ADF Code Corner

87. How to improve ADF Business Components LOV performance using shared application modules



Abstract:

The performance of static list data used in Oracle ADF applications improves using shared application modules in ADF Business Components. SQL statement for static list will be executed only once per application scope, retrieved static data will be preserved in memory and shared across users, thus avoiding expensive queries being executed for each user session. Oracle JDeveloper 11g R2 (11.1.2) has an additional configuration parameter(*jbo.shared.txn*) you set to minimize number of open database connections for Shared Application modules.

Andrejus Baranovskis, Red Samurai Consulting twitter.com/andrejusb 29-JUL-2011

Oracle ADF Code Corner is a loose blog-style series of how-to documents that provide solutions to real world coding problems.

Disclaimer: All samples are provided as is with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: <u>http://forums.oracle.com/forums/forum.jspa?forumID=83</u>

Introduction

This example describes two solutions. First presents usage of shared application modules to reduce SQL execution and optimize data retrieval for static data lists across application users. Second explains how to configure the jbo.shared.txn property with JDeveloper 11g R2 11.1.2.0.0 to minimize the number of opened database connections for shared application modules.

Sample application implements two static lists (Jobs/Departments) and loads these lists into application memory to share between users.

Ð					
	JobId	Salary	CommissionPct	ManagerId	DepartmentId
	Sales Manager	24000			Executive
	Administration Vice Pr	17000		100	Control And Credit
	Administration Vice Pr	17		×	Executive
	Programmer	10			IT
	Programmer	6C EmployeeId	100		Control And Credit
	Programmer	25 FirstName	Steven		IT
	Programmer	45 * LastName	Kina		IT
	Programmer	42			IT
	Finance Manager	12 * Email	SKING		Treasury
	Accountant	90 PhoneNumber	515.123.4567		Finance
	Accountant	82 * HireDate	6/17/1987	20	Finance
	Accountant	77		_ ~	Finance
	Accountant	78 * JobId	Sales Manager	¥	Finance
	Accountant	69 Salary	240	00	Finance
	Purchasing Manager	11 Convincion Data			Purchasing
	Purchasing Clerk	31 CommissionPec			Purchasing
	Purchasing Clerk	29 ManagerId			Purchasing
	Purchasing Clerk	28 DepartmentId	Executive	V	Purchasing
	Purchasing Clerk	26			Purchasing
	Purchasing Clerk	25 Save	Administration	8	Purchasing
	Stock Manager	80	Marketing		Shipping
	Stock Manager	8200	Purchasing Human Resources		Shipping
	Stock Manager	7900	Shippina		Shipping
	Stock Manager	6500	IT		Shipping
	Stock Manager	5800	Public Relations	∃	Shipping
	Stock Clerk	3200	Sales		Shipping
	Stock Clerk	3000	Einance		Shipping
	Stock Clerk	2400	Accounting		Shipping
	Stock Clerk	2200	Treasury		Shipping
	Stock Clerk	3300	Corporate Tax	<u> </u>	Shipping
	Stock Clerk	2800	Sharebolder Services		Shipping
	Stock Clerk	2500	Benefits		Shipping
	Stock Clerk	2100	Manufacturing		Shipping
	Stock Clerk	3300	Construction		Shipping
	Stock Clerk	2900	Contracting	×.	Shipping
	Stock Clerk	2400		122	Shipping

Later in the article we describe how to optimize opened database connections for shared application module instances with jbo.shared.txn property.



Implementing this solution

The sample application that you can download for this article contains three ADF Business Component model projects and one ViewController project.

The **HrCommonModel** and **OrgCommonModel** contain Jobs and Departments LOV's, both of these projects are packaged into ADF libraries and use different application modules.



The **Model** project imports the generated ADF libraries, one containing the **Jobs LOV** and the other containing the **Departments LOV**.



Now you can declare imported AM's to be shared across application scope, this can be done through Application Module Instances section. For this, open the application model project, **Model** in this sample, and select the **ADF Business Components** | **Application Module Instances** node. With the **Application** tab selected find and select the **HrCommonModule** and **OrgCommonModule** entries.

