Oracle Coherence 3.6: Share and Manage Data in Clusters

Activity Guide

D66791GC11

Edition 1.1

October 2012

D79400

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Authors

Mark Lindros

Al Saganich

Technical Contributors and Reviewers

Brian Oliver

Patrick Peralta

Noah Arliss

Christer Fahlgren

David Guy

Robert Lee

Thomas Beerbower

John Speidel

David Leibs

Patrick Fry

Jason Howes

Michele Diserio

Peter Utzschneider

Craig Blitz

Cameron Purdy

Alex Gleyzer

Serge Moiseev

Rob Misek

Randy Stafford

David Felcey

Craig Blitz

Tom Pflaeffle

Madhav Sathe

Rao Bhethanabotla

Editors

Steve Friedburg

Vijayalakshmi Narasimhan

Graphic Designer

Maheshwari Krishnamurthy

Publishers

Sumesh Koshi

Veena Narasimhan

Copyright © 2010, 2012, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
The U.S. Government’s rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Oracle University and ORACLE CORPORATION use only

Table of Contents

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 0

Chapter 1

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 0

Practices Overview

The practices in this course are organized using a straightforward sectioning. Each lesson has one or more practices and begins with a problem statement that reinforces the business purpose behind the skills and technology addressed in the practice. The **Practices Overview** section describes, in non-technical terms, the issues at hand and the business problem, as would be encountered in a real-world environment. A **Skills Learned** section follows the **Problem Statement** section and specifies the skills the students should take away on completion of the exercise.

Skills Learned

The aim of this section is to outline the specific skills a student acquires by completing this practice. These are the learning objectives. The **Skills Learned** section itemizes each of the skills for easy review. The following list provides an example of the skills a student acquires in the practice:

- Configuring a database
- Installing a product
- Starting and stopping product instances
- Exploring the applications

By reviewing the objectives, you see what you accomplish in the practice. The **Skills Learned** section is followed by one or more practices that reinforce the itemized skills, including the **Design**, **Overview Instructions**, and **Detailed Instructions** sections of the practice.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 0-1: Understanding Practice Structure

Overview

In this practice, you do the following:

- Learn about the Overview Instructions section of the practice
- View an example of the Detailed Instructions section

Design

The **Design** section represents, in terms typical to the problem, the design elements. Not every practice requires a **Design** section. But for the practices that do, the **Design** section can contain UML diagrams, a set of required parameters, class diagrams, or other design elements, which illustrate the problem that is solved. The **Design** section information may also be included in the **Overview Instructions** section.

Overview Instructions

The **Overview Instructions** section contains a list of one or more tasks that represent the practice at hand. These tasks are numbered in the order in which they must be performed, and detail at a high level what steps you must perform to complete an exercise, but without cookbook style instructions. Think of these instructions as a set of ingredients that an experienced cook can use to make something, without the need to follow step-by-step instructions. The overview instructions are designed with the experienced user in mind, offering challenges and opportunities to solve business problems creatively. Solution instructions typically include:

1. High-level steps, but never screenshots, and so on
2. Configuration parameters required by the high-level steps, including parameters, passwords, directories, ports, or other information required to complete the task, some of which might be represented in a Design section

Overview instructions do not include the sub-steps required to complete a given task. You should examine the overview instructions and consider them as a guide that can be used to refresh your understanding after a task has been mastered, without the need for the step-by-step approach.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Detailed Instructions

Detailed instructions contain a cookbook or procedural approach to performing a task. Detailed instructions have one heading per overview instruction, but also contain all sub-steps, data, screenshots, and result expectations for a given task. Detailed instructions provide every step required to perform a task, and are an excellent reference for the detailed process and procedure. Often as exercises progress in the course, the detailed instructions for tasks previously performed become less detailed and more general. This is because you are expected to know how to perform the task. For example, initially a set of instructions might appear as:

a) Open a command prompt and change the directory to:
```
cd c:\somepath\somedirectory
```
b) Change the directory to the `db` subdirectory.

c) Open `changeme.sh` using your favorite text editor, for example:
```
notepad changeme.cmd
```
d) Modify the script as follows:
Swap ***1*** with ***2***.

e) On completion, the script should resemble the following:
```
. . .
```
f) Execute the script using the command:
```
c:\. . . > changeme.cmd[cr]
```

Later instructions might appear as:

a) Open a command prompt and change the directory to `c:\somepath\somedirectory\db`, edit the `changeme.cmd` script as required, and execute.

A subsequent exercise might reference the step as:

a) Modify and execute the `changeme.cmd` script as required.

Solution Instructions

Detailed instructions are sometimes followed by **Solution instruction**s for more programming-focused materials. Solution instructions are typically used when a project is broken into an exercise and a solution. For example, when you write code using the Eclipse framework, two projects are provided: the starting point and the completion of the work. The solution instructions typically assist you in navigating the solved project, pointing out changes made along the way and other points of interest.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 1

Chapter 2

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 1

Practices Overview

In these practices, you will learn about the object model that is used for the labs in this course.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 1-1: Review the Retail Domain Object Model

Overview

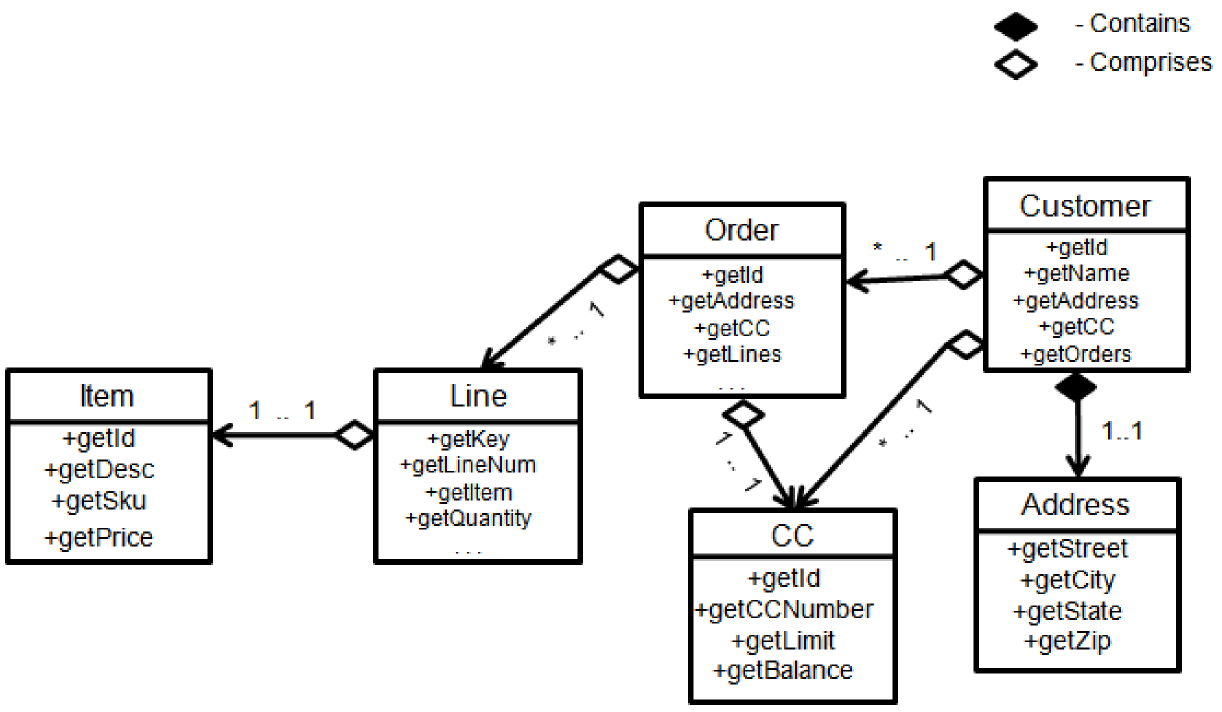
The retail domain object model is made up of a few objects that work together to handle an order processing application. This course focuses on the back end of the application and does not have a front-end component of any kind. This course uses a sample data loader that populates the cache with data that is used for the labs. This lab guides you through the different aspects that comprise the object mode and how it works.

Assumptions

The rest of the labs assume that you have learned this material. Understanding the object model and how it works is a critical part of understanding how objects can be stored within Coherence caches. You should refer back to this section whenever you need to verify how the object model works.

Review the Retail Object Model

The object model includes several relationships between objects; whereby, some objects are stored directly as part of the parent object, and other objects are referenced by a key identifier, much like a database foreign key. The UML diagram below shows the retail object classes and the relationships between them.



Folder Structure

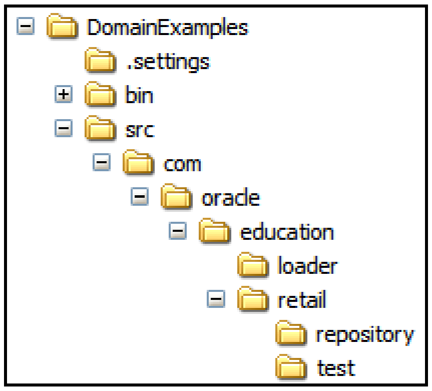
The folder structure of the source code is shown below. All code is stored within the `com.oracle.education` package. The retail classes, and their supporting test and

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

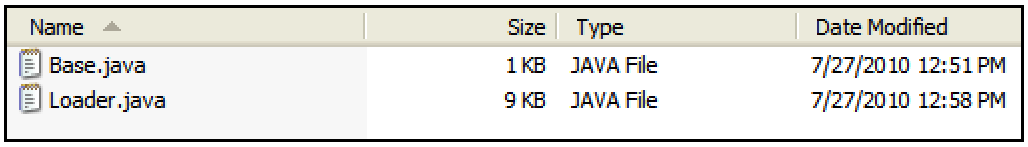
Coherence-aware classes are contained within the retail folder. The classes used to load the retail object model into a Coherence cluster using sample data are found in the loader folder.

The images below show the files in each folder.



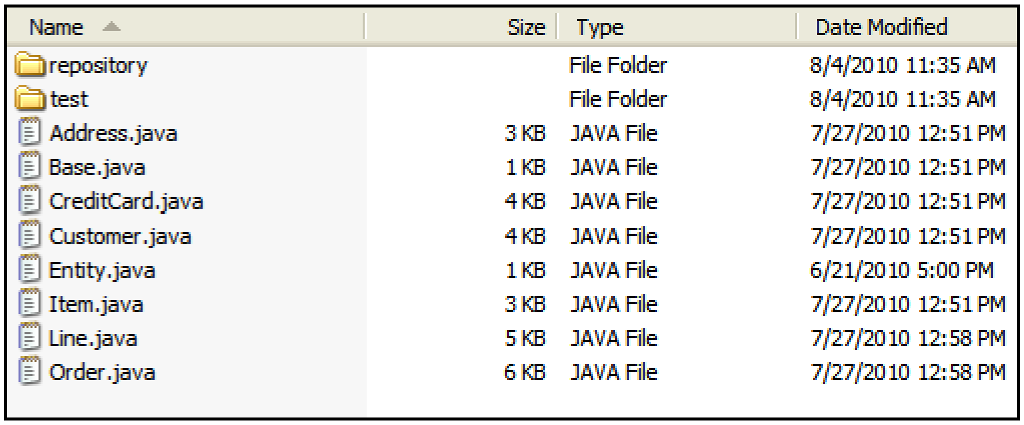
```
DomainExamples
    .settings
    bin
    src
        com
            oracle
                education
                    loader
                    retail
                        repository
                        test
```

The **loader** folder:



Name	Size	Type	Date Modified
Base.java	1 KB	JAVA File	7/27/2010 12:51 PM
Loader.java	9 KB	JAVA File	7/27/2010 12:58 PM

