POSITION DISPLAY, decode(nvl(to_char(EmployeeEO.END_DATE), 'N'), 'N', 'DeleteDisabled', 'DeleteEnabled') AS DELETE_SWITCHER, decode(nvl(to_char(EmployeeEO.END_DATE), 'Y'), 'Y', 'okind status.gif', 'criticalind status.gif') AS EMPLOYEE STATUS FROM FWK_TBX_EMPLOYEES EmployeeEO, FWK_TBX_EMPLOYEES EmployeeEO1, FWK_TBX_LOOKUP_CODES_VL flkp WHERE EmployeeEO.MANAGER ID = EmployeeEO1.EMPLOYEE ID (+) AND flkp.lookup_code = EmployeeEO.POSITION_CODE AND flkp.lookup type = 'FWK TBX POSITIONS'

任务4.2: 添加图片项目到 EmpSearchPG ResultsTable 中

● 在结构窗口中选择 ResultsTable, 右键从上下文菜单中选择 New > Item

属性	值
ID	EmpStatus
Region Style	Image
Data Type	VARCHAR2
View Instance	EmployeeSummaryVO1
View Attribute	EmployeeStatus
Additional Text	Current employment status
Height	16
Width	16
Prompt	Status

● 选择新建的项目,设置如下的属性值:

任务4.2: 更改 EmployeeResultsCO 控制器

添加如下的方法到 EmployeeResultsCO 控制器中 processRequest() 方法中来实现图像项目和视图对象属性 EmployeeStatus 的绑定。

import oracle.cabo.ui.data.BoundValue; import oracle.cabo.ui.data.bind.ConcatBoundValue;

import oracle.cabo.ui.data.bind.FixedBoundValue;

import oracle.apps.fnd.common.MessageToken;

import oracle.apps.fnd.framework.OAException;

import oracle.apps.fnd.framework.webui.OADataBoundValueViewObject;

import oracle.apps.fnd.framework.webui.beans.OAImageBean;

•••

public void processRequest(OAPageContext pageContext, OAWebBean webBean) // Implement the bound value for the Status Image OAImageBean statusImageBean = (OAImageBean)table.findIndexedChildRecursive("EmpStatus"); if (statusImageBean == null) MessageToken[] tokens = { new MessageToken("OBJECT_NAME", "EmpStatus") }; throw new OAException("AK", "FWK TBX OBJECT NOT FOUND", tokens); // Define the OA Framework image directory FixedBoundValue imageDirectory = new FixedBoundValue(APPS_MEDIA_DIRECTORY); // Define a binding between the image bean and the view object attribute // that it will reference to get the appropriate .gif image value name. // Note that the corresponding attribute values are obtained using a // decode() in the EmployeeSummaryVO view object. OADataBoundValueViewObject statusBinding = new OADataBoundValueViewObject (statusImageBean, "EmployeeStatus"); // Concatenate the image directory with the actual image name (as retrieved // from the view object attribute decode() statement) ConcatBoundValue statusCBV = new ConcatBoundValue(new BoundValue[] {imageDirectory, statusBinding}); // Tell the image bean where to get the image source attribute statusImageBean.setAttributeValue(SOURCE ATTR, statusCBV); // For accessibility compliance, you always specify the alternate text for an // image. Note that you should never use static text as shown (always source // translatable text from Message Dictionary when setting display text values // programmatically), and ideally, the alternate text should in this case // should clearly indicate the status the image represents. // Generally, we would recommend that you use a Switcher as shown for the // Delete column to easily show different images with associated alternate // text, but we wanted to show how to use a bound value also in this lab.

statusImageBean.setAttributeValue(SHORT_DESC_ATTR, "Employee status indicator");

任务4.3: 测试

Open and Closed Issues for this Deliverable

Open Issues

ID	Issue	Resolution	Responsibility	Target Date	Impact Date

Closed Issues

ID	Issue	Resolution	Responsibility	Target Date	Impact Date