😂 Project Properties - C:\JDeveloper\myworkr2\SharedAMSample\Model\Model.jpr 🛛 🛛 🔀								
Q Search ADF Business Components: Application Module Instances								
Project Source Paths ADF Business Components ADF Business Components Application Module Inst Base Classes Imports Options Packages Preloaded Objects Property Sets Regex Settings Registreed Rules Substitutions View Object We ADF Model	Application Session Select an application module from the available tree, and click's' to the shared application module. The instance can be shared in the and available Application Modules: Application Modules: Available Application Modules: Application Modules: Application Model Image: Imag	create a named instance of oplication level. dyle Instances: amurai.model.Model amonModule ammonModule						
ADF View Ant Compiler Dependencies Deployment EJB Module Extension Facelets Tag Libraries	Default Name: Instance Nam Definition:	e:						
Help	Help OK Cancel							

Next, you use the static list data in the main **Model** project. For this, instead of using VO instance from the Model project directly, you select a View Object instance from one of the shared application modules and declare it as the list data source.

How to improve ADF Business Components LOV performance using shared application modules

🖄 Employees Vie	w.xml ×		😂 Edit List of Va	alues		
			List of Values <u>N</u> ame	: LOV_JobId		
General Entity Objects Attributes Business Rules	s Attributes View object attributes can be mapped to entity attributes, calcula ss Q Name		Configuration Select a view acc that maps to the List <u>D</u> ata Source:	UI Hints cessor for the list data source, and then choose the list attribute current view object attribute. HrCommonModule_JobsLovView1		
Query	Name	Tune	List Attribute:	lohid 👻		
Java	EmployeeId	Integer	ESC Heribaco.	Jobid		
View Accessors	FirstName	String f	List Return Value	·s		
List UI Hints	LastName	View Accessors				
OI Categories	Email					
	PhoneNumber	Select a view object or shared view	v instance and shuttle it to the selected list to create a view accessor.			
	HireDate					
	JobId	Available View Objects:		View Accessors: Edit Override		
	Salary	com.redsamurai.model.Model	Ling and a state	Market Street S		
	CommissionPct	JobsLovView1	Arcommonimodule	OrgCommonModule_boostownew1		
	ManagerId	🗄 🍈 com.redsamurai.hr.model.	views.lovs	· · · ·		
	DepartmentId	⊕	ws	>		
	Details UI Hints Enl	com.redsamurai.org.model	I.OrgCommonModule	<		
	Enable this attribute to di					
	Lists of Values:					
	Name	Name: HrCommonModule_JobsLov	vView1_1	Accessor Name: HrCommonModule_JobsLovView1		
	CONTROL			Definition: com.redsamurai.hr.model.views.lov		
		Help		OK Cancel		

The rest is business as usually for those savvy with model driven List of Value sin ADF BC.

Edit List of Va	lues		
ist of Values <u>N</u> ame:	LOV_JobId		
Configuration	JI Hints		
Select a view acce that maps to the c	essor for the list data : current view object at	source, and then choose the tribute.	list attribute
List <u>D</u> ata Source:	HrCommonModule_3	lobsLovView1	
List <u>A</u> ttribute:	JobId		•
always returns a v	value to the attribute	for which the list is defined).	object (it
View Attribute		List Attribute	
JobId		JobId	

Set the **UI Hints** to render the List of Values as a Choice List and select the attributes to display in the list.

The choice list as a type is used for a reason! List of Values, like input Text with List of Values, should not be used with shared application modules, because shared data collection should not be filtered as it would impact other user sessions as well.

Now let's see how it works, first user is loading ADF page with choice list data.

Ь					
	JobId	Salary	CommissionPct	ManagerId	DepartmentId
	Sales Manager	24000			Executive
	Administration Vice Pr	17000		100	Control And Credit
	Administration Vice Pr	17		\mathbf{X}	Executive
	Programmer	10			IT
	Programmer	6C EmployeeId	100		Control And Credit
	Programmer	25 FirstName	Steven		IT
	Programmer	45 * LastName	King		IT
	Programmer	42	i ving		IT
	Finance Manager	12 * Email	SKING		Treasury
	Accountant	90 PhoneNumber	515.123.4567		Finance
	Accountant	82 * HireDate	6/17/1987	- 124	Finance
	Accountant	77	0/17/1507		Finance
	Accountant	78 * JobId	Sales Manager	×	Finance
	Accountant	69 Salary	240	00	Finance
	Purchasing Manager	11 Convincion Data			Purchasing
	Purchasing Clerk	31 CommissionPcc			Purchasing
	Purchasing Clerk	29 ManagerId			Purchasing
	Purchasing Clerk	28 DepartmentId	Executive		Purchasing
	Purchasing Clerk	26			Purchasing
	Purchasing Clerk	25 Save	Administration	<u> </u>	Purchasing
	Stock Manager	80	Marketing		Shipping
	Stock Manager	8200	Human Resources		Shipping
	Stock Manager	7900	Shipping		Shipping
	Stock Manager	6500	IT		Shipping
	Stock Manager	5800	Public Relations	∃	Shipping
	Stock Clerk	3200	Sales Executive		Shipping
	Stock Clerk	3000	Finance		Shipping
	Stock Clerk	2400	Accounting		Shipping
	Stock Clerk	2200	Treasury		Shipping
	Stock Clerk	3300	Corporate Tax Coptrol And Credit		Shipping
	Stock Clerk	2800	Shareholder Services		Shipping
	Stock Clerk	2500	Benefits		Shipping
	Stock Clerk	2100	Manufacturing		Shipping
	Stock Clerk	3300	Construction		Shipping
	Stock Clerk	2900	contracting		Shipping
	Stock Clerk	2400		122	Shipping