The **retail** folder:



Name	Size	Type	Date Modified
repository		File Folder	8/4/2010 11:35 AM
test		File Folder	8/4/2010 11:35 AM
Address.java	3 KB	JAVA File	7/27/2010 12:51 PM
Base.java	1 KB	JAVA File	7/27/2010 12:51 PM
CreditCard.java	4 KB	JAVA File	7/27/2010 12:51 PM
Customer.java	4 KB	JAVA File	7/27/2010 12:51 PM
Entity.java	1 KB	JAVA File	6/21/2010 5:00 PM
Item.java	3 KB	JAVA File	7/27/2010 12:51 PM
Line.java	5 KB	JAVA File	7/27/2010 12:58 PM
Order.java	6 KB	JAVA File	7/27/2010 12:58 PM

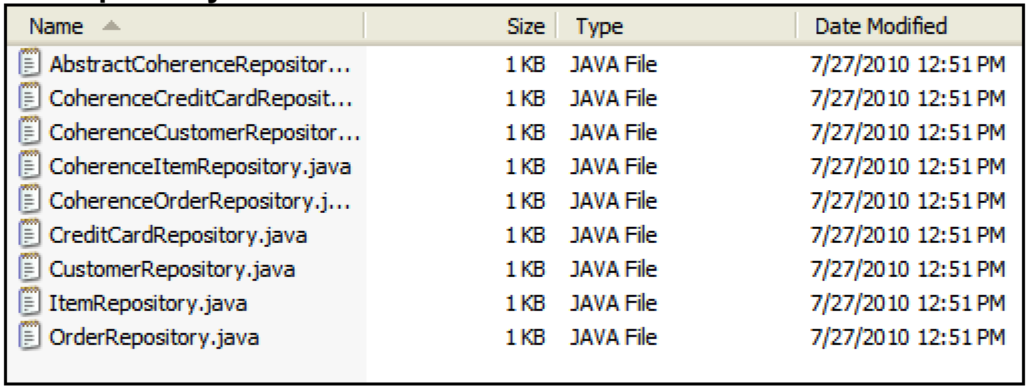
Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

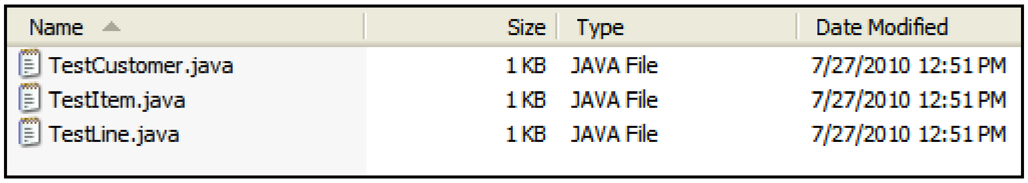
Oracle University and ORACLE CORPORATION use only

The **repository** folder:



Name	Size	Type	Date Modified
AbstractCoherenceRepositor...	1 KB	JAVA File	7/27/2010 12:51 PM
CoherenceCreditCardReposit...	1 KB	JAVA File	7/27/2010 12:51 PM
CoherenceCustomerRepositor...	1 KB	JAVA File	7/27/2010 12:51 PM
CoherenceItemRepository.java	1 KB	JAVA File	7/27/2010 12:51 PM
CoherenceOrderRepository.j...	1 KB	JAVA File	7/27/2010 12:51 PM
CreditCardRepository.java	1 KB	JAVA File	7/27/2010 12:51 PM
CustomerRepository.java	1 KB	JAVA File	7/27/2010 12:51 PM
ItemRepository.java	1 KB	JAVA File	7/27/2010 12:51 PM
OrderRepository.java	1 KB	JAVA File	7/27/2010 12:51 PM

The **test** folder:



Name	Size	Type	Date Modified
TestCustomer.java	1 KB	JAVA File	7/27/2010 12:51 PM
TestItem.java	1 KB	JAVA File	7/27/2010 12:51 PM
TestLine.java	1 KB	JAVA File	7/27/2010 12:51 PM

Classes

Retail Classes

Note: All concrete retail subclasses also implement `toString()`, `hashCode()`, and `equals()` methods. Also note that the retail classes themselves are not aware of Coherence in any way. They use a set of repository classes to integrate with Coherence, which are discussed in the next section.

File	Description
`Base.java`	The base class for all retail objects that have an ID. This class provides a constructor that is used to create the generated ID for new retail objects. This class provides the following methods: • `Base(Class ItemClass)` • `IdentityGenerator<Long> getGenId()`
`Entity.java`	An interface that is implemented by all retail objects that have an ID. It uses the Java generics feature to allow for keys of any object type. The course only uses Long ID types for the most part. This class defines the following methods: • `T getId()` • `String toString()`

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

`Item.java`	This class represents the items that make up the inventory of the retail domain model. All items that are available are stored in an Item object and stored in a Coherence cache. This class provides the following methods: • `Item(String description, float price, String sku)` • `Long getId()` • `String getDescription()` • `void setDescription(String description)` • `float getPrice()` • `void setPrice(float price)` • `String getSku()` • `void setSku(String sku)`
`Customer.java`	This class represents the customers that are part of the system that can place orders. A customer has an ID, a name, an address, one or more credit cards, and zero or more orders. The IDs for the customer's credit cards and orders are stored as part of the Customer object in the Coherence cache, and the repository fields used to read credit card and order data are marked as `transient` so the related objects are not serialized and stored in the cache as part of the Customer object. The cache name associated with the Customer object is called, Customers. This class provides the following methods: • `Customer()` • `Customer(String name, Address address)` • `Long getId()` • `String getName()` • `void setName(String name)` • `Address getAddress()` • `void setAddress(Address address)` • `private CreditCardRepository getCreditCardRepository()` • `Collection<CreditCard> getCreditCards()` • `void addCreditCard(CreditCard creditCard)` • `private OrderRepository getOrderRepository()` • `Collection<Order> getOrders()` • `void addOrder(Order order)`

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

`Address.java`	This class represents a customer's address. It does not have an ID and is serialized and stored directly in the Coherence cache as part of the Customer and Order objects. This class provides the following methods: • `String getStreet()` • `void setStreet(String street)` • `String getCity()` • `void setCity(String city)` • `String getState()` • `void setState(String state)` • `String getZip()` • `void setZip(String zip)`
`CreditCard.java`	This class represents the credit cards that belong to customers, and may be associated with specific orders. A credit card has an ID, a card number, a description, expiration date, csv code, a limit, and a balance. The cache name associated with the CreditCard object is called, CreditCards. This class provides the following methods: • `CreditCard(String cardNumber, String description, String expiration, String csv)` • `Long getId()` • `String getCreditCardNumber()` • `void setCreditCardNumber(String cardNumber)` • `String getDescription()` • `void setDescription(String description)` • `float getLimit()` • `void setLimit(float limit)` • `float getBalance()` • `void setBalance(float balance)` • `String getExpiration()` • `void setExpiration(String expiration)` • `String getCsv()` • `void setCsv(String csv)`
`Order.java`	This class represents the orders that are placed by customers. An order has an ID, an associated customer, a shipping address, an associated credit card, an order date, an order total, and zero or more line items. The IDs for the order's customer and credit card are stored as part of the Order object in the Coherence cache, and the repository fields used to read customer and credit card data are

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

	marked as `transient` so the related objects are not serialized and stored in the cache as part of the Order object. The Line objects that are associated with an order are serialized and stored in the Coherence cache as part of the Order object. The cache name associated with the Order object is called, Orders. This class provides the following methods: • `Order(Address shippingAddress, Date orderDate)` • `Long getId()` • `String getName()` • `Long getCustomerId()` • `Address getShippingAddress()` • `void setShippingAddress(Address shippingAddress)` • `Date getOrderDate()` • `void setOrderDate(Date orderDate)` • `Collection<Line> getLines()` • `void setLines(Collection<Line> lines)` • `void addLine(Line line)` • `private CustomerRepository getCustomerRepository()` • `Customer getCustomer()` • `void setCustomer(Customer customer)` • `private CreditCardRepository getCreditCardRepository()` • `CreditCard getCreditCard()` • `void setCreditCard(CreditCard creditCard)` • `floar getOrderTotal()` • `void setOrderTotal(float orderTotal)`

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

`Line.java`	This class represents the individual line items of an order. Each Line represents a single item, and the quantity of that item to purchase. A line has a key ID (represented by a `LineKey` object), an associated item, a quantity, and a line total. The ID for the line's associated item is stored as part of the Line object, which is stored as part of the Order object in the Coherence cache. The repository fields used to read item data is marked as `transient` so the related Item object is not serialized and stored in the cache as part of the Order object. This class provides the following methods: • `Line(Long orderId, long itemId)` • `LineKey getKey()` • `Long getLineNumber()` • `Long getOrderId()` • `Long getItemId()` • `void setItemId(long itemId)` • `long getQuantity()` • `void setQuantity(long quantity)` • `float getLineTotal()` • `void setLineTotal(float lineTotal)` • `void updateLineTotal()` • `private ItemRepository getItemRepository()` • `Item getItem()` • `void setItem(Item item)`

Coherence Aware Classes: Repositories

File	Description
`AbstractCoherenceRepository.java`	Provides the methods that get data from a Coherence cache, and put (or save) data to a Coherence cache. This class provides four methods: • `V get(K key)` • `Collection<V> getAll(Collection<K> keys)` • `void save(V value)` • `void saveAll(Collection<V> values)`
`ItemRepository.java`, `CoherenceItemRepository.java`	Implements the Coherence-aware class that defines the methods used to interact with a Coherence cache for

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

	the Item object. Extends `AbstractCoherenceRepository` Fields: • `static final String CACHENAME = "Items"` • `static final NamedCache m_cachedItems` Methods: • `NamedCache getCache()` • `Item getById(Long key)` • `Collection<Item> getAll(Collection<Long> keys)`
`CustomerRepository.java, CoherenceCustomerRepository.java`	Implements the Coherence-aware class that defines the methods used to interact with a Coherence cache for the Customer object. Extends `AbstractCoherenceRepository` Fields: • `static final String CACHENAME = "Customers"` • `static final NamedCache m_cachedItems` Methods: • `NamedCache getCache()` • `Item getById(Long key)` • `Collection<Customer> getAll(Collection<Long> keys)`

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

`CreditCardRepository.java`, `CoherenceCreditCardRepository.java`	Implements the Coherence-aware class that defines the methods used to interact with a Coherence cache for the CreditCard object. Extends `AbstractCoherenceRepository` Fields: • `static final String CACHENAME = "CreditCards"` • `static final NamedCache m_cachedItems` Methods: • `NamedCache getCache()` • `Item getById(Long key)` • `Collection<CreditCard> getAll(Collection<Long> keys)`
`OrderRepository.java`, `CoherenceOrderRepository.java`	Implements the Coherence-aware class that defines the methods used to interact with a Coherence cache for the Order object. Extends `AbstractCoherenceRepository` Fields: • `static final String CACHENAME = "Orders"` • `static final NamedCache m_cachedItems` Methods: • `NamedCache getCache()` • `Item getById(Long key)` • `Collection<Order> getAll(Collection<Long> keys)`

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Loader and Coherence Caches

Coherence cached data does not exist until there is some data loaded into the cache. This section describes how the data is cached when the loader program loads its sample data for the labs. The retail objects are stored into the Coherence partitioned cache as four NamedCaches:

- **Items:** All the items that are available for orders
-
- **Customers:** All the customers that are created by the loader. By default this is 10 customers. Each customer has one address, 1-3 credit cards, and 1-3 orders. The address object does not contain an ID, and is stored directly in the Customer object. Additionally, the Address object for an order is also stored in the Order object directly as a shipping address.
-
- **CreditCards:** Customer CreditCard objects are stored in their own Coherence cache as they are retrieved by the Customer objects and the Order objects.
-
- **Orders:** All the orders that belong to customers. Each order is composed of 1-3 line items. As each line item is added to the order, its value is calculated and added to the order total. Each line item has a reference to an itemId, which is actually stored in the cache. Each line item itself is stored directly within the Order object, and serialized as part of the Order object. So the Line object is serialized when an Order object is placed in the cache, but the corresponding Item object is stored separately in the cache and retrieved via the Coherence caching API via the `get()` method.

The loader can be used by any Java program by running `Loader.load()`. This can be done before or after running multiple cache servers; however, starting the cache servers first eliminates the need for partition rebalancing of the loaded data when a new cache server is started. The loader will populate a partitioned cache across any number of cache servers using the default `coherence-cache-config.xml` file found within `coherence.jar`.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 2

Chapter 3

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 2

Practices Overview

Developers must be able to install, start, stop, and otherwise manage instances Coherence, as well as Eclipse, to successfully develop applications in a safe, Eclipse-based environment.

The objective of this practice is to assist the developer in becoming comfortable with the day-to-day activities associated with working with Coherence and Eclipse.

In this practice, you will perform the following tasks:

- Install, test, configure and start a Coherence local cluster
- Start, stop and otherwise use the Coherence console
- Create a Coherence application in Eclipse

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 2-1: Install, configure, and start a Coherence cluster

Skills Learned

At the end of this practice, you should be able to:

- Unpackaged and install Oracle Coherence
- Specify command line arguments to the Coherence server start script
- Create and use a Coherence development configuration which creates a local cluster
- Start a Coherence server instance in development mode using a custom configuration script.

Problem Statement

Developers need to work in a “sand box,” a protected area where they can work without concern for impacting, or being impacted by, others. Coherence, by its very nature, is designed to seek out and interact with other Coherence instances, immediately sharing data and creating a cluster. A “local” cluster, that is, one which does *not* interact with other instances of Coherence beyond the local environment, is required to provide this “sand box.” During this exercise, you will install, test, and configure a Coherence development environment, which runs locally.

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources`**`\`	Contains a set of practice-specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various `.jar` files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Coherence Logging Levels (`tangosol.coherence.log.level`) Specifies which logged messages will be output	
0	Only output without a logging severity level specified will be logged
1	0 plus errors
2	1 plus warnings
3	2 plus informational
4-9	3 plus debugging, the higher the level, the more messages. Default = 5
-1	Disable all output

Table 2.1

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Coherence Time to Live (`tangosol.coherence.ttl`) Number of hops a packet will traverse	
0	Single server cluster, meant to keep packets from leaving the originating machine
1	Appropriate for single switched backbones
2	Appropriate for advanced backbones with intelligent switching

Table 2.2

Overview Instructions

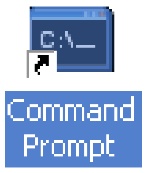
1 Install and validate Oracle Coherence.

2 Configure and start a local cluster.

Detailed Instructions

Install and validate Oracle Coherence

2-1.1 Using the desktop command line shortcut icon, open a command prompt.



Note: The command line opened by this prompt sets several environment variables, including `coherence_home`.

2-1.2 Oracle Coherence can be installed anywhere. For consistency with other products, install it into `/opt/oracle`. It is also common to use `{device:}\oracle` as an installation root directory.

Change directory to `D:\Oracle`.
Note: A common error is to unzip Coherence into the wrong directory. The next step ensures that students use correct directory.

2-1.3 Confirm that Coherence will unzip into the correct directory using the `cd` command.

```
cd [return]
```

The correct directory is `D:\Oracle`

2-1.4 A copy of Coherence 3.6 is provided in `D:\oracle\student\software`.

Unzip Coherence using a command similar to:

```
D:\Oracle> unzip student\software\coherence-java-3.6.0.zip
```

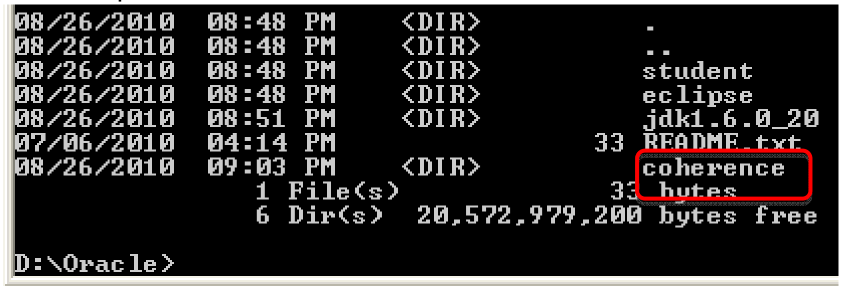
Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-1.5 Ensure that Coherence was unpacked into the correct directory using the `dir` command. It should produce a result similar to:



```
08/26/2010  08:48 PM    <DIR>          .
08/26/2010  08:48 PM    <DIR>          ..
08/26/2010  08:48 PM    <DIR>          student
08/26/2010  08:48 PM    <DIR>          eclipse
08/26/2010  08:51 PM    <DIR>          jdk1.6.0_20
07/06/2010  04:14 PM                33 README.txt
08/26/2010  09:03 PM    <DIR>          coherence
               1 File(s)             33 bytes
               6 Dir(s)  20,572,979,200 bytes free

D:\Oracle>
```

Note the Coherence directory is `D:\Oracle\coherence`

2-1.6 Change directory to `D:\Oracle\coherence\bin`.

2-1.7 Run the a multicast test using a time to live of 0 using a command similar to:

```
D:\Oracle …> start multicast-test -ttl 0
```

Note the use of the windows `start` command to create a separate command window, and start the multicast test inside it.

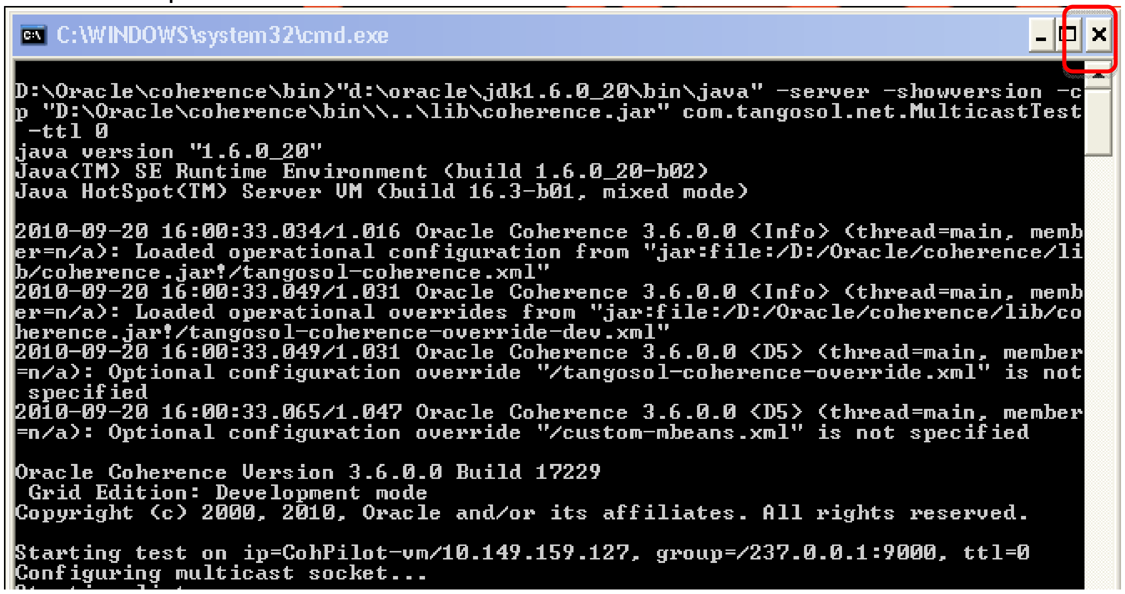
You should see output similar to:

```
D:\Oracle\coherence\bin>"d:\oracle\jdk1.6.0_20\bin\java" -server
-showversion -cp "D:\Oracle\coherence\bin\\..\lib\coherence.jar"
com.tangosol.net.MulticastTest -ttl 0
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Server VM (build 16.3-b01, mixed mode)

2010-09-20 16:00:33.034/1.016 Oracle Coherence 3.6.0.0 <Info>
(thread=main, member=n/a): Loaded operational configuration from
"jar:file:/D:/Oracle/coherence/lib/coherence.jar!/tangosol-
coherence.xml". . .

Oracle Coherence Version 3.6.0.0 Build 17229
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All
rights reserved.

Starting test on ip=CohPilot-vm/10.149.159.127,
group=/237.0.0.1:9000, ttl=0
. . .
Mon Sep 20 16:00:35 BST 2010: Sent packet 2.

. . .
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-1.8 Click **X** to stop the test and close the window.



```
D:\Oracle\coherence\bin>"d:\oracle\jdk1.6.0_20\bin\java" -server -showversion -c
p "D:\Oracle\coherence\bin\\..\lib\coherence.jar" com.tangosol.net.MulticastTest
 -ttl 0
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Server VM (build 16.3-b01, mixed mode)

2010-09-20 16:00:33.034/1.016 Oracle Coherence 3.6.0.0 <Info> (thread=main, memb
er=n/a): Loaded operational configuration from "jar:file:/D:/Oracle/coherence/li
b/coherence.jar!/tangosol-coherence.xml"
2010-09-20 16:00:33.049/1.031 Oracle Coherence 3.6.0.0 <Info> (thread=main, memb
er=n/a): Loaded operational overrides from "jar:file:/D:/Oracle/coherence/lib/co
herence.jar!/tangosol-coherence-override-dev.xml"
2010-09-20 16:00:33.049/1.031 Oracle Coherence 3.6.0.0 <D5> (thread=main, member
=n/a): Optional configuration override "/tangosol-coherence-override.xml" is not
 specified
2010-09-20 16:00:33.065/1.047 Oracle Coherence 3.6.0.0 <D5> (thread=main, member
=n/a): Optional configuration override "/custom-mbeans.xml" is not specified

Oracle Coherence Version 3.6.0.0 Build 17229
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Starting test on ip=CohPilot-vm/10.149.159.127, group=/237.0.0.1:9000, ttl=0
Configuring multicast socket...
```

Configure and start a local Coherence cluster

2-1.9 Ensure that the current directory is `D:\Oracle\coherence\bin`.

2-1.10 Start a Coherence server instance in the current window by executing the script:

```
D:\oracle…>start cache-server.cmd
```

If you were successful, a server should start and produce only output specific to the `java` command. Verify that little or no output was produced, and then use Ctrl + C to exit the script.

2-1.11 A cache server instance should start and produce output similar to:

```
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Server VM (build 16.3-b01, mixed mode)

2010-09-20 16:47:14.862/0.875 Oracle Coherence 3.6.0.0 <Info>
(thread=main, memb
er=n/a): Loaded operational configuration from
"jar:file:/D:/Oracle/coherence/lib/coherence.jar!/tangosol-
coherence.xml"
2010-09-20 16:47:14.940/0.953 Oracle Coherence 3.6.0.0 <Info>
(thread=main, member=n/a): Loaded operational overrides from
"file:/D:/Oracle/coherence/bin/tangosol-coherence-override-dev.xml". . .
Oracle Coherence Version 3.6.0.0 Build 17229
Grid Edition: Development mode
. . .

Started DefaultCacheServer...
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-1.12 Close the command window by clicking the **X** in the upper right corner.