During first load of ADF page for the first user, SQL statements for choice lists are generated and executed in database. One for first choice list – Jobs.

And another SQL statement, for second choice list – Departments.



When a next user accesses the application, there is no SQL statements generated for fetching the list data, as data is fetched from the memory. This happens, because the LOVs are populated from a shared application module instance, which means that no additional module instance will be created for new users.

This means, static data for all uses will be loaded really fast. Typically enterprise applications tend to have many static data choice lists, so it can be really good performance improvement to use shared application modules for such type of lists.

İ	JobId	Salary	CommissionPct	ManagerId	DepartmentId
1	Sales Manager	24000			Executive
	Administration Vic	17000		100	Control And Credit
	Administration Vic	17000		100	Executive
	Programmer	10000		102	IT
	Programmer	6000		103	Control And Credit
	Programmer	2500		103	IT
	Programmer	4500		103	IT
	Programmer	4200		103	IT
	Finance Manager	12000		101	Treasury
	Accountant	9000		108	Finance
	Accountant	8		×	inance
	Accountant	2			inance
	Accountant	; EmployeeId	108		inance
	Accountant	6 FirstName	Nancy		inance
	Purchasing Manager	1 *LactName	Greenberg		urchasing
	Purchasing Clerk	Lasundine	Greenberg		urchasing
	Purchasing Clerk	2 * Email	NGREENBE		urchasing
	Purchasing Clerk	2 PhoneNumbe	President		urchasing
	Purchasing Clerk	2 XIEns Date	Administration Vi	ce President	urchasing
	Purchasing Clerk	- HireDat	Administration As	sistant	urchasing
	Stock Manager	₹ JobI 🗸	Finance Manager		shipping
	Stock Manager	8 Salar	Accountant		hipping
	Stock Manager	7	Accounting Mana	ger	hipping
	Stock Manager	(CommissionPo	Public Accountant	t	hipping
	Stock Manager	5 ManagerI	Sales Manager		hipping
	Stock Clerk	DepartmentI	Sales Representat	ive	hipping
	Stock Clerk	Department	Purchasing Manag	ger	shipping
	Stock Clerk	2 Save	Purchasing Clerk		shipping
	Stock Clerk	2	Stock Manager		hipping
	Stock Clerk	3300	Stock Clerk	101	Shipping
	Stock Clerk	2800	Shipping Clerk	121	Shipping
	Stock Clerk	2500	Programmer	121	Shipping
	Stock Clerk	2100	Marketing Manag	er	Shipping
	Stock Clerk	3300	Marketing Repres	entative	Shipping
	Stock Clerk	2900	Human Resources	Representative	Shipping
	Stock Clerk	2400	Public Relations R	epresentative	Shipping
	Stock Clerk	2200		122	Shipping
1					

Controlling the number of database connections

Let's move on to the second part of this article and look at the new *jbo.shared.txn* property available in Oracle JDeveloper 11g R2 (11.1.2.0.0).

When you check, how many database connections are opened for the sample application, will count three. The is one connection for each shared Application Module and one for the application itself.



You can however configure the initial number of opened database connections to be 2 (one for main application and another one to be shared between two shared application modules). Fr this, you use the *jbo.shared.txn* property.

Before configuring *jbo.shared.txn*, have a look into DataBinding.cpx, located in the ViewController project, which has the application ADF Business Component Data Control reference is defined.



The Data Control reference in ADF usually points to the **Local** application module configuration (see highlighted text). However, when application modules are defined to be shared, the ADF framework internally will use a **Shared** configuration. This means, that the *jbo.shared.txn* property must not be set for the Local Application Module configuration, but the **Shared**.

For this, open the application module configuration in the **HrCommonModel** project. Select the shared configuration as in the image shown below. Then provide a name value to the *jbo.shared.txn* property, for example: SharedAMGroup. If you use the same *jbo.shared.txn* value in other shared application modules, then, at runtime, ADF Business Components nests all of them under the same transaction so only a single database connection is used.