2-1.13 Extract the `tangosol-coherence-override-dev.xml` file from the `Coherence.jar` into the `coherence\bin` directory using a command similar to:

```
D:\. . . \bin>jar -xvf ..\lib\coherence.jar tangosol-coherence-
override-dev.xml
```

You should see output similar to:

```
inflated: tangosol-coherence-override-dev.xml
```

2-1.14 Modify the file using Notepad, and specify a **time to live** which represents no packets leaving the current machine. The result should resemble:

```
<multicast-listener>
   <time-to-live system-property=". . . ">0</time-to-live>
    <join-timeout-milliseconds>3000</join-timeout-milliseconds>
</multicast-listener>
```

Save your changes.

2-1.15 Specify a logging level of nothing. The result should resemble:

```
<logging-config>
<severity-level system-property=". . .">-1</severity-level>
    <character-limit system-
property="tangosol.coherence.log.limit">0</character-limit>
  </logging-config>
```

Save your changes and exit notepad.

2-1.16 Edit `cache-server.cmd` and search for the line which begins with:

```
"%java_exec%" -server -showversion "%java_opts%
```

2-1.17 Add the current directory at the beginning of the –`cp` argument. This will cause `DefaultCacheServer` to load the developer defaults. The new classpath should now resemble:

```
-cp "%coherence_home%\bin;%coherence_home%\lib\coherence.jar"
```

Note the new addition of the **`%coherence_home%\bin`** directory.
Save your changes and exit.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-1.18	Start a Coherence server instance in the current window by executing the script: `D:\oracle…> cache-server.cmd` If you were successful, a server should start and produce only output specific to the `java` command. Verify that little or no output was produced and then use Ctrl + C to exit the script.
2-1.19	Modify the development configuration file to specify a logging level of **5** and restart the server in a new command window. For convenience you can use a command similar to: `D:\oracle…> ` **`start cache-server.cmd`**

2-1.20 If you were successful, you should see output similar to:

```
TcpRing{Peer=(none)}
IpMonitor{AddressListSize=0}

2010-06-18 15:57:05.384/5.766 Oracle Coherence GE 3.6.0.0 DPR2
<Info> (thread=main, member=1):
Services
  (
  ClusterService{Name=Cluster, State=(SERVICE_STARTED,
          STATE_JOINED), Id=0, Version=3.6, OldestMemberId=1}
  InvocationService{Name=Management, State=(SERVICE_STARTED),
          Id=1, Version=3.1, OldestMemberId=1}
  PartitionedCache{Name=DistributedCache,
           State=(SERVICE_STARTED), LocalStorage=enabled,
           PartitionCount=257, backupCount=1,
           AssignedPartitions=257, BackupPartitions=0}
   ReplicatedCache{Name=ReplicatedCache,
           State=(SERVICE_STARTED), Id=3, Version=3.0,
           OldestMemberId=1}
   Optimistic{Name=OptimisticCache, State=(SERVICE_STARTED),
           Id=4, Version=3.0, OldestMemberId=1}
   InvocationService{Name=InvocationService,
           State=(SERVICE_STARTED), Id=5, Version=3.1,
           OldestMemberId=1}
  )

Started DefaultCacheServer...
```

2-1.21	Click X in the corner of the window, or press Ctrl + C to shut down the server instance.
2-1.22	Remove the addition to the cache server class path so that later runs of Coherence do not pick up the changed override file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Solution Instructions

Copy the `tangosol-coherence-override-dev.xml.solution` file from the `resources\Practice.02.01` to the `coherence\bin` as `tangosol-coherence-override-dev.xml`, and follow the directions from 2-1.16.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 2-2: Using the Coherence Console

Skills Learned

At the end of this practice, you should be able to:

- Start and stop the Coherence console
- Create a cache
- Add items to a cache
- Size and examine the contents of a cache

Problem Statement

The Coherence console is useful for examining cache contents, but is not considered a production tool. However, the console is still useful for testing, and can be used with custom objects, provided they are in the classpath. Note that there is no solution to this practice.

Overview Instructions

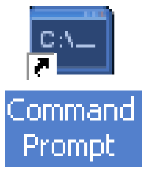
1	Start the Coherence Console.
2	Test adding elements to a cache.
3	Configure the console for local storage and add cache elements.

Detailed Instructions

Start the Coherence Console

2-2.1 Ensure that all prior instances of the **coherence-cache** command script are stopped and no other Coherence instances are running.

2-2.2 If required, open a command prompt.



2-2.3 Change directory to `d:\Oracle\coherence\bin`.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-2.4 Start the Coherence console using the `coherence.cmd` script.

```
D:\Oracle\coherence\bin> coherence.cmd
```

You will see output similar to:

```
** Starting storage disabled console **
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Server VM (build 16.3-b01, mixed mode)

2010-06-22 11:29:55.596/0.594 Oracle Coherence 3.6.0.0 DPR2
<Info> (thread=main, member=n/a): Loaded operational
configuration from
"jar:file:/D:/oracle/coherence/lib/coherence.jar!/tangosol-
coherence.xml"
. . .

2010-06-22 11:29:59.986/4.984 Oracle Coherence GE 3.6.0.0 DPR2
<D5> (thread=Invocation:Management, member=1): Service Management
joined the luster with senior service member 1

Map (?):
```

Test adding elements to a cache

2-2.5 Using the `cache` command, create a new cache.
The syntax is `cache <name>`

```
Map(?): cache testcache
```

Cache configuration data will be displayed, then the prompt will change to:

```
Map (testcache) :
```

Note: Enter names with special characters, such as spaces, by enclosing them in quotes.

2-2.6 Put an entry into the cache using the put command.
The syntax of is: `put <key> <value>`

Enter values with special characters using quotes.

For example:

```
put "A" "value for A" [cr]
```

What happens?

2-2.7 Using the `bye` command, exit the Coherence console.

Configure the console for local storage and add cache elements

2-2.8 Using Notepad, open the `coherence.cmd` file, and look for the line:

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

```
set storage_enabled=false
```

This setting defines whether the console is allowed to use local storage. Change the setting to `true`, save your changes, and restart the console.

```
set storage_enabled=true
```

2-2.9 Using the `cache` command create a new cache.
The syntax is `cache <name>`

```
Map(?): cache testcache  [CR]
```

2-2.10 Put entries into the cache.

For example:

```
put "A" "value for A" [cr]

put "MA" "Massachusetts"
put "NH" "New Hampshire"
put "CT" "Connecticut"
```

2-2.11 Using the `list` command, list all the elements in the cache.
How many items in the `test` cache?

2-2.12 Explore the cache using `size`, `remove`, and other console commands. Use the `help` command for a complete list of all console commands.

2-2.13 Exit the console using the `bye` command when complete.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 2-3: Coherence Hello World

Skills Learned

At the end of this practice, you should be able to:

- Create an Eclipse project
- Add the `coherence.jar` and associated API documentation to a project
- Add a Java class to a project
- Interact with a Coherence cache using:
  - com.tangosol.net.CacheFactory
  - com.tansosol.net.NamedCache

Problem Statement

Java applications which interface to Oracle Coherence need to start with some basics, including configuring a project, writing the core classes and interacting with Coherence through well-defined interfaces.

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources`**`\`	Contains a set of practice specific resource directories

Overview Instructions

1	Start Eclipse.
2	Create a Java project configured for Coherence.
3	Create a Java class which interacts with a Coherence cache.
4	Test using the Coherence console.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Hints:

- If you see a message in the console similar to:

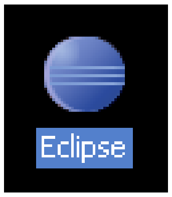
```
This member could not join the cluster because of a configuration
mismatch between this member and the configuration being used by the
rest of the cluster. This member specified a cluster name of
"cluster:xxxx" which did not match the name of the running cluster. . .
Failed to start Service "Cluster" (ServiceState=SERVICE_STOPPED,
STATE_JOINING)
```

  You may need to add `-Dtangosol.coherence.cluster=test` to all running Coherence instances, including run configurations.

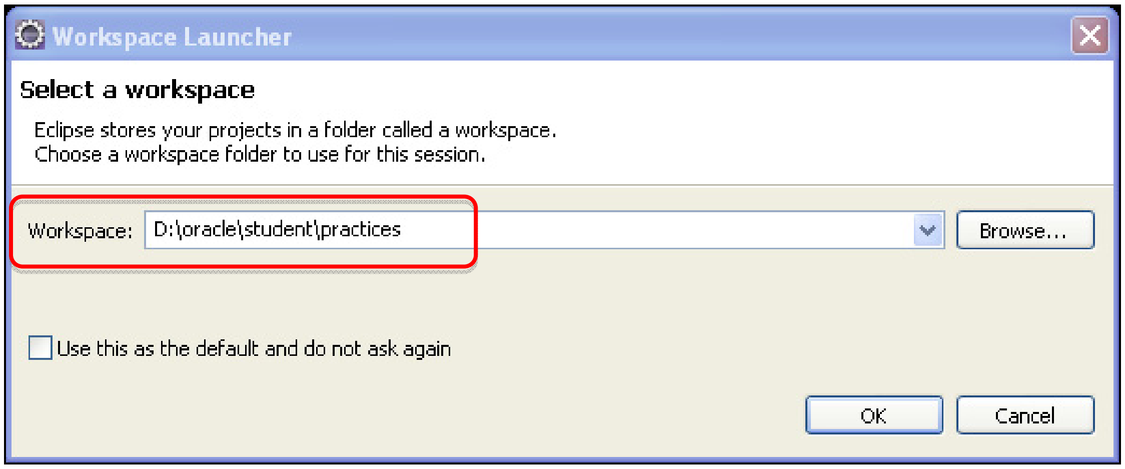
Detailed Instructions

Start Eclipse

2-3.1 Using the Eclipse desktop shortcut, start **Eclipse**.



2-3.2 When prompted for a workspace, ensure that **D:\Oracle\student\practices** is selected and click **OK**.



Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: D:\oracle\student\practices

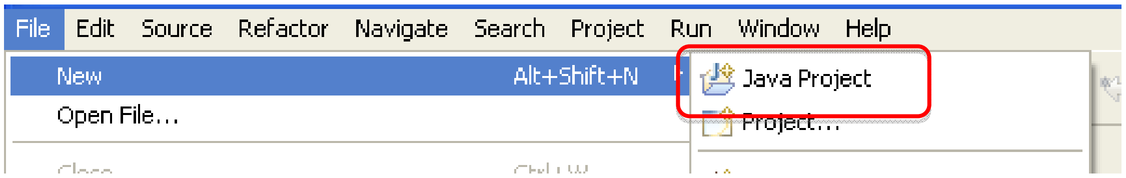
Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

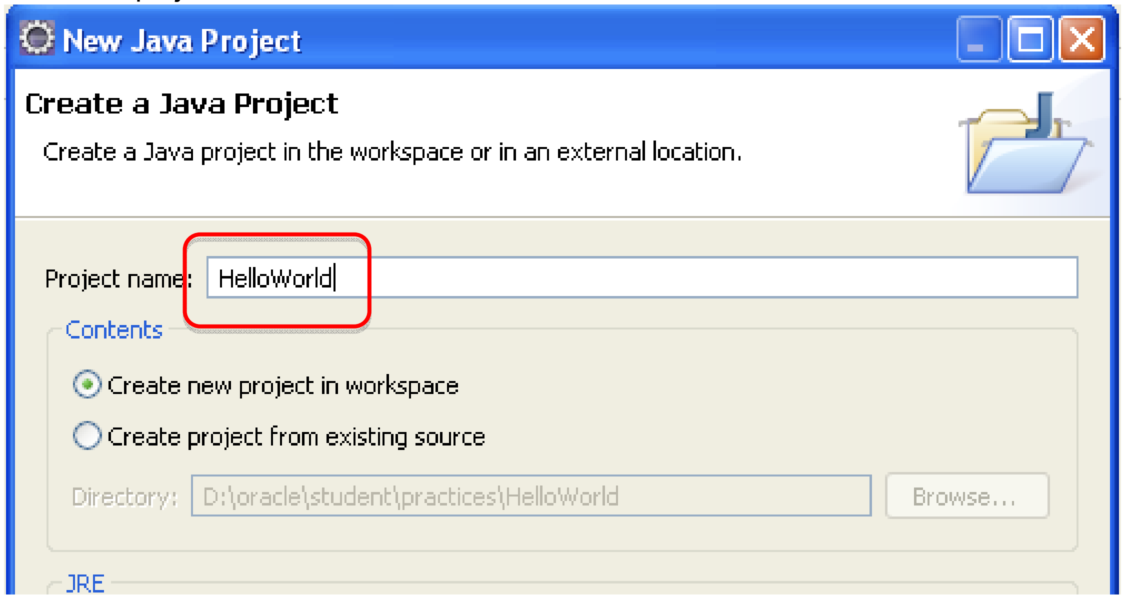
Oracle University and ORACLE CORPORATION use only

Create a Java project configured for Coherence

2-3.3 From the **File > New** menu, choose **Java Project**.



2-3.4 Name the project **HelloWorld** and click **Finish**.



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: HelloWorld

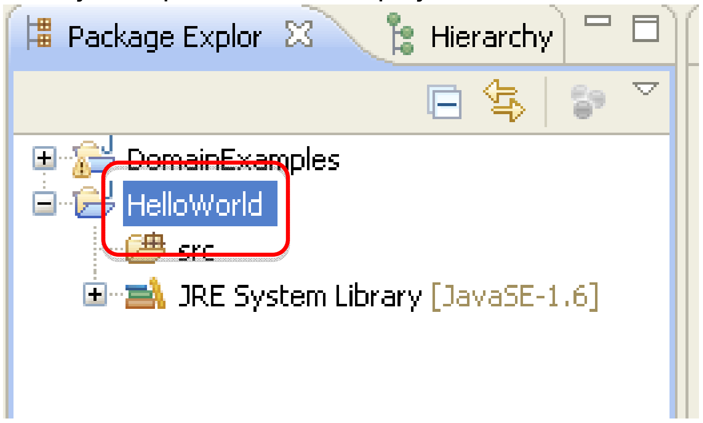
Contents

(●) Create new project in workspace

( ) Create project from existing source

Directory: D:\oracle\student\practices\HelloWorld

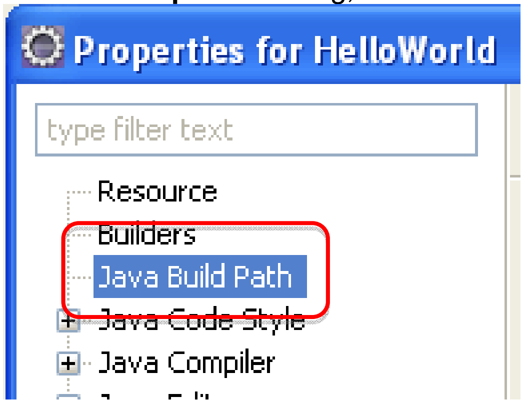
2-3.5 In Project Explorer, select the project.



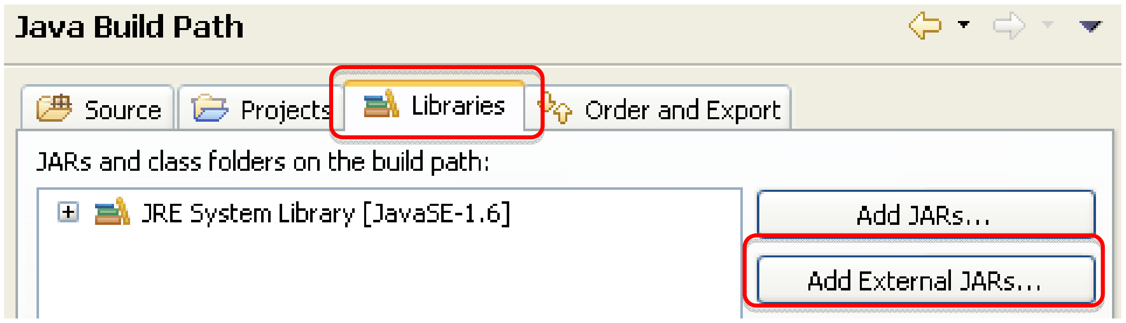
THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-3.6 From the **Properties** dialog, choose **Java Build Path**.



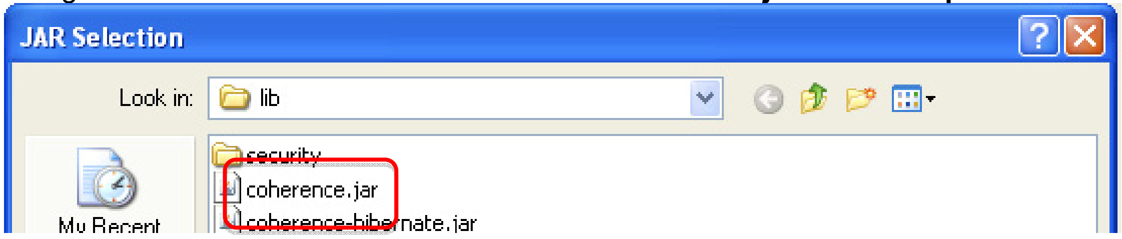
2-3.7 Click the **Libraries** tab and then **Add External Jars**.



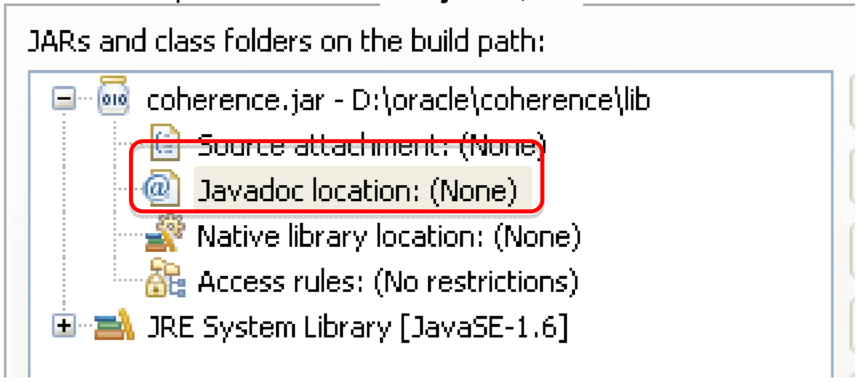
JARs and class folders on the build path:

JRE System Library [JavaSE-1.6]

2-3.8 Navigate to **D:\Oracle\coherence\lib** and select **coherence.jar** and click **Open**.



2-3.9 Select and expand the **coherence.jar** file, and select **Javadoc location**, then click **Edit**.



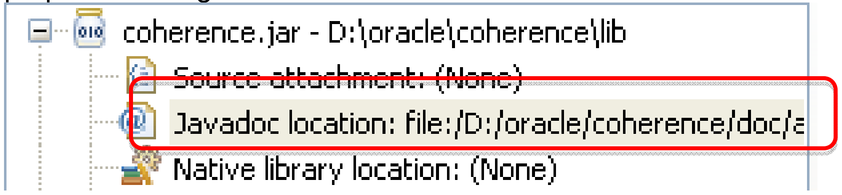
JARs and class folders on the build path:

coherence.jar - D:\oracle\coherence\lib
Source attachment: (None)
Javadoc location: (None)
Native library location: (None)
Access rules: (No restrictions)
JRE System Library [JavaSE-1.6]

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

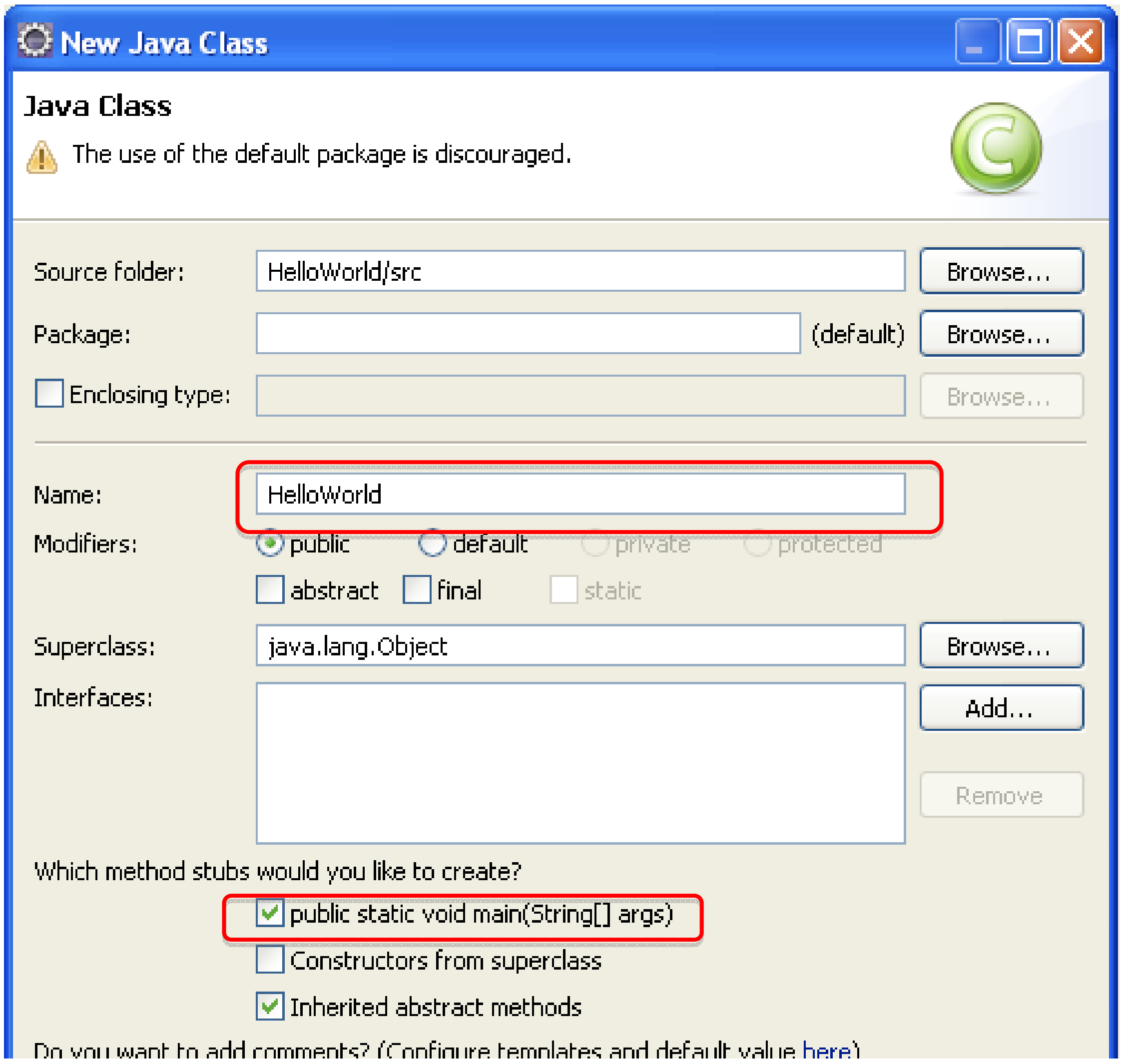
2-3.10 Navigate to the **D:\Oracle\coherence\doc\api** directory and click **OK**. The `coherence.jar` should show an associated Javadoc location. Click OK to close the properties dialog once the Javadocs have been associated with the jar file.


coherence.jar - D:\oracle\coherence\lib
~~Source attachment: (None)~~
Javadoc location: file:/D:/oracle/coherence/doc/a
Native library location: (None)

Create a Java class which interacts with a Coherence cache

2-3.11 Expand **HelloWorld** project and select the **src** folder.
Right-click and choose **New** > **Class**.