Configurations	ĺ	৬ Edit Configurati	ion			×
The configurations allow Business Comp Configuration		Configuration Name: HrCommonModuleSh		Shared		
		Application Module	Pooling and Scalabili	ty Properties		
		Property		Value		
HrCommoniModuleShar	ea	jbo.saveforlater		false		
		jbo.security.config jbo.security.context jbo.security.enforce jbo.security.loginmodule jbo.server.internal_connection jbo.server.retainAssocAccessor				
Default Configuration:	HrCommonMod			oracle.security.jazn		
				None		
Configuration File:	bc4j.xcfq			oracle.security.jazn.oc4j.JAZNUserMana		serMana
				false		
		jbo.server.useNullDbTransaction		false		
		jbo.shared.txn		SharedAMGroup		
		jbo.simulate.remote		false		
		jbo.snapshotstore.undo jbo.sparse.array.threshold jbo.sql92.DbTimeQuery jbo.sql92.JdbcDriverClass jbo.sql92.LockTrailer jbo.SQLBuilder		persistent		
				20		-
				select sysdate from dual		
				FOR UPDATE		
				Oracle		
		jbo.stringmanager.factory.class		use_default		
		joo.tmpair the two discourses level				
				falce v		
		Define a newed PC41	kunnanskinn fav use h	u an Annli	A dia	Deceb
		Press F9 to sort table	by property name.	y an Appil	ж <u>а</u> а	Reset
		Help			ОК	Cancel

Repeat the same for OrgCommonModel project, specify jbo.shared.txn = SharedAMGroup.

Configurations		😓 Edit Configuration 🛛 🔀			
The configurations allow Business Co		Configuration Name: OrgCommonModuleShared			
Configuration		Application Module	Pooling and Scalability	/ Properties	
OrgCommonModuleLocal		Property		Value	
OrgCommonModuleSha	ared	jbo.security.enforce		None	▲
		jbo.security.loginmodu	le (oracle.security.jazn.od	:4j.JAZNUserMana
Default Configuration:	OrgCommor	jbo.server.internal_co	nnection		
	orgcommor	jbo.server.retainAssocAccessor		false	
Configuration File:	bc4i.xcfa	jbo.server.useNullDbT	ransaction	false	
coningaradorn no.	bernwerg	jbo.shared.txn		SharedAMGroup	
		jbo.simulate.remote		false	
		jbo.snapshotstore.undo		persistent	
		jbo.sparse.array.threshold		20	
		jbo.sql92.DbTimeQuer	y :	select sysdate from du	ial
		jbo.sql92.JdbcDriverCl	ass		
		jbo.sql92.LockTrailer	1	FOR UPDATE	
		jbo.SQLBuilder		Oracle	
		jbo.stringmanager.factory.class		use_default	
		jbo.tmpdir			
		jbo.txn.disconnect_level		0	
		jbo.txn.handleafterpostexc		false	
		jbo.txn_seq_inc		50	
		jbo.txn_seq_name		PS_TXN_seq	
		ibo typ table name		PS TXN	
		Define a named BC4J Press F9 to sort table <u>H</u> elp	transaction for use by by property name.	an Appli Add	I <u>R</u> eset

When you run the application with the *jbo.shared.txn* property applied, only two database connections are being opened for the first application access – one for main application and another one for both shared application modules nested into single group.



When a next user accesses the application, only one additional database connection is opened because the shared application module data is already available in memory from where it is reused.



Same behavior is repeated for third user, etc. – only one database connection will be added for main application module.



Running the Sample

You can download the Oracle JDeveloper 11g R2 workspace for this example as sample #87 from the ADF Code Corner website

http://www.oracle.com/technetwork/developer-tools/adf/learnmore/index-101235.html

Configure the database connection used by the workspace to access the Oracle HR schema.

In Oracle JDeveloper 11g R2 11.1.2.0.0, select the main.jsf page in the adfc-config.xml unbounded task flow in the ViewController project and choose **Run** from the context menu.

You can try to open application URL from different browsers, using different sessions – review generated log messages to see that SQL statements for shared static lists are generated only once, on first access by first user. Using WebLogic Console open dashboard screen to review currently opened database connections.

RELATED DOCOMENTATION

10.2.6 What You May Need to Know About Shared Application Modules and Connection Pooling
 <u>http://download.oracle.com/docs/cd/E16162_01/web.1112/e16182/bclookups.htm#BABCEIF</u>
 <u>B</u>