2-3.12 Name the class **HelloWorld** and select the public **static void main()** check box. Then click **Finish**.


New Java Class

Java Class

The use of the default package is discouraged.

Source folder: HelloWorld/src

Package: (default)

Enclosing type: [ ]

Name: HelloWorld

Modifiers: [✓ public] default private protected

[ ] abstract [ ] final [ ] static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

[✓] public static void main(String[] args)

[ ] Constructors from superclass

[✓] Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

2-3.13 Inside the **main** method add code to create a near cache named **testcache**".
Note: For convenience, the bulk of the method code for this exercise can be found in the resources subdirectory for **Practice.02.03** in the **code.snippet.txt** file.

Your code should resemble:

```
NamedCache cache= CacheFactory.getCache("testcache");
```

2-3.14 Using the "quick fix" functionality of Eclipse, add the required import:
To use quick fix:

a. Select the quick fix icon ( ) next to the code line in error.
b. Right-click and choose Quick Fix. Alternatively you can press Ctrl + 1.
c. From the quick fix corrections, chose the import for `com.tangosol.net.CacheFactory`.

An import resembling "`import com.tangosol.net.CacheFactory;`" will be added to the class.

2-3.15 Using quick fix, add an import for `NamedCache`.
An import resembling "`import com.tangosol.net.NamedCache;`" will be added to the class.

2-3.16 Put values for Massachusetts, New Hampshire and Connecticut into the cache using the two character state abbreviation. Notice that Massachusetts is intentionally misspelled.

```
    cache.put("MA","Massachustts");
    cache.put("NH","New Hampshire");
    cache.put("CT","Connecticut");
```

2-3.17 Add code to return the set of entries. The code should resemble:

```
Set<Map.Entry> entries = cache.entrySet(null,null);
for (Map.Entry entry: entries) {
    System.out.println("Returned '"+
                           entry.getKey()+
                           "' for '"+entry.getValue()+"'");
}
```

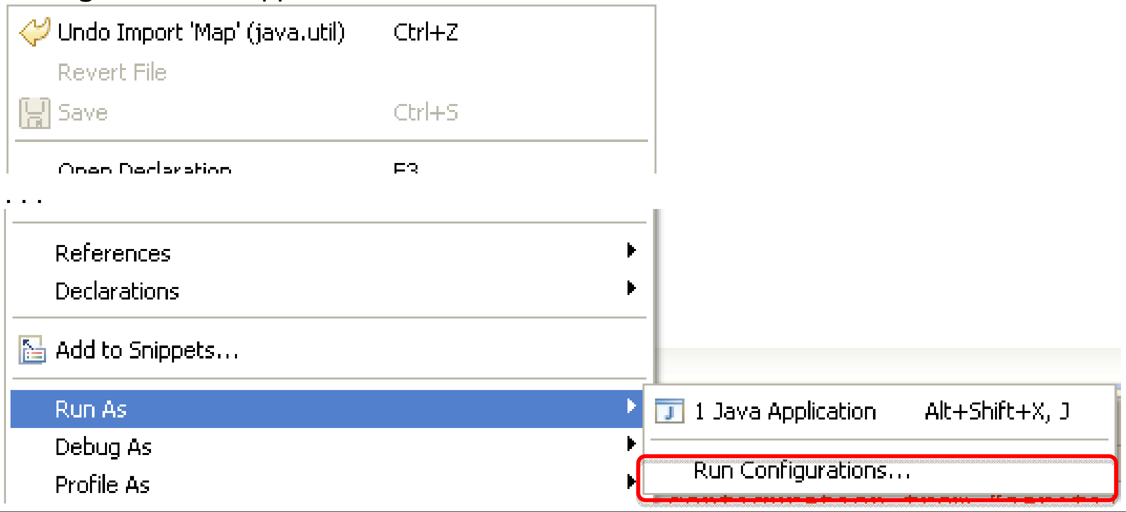
Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

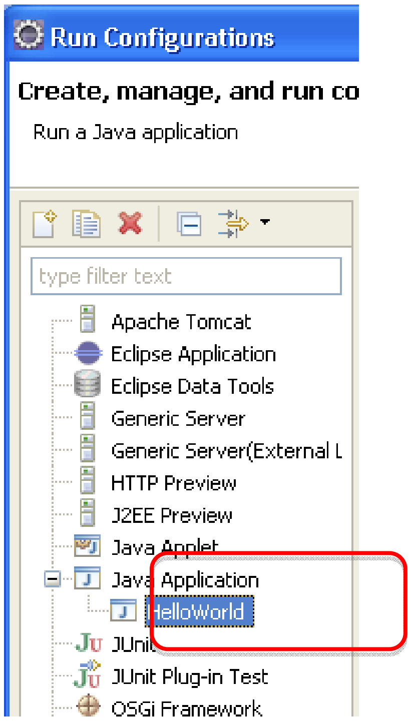
THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-3.18 Create a run configuration by right-clicking anywhere in the application and choosing **Run Configurations…** Application.



2-3.19 Double-click **Java Application**, expand if required, and then choose **HelloWorld**.

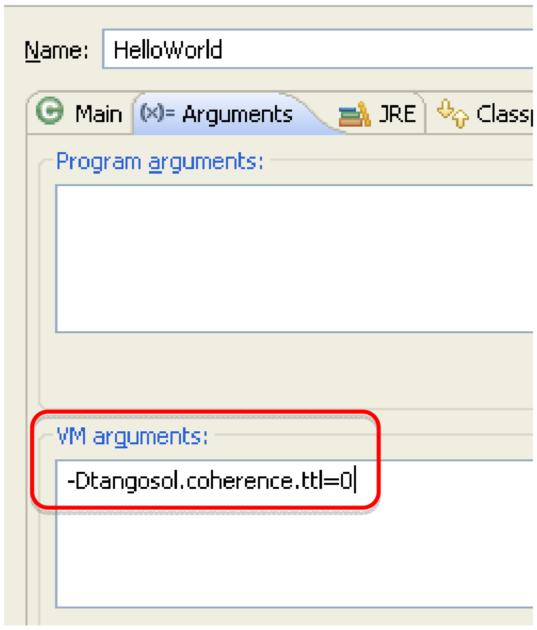


Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

2-3.20 Choose the **Arguments** tab and enter `-Dtangosol.coherence.ttl=0` in the VM arguments box.



Name: HelloWorld

VM arguments: -Dtangosol.coherence.ttl=0

2-3.21 Click **Apply** and then **Run.**

2-3.22 Examine the **Console** tab, intermixed with the Coherence output should be output similar to:

```
IpMonitor{AddressListSize=0}
Returned 'CT' for 'Connecticut'
Returned 'MA' for 'Massachustts'
Returned 'NH' for 'New Hampshire'
```

2-3.23 Add code similar to that shown below to replace the incorrectly spelled state name and return the value. Rerun the **Run Configuration** and examine the results. On the second `get`, the value returned should be the correct value for Massachusetts.

```
cache.put("MA", "Massachusetts");
String value = (String)cache.get("MA");
System.out.println("Returned '"+value+"' for 'MA'");
```

Test using the Coherence console

2-3.24 Using either an existing command prompt or opening a new one, change directory to **coherence\bin.**

```
D:\...> cd D:\oracle\coherence\bin
```

2-3.25 Start the Coherence console:

```
D:\Oracle\coherence\bin> Coherence.cmd
```

2-3.26 Using the `map` command, connect to the **testcache.**

```
Map (?): map testcache
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

2-3.27	Using the `list` command, list the contents of the cache. `Map (testcache): list` What is returned? Why?
2-3.28	Return to Eclipse and rerun the application.
2-3.29	Return to the console and list the caches contents. What is returned this time? Why? **Note:** You may see an error about cluster name mismatches during this step. See the hints section for details.
2-3.30	For convenience, return to Eclipse, select the project root, right-click and choose **Close Project.**

Solution Instructions

To run the solution, start Eclipse and connect to the **D:\Oracle\student\practices** workspace then open **Practice.02.03.solution**. Follow the directions for creating a run-configuration but select the **Practice.02.03.solution** and choose **Run**. Follow the steps from 2-3.23, ignoring any steps to add code, all of which is in the solution. See the hints section for a solution to a cluster name mismatch error if encountered when running the solution.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 3

Chapter 4

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 3

Practices Overview

Data objects are the most important artifacts in a data rich application.

The objective of this practice is to assist the developer with the creation and serialization of objects for use in a Coherence cache

In this practice, you will perform the following tasks:

- Create and Insert serializable objects into a cache specifying appropriate identity
- Extend objects to implement `ExternalizableLite` interface to improve serialization performance
- Extend objects to implement Portable Object Format to support multiple language bindings, as well as to improve serialization performance

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 3-1: Developing with Complex Objects

Skills Learned

At the end of this practice, you should be able to:

- Create an identifiable, serializable object, and add it to a cache
- Package the object(s) into a JAR which can be used with the console to display an object

Problem Statement

From:	bill.iards@retail-is-us.com
Sent:	Wednesday, June 30th, 2:45pm
To:	you@acmedev.com
Subject:	Coherence example

Hello!

As you know, we are pushing forward with our new Coherence initiative to create a retail system for retail-is-us.com. We fully expect that Oracle Coherence is the best tool to provide us the scalability and performance.

With this in mind, we need you to create a starting set of objects that can be used to represent a simple customer and associated address. Our engineering staff has provided a baseline Eclipse project, which hopefully can be adapted as an example.

I understand that to store objects in a Coherence cache, they must meet two core requirements:

1. They must support some form of serialization.
2. They must support a form of identification.

We look forward to seeing the example you create.

Regards,

Bill Iards
Retail Is Us
bill.iards@retail-is-us.com

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Resources and References

Resource Directories		
Practice Root Directory	`D:\Oracle\student\`**`practices`**`\`	Contains a set of practice projects, each representing the start of a practice
Practice	`Practice.XX.YY [.solution]`	Practices and associated solutions are named as Practice.***XX.YY***[.solution] where *XX* represents lesson and *YY* a specific practice for that lesson.

Overview Instructions

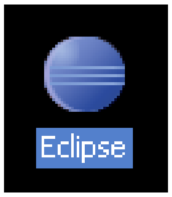
1	Open Eclipse and select a workspace.
2	Modify classes to support serialization and identity and test.
3	Export a JAR file.
4	Test using an exported JAR file and the console.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects except the current one to minimize the number of tasks displayed.

Detailed Instructions

Open Eclipse and select a workspace

3-1.1 Using the desktop shortcut, start Eclipse.



3-1.2 If prompted for a workspace, ensure that **D:\Oracle\student\practices** is selected and click **OK** and open the project **Practice.03.01**.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

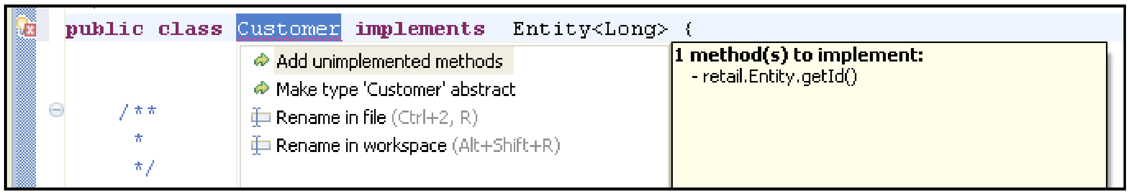
THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Modify classes to support serialization and identity

3-1.3 Navigate to the **src\retail** package and examine the **Customer.java** class. Note that it extends `Entity<Long>`, but is missing one of the required methods.
Note: The Entity interface defines two required methods: `String toString()` and `<T>getId()`. The implementation of `toString()` is provided for you.

3-1.4 Using either quick fix or by hand, add the required method.



```
public class Customer implements  Entity<Long> {

    /**
     *
     */
```

1 method(s) to implement:
- retail.Entity.getId()

Add unimplemented methods
Make type 'Customer' abstract
Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)

The required identity method should resemble:

```
public Long getId() { return id; }
```

3-1.5 Modify the class to also implement **java.io.Serializable**. You may either add the entire signature for **Serializable**, or add an **import**.

The end result should resemble:

```
import java.io.Serializable;

public class Customer implements Serializable, Entity<Long> {
```

3-1.6 Navigate to the **TestCustomer.java** class, right-click and choose **Run As > Java Application**. Was an error displayed in the Console window? It may have been similar to:

```
Caused by: java.io.NotSerializableException: retail.Address
. . .
at
com.tangosol.util.ExternalizableHelper.toBinary(ExternalizableHel
per.java:211)
... 5 more
```

Coherence requires that all members be serializable or implement the serializable interface.

3-1.7 Navigate to the **Address.java** class, and modify it to implement **Serializable**.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

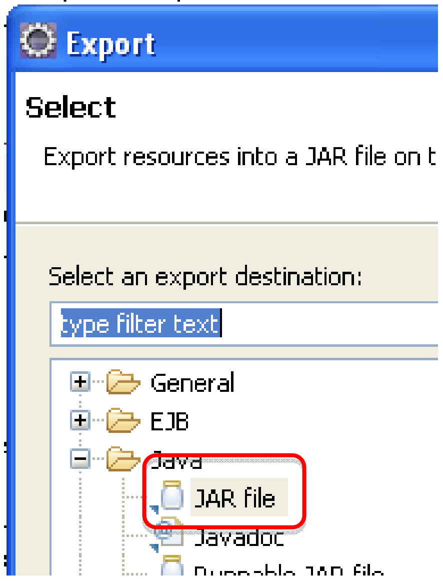
3-1.8 Return to **TestCustomer.java** and rerun the test. The console window should show output containing:

```
Returned: Customer:
    ID:1
    Name:Bill Iards
       Address:
              Street:8 Van de Graaf Drive
              City:Burlington
              State:MA
              Zip:01803
```

Export a JAR file

3-1.9 Select **File > Export** from the main menu.

3-1.10 If required, expand **Java** and choose **JAR file** and click **Next**.

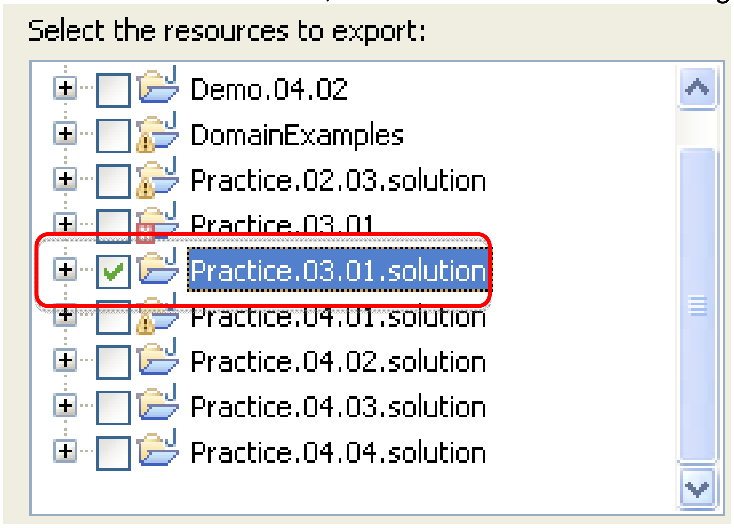


Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

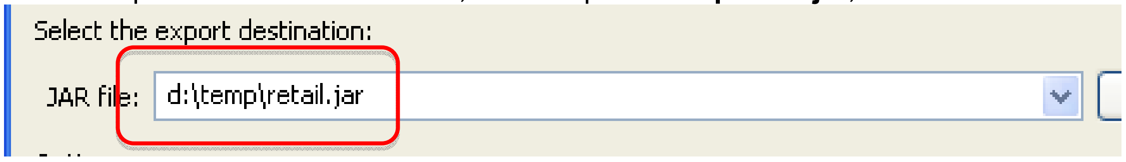
3-1.11 Select the **Practice.03.01**, or the **.solution** version if using the solution.



Select the resources to export:

- Demo.04.02
- DomainExamples
- Practice.02.03.solution
- Practice.03.01
- ✓ Practice.03.01.solution
- Practice.04.01.solution
- Practice.04.02.solution
- Practice.04.03.solution
- Practice.04.04.solution

3-1.12 Enter a export destination and name, for example **d:\temp\retail.jar**, and click Finish



Select the export destination:

JAR file: d:\temp\retail.jar

Test using an exported jar file and the console

3-1.13 Using the desktop shortcut, open a command prompt and change directory to the `D:\Oracle\coherence\bin` directory.

3-1.14 Modify the **coherence.cmd** and find the command containing:

```
-cp " %coherence_home%\li . . .
```

3-1.15 Modify the **classpath** to include the exported `.jar` file. The result should resemble:

```
. . . -cp "d:\temp\retail.jar;%coherence_home%\. . .
```

3-1.16 Save the result and exit the editor.

3-1.17 Run the script and connect to a cache using a command similar to:

```
Map (?): cache retail.customers
```

The command prompt should change to reflect the selected cache. For example:

```
Map (retail.customers):
```

3-1.18 `List` the contents of the cache, which should show nothing.

3-1.19 Return to Eclipse and rerun the **TestCustomer** class.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

3-1.20 Return to the Coherence console and rerun the `list` command.

The newly added customer should be displayed using its `toString` method and produce results similar to:

```
Map (retail.customers): list
1 = Customer:
        ID:1
        Name:Bill Iards
                Address:
                        Street:8 Van de Graaf Drive
                        City:Burlington
                        State:MA
                        Zip:01803



Map (retail.customers):
```

Hint: If you run into an error about mismatched cluster names, consider adding `-Dtangosol.coherence.cluster=test` to the run config and to the console script.

Solution Instructions

To run the solution, start Eclipse and connect to the **D:\Oracle\student\practices** workspace. Open the project **Practice.03.01.solution** and follow the directions for testing the application from steps 3-1.9.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 3-2: Serialization using `ExternalizableLite`

Skills Learned

At the end of this practice, you should be able to:

- Modify a Serializable class to use ExternalizableLite
- Externalize parent and child classes
- Test an ExternalizableLite class.

Problem Statement

From:	Tom.Foolery@retail-is-us.com
Sent:	Thursday, July 1st, 11:38am
To:	you@acmedev.com
Subject:	Serialization improvements

Good morning,

I heard from Bill about your Coherence example. Nice work, but I understand there are a number of ways to decrease the size of the serialized output.

Can you update your example to use a smaller form of serialization?
I understand that `ExternalizableLite` can be used rather the core serialization if the classes meet certain requirements such as:

1. Argumentless constructor.
2. Implements the `ExternalizableLite` interface

At this point we don't need cross-platform compatibility, so this should be sufficient!

Regards,

Tom Foolery
Retail Is Us
Tom.Foolery@retail-is-us.com

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Overview Instructions

1	Start Eclipse and open the provided project.
2	Test the existing serializable sources.
3	Modify the Customer and Address classes to support `ExternalizableLite`.
4	Test the updated classes.

Hints

- String ExternalizableHelper.readSafeUTF(DataInput in) can be used to read a Java String. Conversely ExternalizableHelper.writeSafeUTF(DataOutput out, String value) can be used to write a Java string.
- ExternalizableHelper has methods for readLong, writeLong as well as methods for all primitive types and their object counterparts.
- ExternalizableHelper.readObject can be used to read an entire object's contents, but the result must be cast appropriately. There is also an equivalent writeObject method for storing an entire object which implements ExternalizableLite.

Detailed Instructions

Start Eclipse and open the provided project

3-2.1 Using the desktop shortcut, start Eclipse.

3-2.2 Open the **Practice.03.02** project
If required, import the project by:

A. Selecting **File > Import**
B. Expand the **General** directory and choose **Existing Projects into Workspace**
C. Enter the root directory of the Workspace **D:\Oracle\student\practices** and select **Practice 03.02**
D. Click **Finish**

Test the existing serializable sources

3-2.3 Navigate to the **retail.test** package and open **TestCustomer**.

3-2.4 Anywhere in the editor window, right-click and choose **Run As -> Java Application.**

3-2.5 Examine the console output. The test client was updated to capture the serialized result and output the binary values and its length using code similar to:

```
import com.tangosol.util.ExternalizableHelper;
import com.tangosol.util.Base;

Binary binValue = ExternalizableHelper.toBinary(customer);
System.out.println("\n\nBinary value ("+binValue.length()+"):" +
Base.toHexDump(binValue.toByteArray(), 16) +"\n\n");
```

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

The output should resemble:

```
Original: Customer:
   ID:1
. . .
Binary value (339):0000:  0B AC ED 00 05 73 72 00 0F 72 65. . .
```

Note: You may need to add **`-Dtangosol.coherence.cluster=test`** to the run configuration in the event of a cluster name mismatch error.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Modify the Customer and Address classes to support `ExternalizableLite`	
3-2.6	Navigate to the **retail** package and open the **Address** class.
3-2.7	Add a no-argument constructor to the Address class. The constructor should resemble: `public Address() { }`
3-2.8	Modify the class definition to implement `ExternalizableLite` rather then `Serializable`. The updated class should resemble: `public class Address implements` ***`ExternalizableLite`***

3-2.9	Using quick fix or adding the methods by hand, add the required imports.
	
	The new imports should resemble:
	
	`import java.io.DataInput;`
	`import java.io.DataOutput;`
	`import java.io.IOException;`
	
	`import com.tangosol.io.ExternalizableLite;`
	`import com.tangosol.util.ExternalizableHelper;`

3-2.10	Using quick fix or by hand, add the required **readExternal** and **writeExternal** methods. The Unimplemented method should resemble:
	
	`@Override`
	`public void readExternal(DataInput in) throws IOException {`
	`    // TODO Auto-generated method stub`
	`}`
	`@Override`
	`public void writeExternal(DataOutput out) throws IOException {`
	`    // TODO Auto-generated method stub`
	`    }`

3-2.11	Complete the `readExternal` method to use `ExternalizableHelper.readSafeUTF` to read all four class variables. The resulting method should resemble:
	
	`public void readExternal(DataInput in) throws IOException {`
	`    street = ExternalizableHelper.readSafeUTF(in);`
	`    city = ExternalizableHelper.readSafeUTF(in);`
	`    state = ExternalizableHelper.readSafeUTF(in);`
	`    zip = ExternalizableHelper.readSafeUTF(in);`
	`}`

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

3-2.12 Complete the `writeExternal` method to use `ExternalizableHelper.writeSafeUTF` to wtoreall four class variables. The resulting method should resemble:

```
public void writeExternal(DataOutput out) throws IOException {
    ExternalizableHelper.writeSafeUTF(out, street);
    ExternalizableHelper.writeSafeUTF(out, city);
    ExternalizableHelper.writeSafeUTF(out, state);
    ExternalizableHelper.writeSafeUTF(out, zip);
}
```

3-2.13 Save the changes.

3-2.14 Open the `Customer` class and add a no argument constructor.

3-2.15 Modify the class to implement `ExternalizableLite` rather the `Serializable`.

3-2.16 Use quick fix to add the required imports.

3-2.17 Use quick fix to add the required `readExternal` and `writeExternal` shell methods.

3-2.18 Complete the implementation of the `readExternal` method, which should resemble:

```
public void readExternal(DataInput in) throws IOException {
    id = new Long(in.readLong());
    name = ExternalizableHelper.readSafeUTF(in);
    address = (Address) ExternalizableHelper.readObject(in);
}
```

3-2.19 Complete the implementation of the `writeExternal` method, which should resemble:

```
public void writeExternal(DataOutput out) throws IOException {
    out.writeLong(id.longValue());
    ExternalizableHelper.writeSafeUTF(out, name);
    ExternalizableHelper.writeObject(out, address);
}
```

3-2.20 Save the changes.

Test the updated classes

3-2.21 Return to the **TestCustomer** class in the **retail.tests** package. Rerun the test.

3-2.22 Examine the resulting output in the **Console** tab, what size was the resulting serialized output? Was it smaller? By roughly how much? Note the size for comparison with the next practice.

Solution Instructions

To run the solution, start Eclipse and connect to the **D:\Oracle\student\practices** workspace and open **Practice.03.02.solution**. Follow the directions for testing the application from steps 3-2.21.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 3-3: Serializing using Portable Object Format

Skills Learned

At the end of this practice, you should be able to:

- Convert an object to support `PortableObject`
- Add required serialization methods, `readExternal` and `writeExternal`
- Modify a test class to create and register a `SimplePofContext`
- Test using `ExternalizableHelper`

Problem Statement

From:	Tom.Foolery@retail-is-us.com
Sent:	Tuesday, July 6th, 1:04pm
To:	you@acmedev.com
Subject:	Portable Objects

Good morning,

I understand that Portable Object allows us to use our objects between Java and other languages, such as .NET.

Can you update your example to use this form of serialization?
I understand that all that is involved is to:

1. Add support for Portable Object along with replacement read and write methods
2. Test using a `SimplePofContext`

If you could add that example, it would be great!

Regards,

Tom Foolery
Retail Is Us
Tom.Foolery@retail-is-us.com

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources\`**	Contains a set of practice specific resource directories

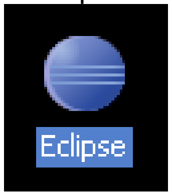
Overview Instructions

1	Start Eclipse and select the practices workspace.
2	Modify classes to support `PortableObject`.
3	Modify Test class to use `SimplePofContext` and `ExternalizableHelper`.
4	Test.

Detailed Instructions

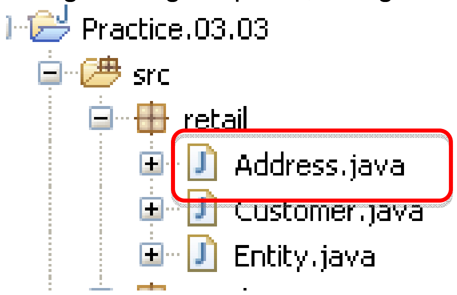
Start Eclipse and select the practices workspace

3-3.1 If not already running, start **Eclipse** and select the **D:\Oracle\student\practices** workspace.



Modify Objects to support `PortableObject`

3-3.2 Using Package explorer, Navigate to the **Address** class in the **retail** package and open it.



3-3.3 In the editor window modify the class definition so that **Address** implements **PortableObject**.

Your code should resemble:

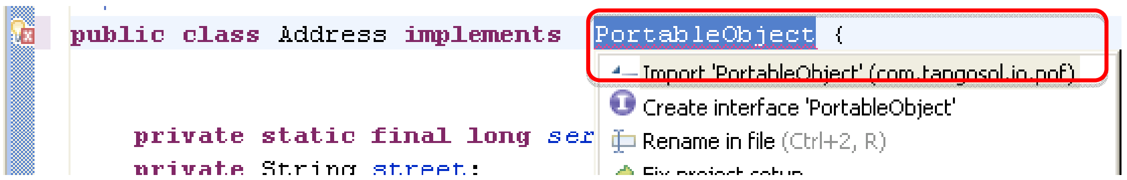
```
public class Address implements PortableObject
```

3-3.4 Using quick fix or by hand, add the import **com.tangosol.io.pof.PortableObject**.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

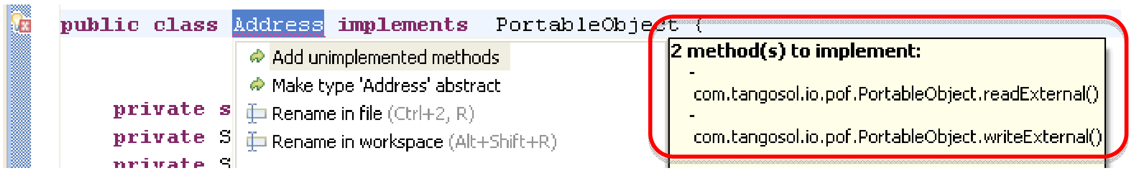


```
public class Address implements PortableObject {
    private static final long ser
    private String street;
```

Import 'PortableObject' (com.tangosol.io.pof)
Create interface 'PortableObject'
Rename in file (Ctrl+2, R)
Fix project setup

3-3.5 Using quick fix or by hand, add the required **readExternal** and **writeExternal** methods.

Remember to remove the previous versions from the implementation of `ExternalizableLite`.



```
public class Address implements PortableObject {
    private s
    private S
    private S
```

Add unimplemented methods
Make type 'Address' abstract
Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)

2 method(s) to implement:
- com.tangosol.io.pof.PortableObject.readExternal()
- com.tangosol.io.pof.PortableObject.writeExternal()

3-3.6 Implement the `readExternal` method to use the `PofReader` and the `readString` method to read each variable.

The completed code should resemble:

```
public void readExternal(PofReader in) throws IOException {
    street = in.readString(0);
    city = in.readString(1);
    state = in.readString(2);
    zip = in.readString(3);
}
```

.

3-3.7 Implement the `writeExternal` method to use the `PofWriter` and the `writeString` method to write each variable.

The completed code should resemble:

```
public void writeExternal(PofWriter out) throws IOException {
    out.writeString(0,street);
    out.writeString(1,city);
    out.writeString(2,state);
    out.writeString(3,zip);
}
```

Note: A better implementation of the read and write methods would use constants rather than embedding actual values into the code.

3-3.8 Save the changes to the **Address** object.

3-3.9 Open the Customer object and replace `ExternalizableLite` with `PortableObject`

3-3.10 Using quick fix or by hand, add the import **com.tangosol.io.pof.PortableObject**.

3-3.11 Using quick fix or by hand, add the required **readExternal** and **writeExternal** methods. **Note:** Remember to remove the **ExternalizableLite** versions of these methods.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

3-3.12 Implement the `readExternal` method to use the `PofReader` and the `readLong and readObject` methods to read each variable and the child **Address** class.

The completed code should resemble:

```
public void readExternal(PofReader in) throws IOException {
    id = in.readLong(1);
    name = in.readString(2);
    address = (Address)in.readObject(3);
}
```

Note that the index used must increase from the value used in the **Address** class.

3-3.13 Implement the `writeExternal` method to use the `PofWriter` and the `writeLong`, `writeString` and `writeObject` methods to write each variable.

The completed code should resemble:

```
public void writeExternal(PofWriter out) throws IOException {
    out.writeLong(1, id.longValue());
    out.writeString(2,name);
    out.writeObject(3, address);
}
```

3-3.14 Save your changes to the **Customer** object.

Modify Test class to use SimplePofContext and ExternalizableHelper

3-3.15 Navigate to the **retail.test** package and open the **TestCustomer** class.

3-3.16 Remove all references to the **NamedCache cache**, as they are not required.

3-3.17 Add a reference to **SimplePofContext**, along with its required import **com.tangosol.io.pof.SimplePofContext**.

The code should resemble:

```
public static void main(String[] args) {
     SimplePofContext ctx = new SimplePofContext();
```

3-3.18 Register two portable object serializers: one **Customer** and **Address** respectively.

The code should resemble:

```
ctx.registerUserType(1000, Customer.class ,
                     new PortableObjectSerializer(1000));
ctx.registerUserType(1001, Address.class ,
                     new PortableObjectSerializer(1001));
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

3-3.19	Add the required **context** to the `toBinary` method call. The updated code should resemble: `Binary binary   = ExternalizableHelper.toBinary(original, ctx);`

3-3.20	Modify the remaining code to return a copy from the binary data using the `fromBinary` method.
	
	The updated code should resemble:
	
	`Object copy  = ExternalizableHelper.fromBinary(binary, ctx);`
	`System.out.println("Returned: " + copy);`
	`System.out.println("Copy: " + original);`

Test	
3-3.21	Right-click anywhere in the **TestCustomer** class, and choose **Run As > Java Application**.
3-3.22	Examine the results in the Console tab. Note the size of the binary object. How does it compare with the prior practice?

Solution Instructions

To run the solution, start Eclipse and connect to the **D:\Oracle\student\practices** workspace and open **Practice.03.03.solution**. Follow the directions for testing the application from steps 3-3.15, ignoring any steps to add code, all of which is in the solution.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 4

Chapter 5

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Practices for Lesson 4

Practices Overview

Developers must be able to configure the various different caching topologies offered by Coherence.

The objective of this practice is to assist the developer with configuring Coherence caches.

In this practice, you will perform the following tasks:

- Configure, run, and review a local cache topology.
- Configure, run, and review a replicated cache topology.
- Configure, run, and review a partitioned cache topology.
- Configure, run, and review a near cache topology.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 4-1: Configure, Run, and Review a Local Cache Topology

Skills Learned

At the end of this practice, you should be able to:

- Specify command line arguments to the Coherence server start script to override the default `coherence-cache-config.xml` file.
- Configure a local cache scheme and map it to a cache.
- Start a Coherence two-node cluster that uses the local cache.
- Identify that Coherence is using the specific configuration.
- Review and explain where data is stored, and why it proves a local cache is in use.

Problem Statement

From:	Billi.Onair@retail-is-us.com
Sent:	Thursday, July 7th, 2:12pm
To:	you@acmedev.com
Subject:	Configuring a Coherence Local Cache

Hello!

Our application is experiencing performance problems because our database is overloaded. We would like to alleviate the load by caching the data locally to avoid using the database when the data is frequently reused.

As we understand it, caching the data locally makes sense for applications that are read-only or read-mostly, can tolerate slightly stale data, and the data size is small.

Our engineers say that Coherence can provide this caching capability for our application, and that you have the skills to configure Coherence caches. We look forward to seeing an example of a Coherence local cache configuration, and understanding how to tell that our configuration is being used.

Regards,

Billi Onair
Retail Is Us
Billi.Onair@retail-is-us.com

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Design

In this practice you will add a local cache configuration to a `coherence-cache-config.xml` file. This file is used by Coherence to construct the various caches that the node will use. You will use the `<local-scheme>` element to configure the local cache, and the `<cache-mapping>` element to map all caches to your local scheme. The diagram below shows two Coherence nodes, called CacheTester, each with a local cache configured. Note that each local cache is shaded or colored differently to represent that each may contain different data.

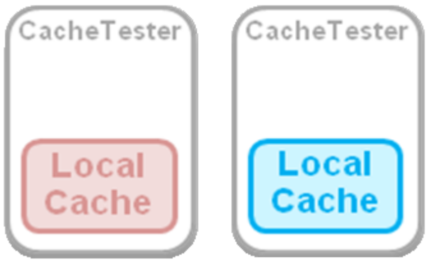


Figure 1 Local Cache Cluster

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources`**`\`	Contains a set of practice specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Coherence Cache Override (`-Dtangosol.coherence.cacheconfig`) Specifies which configuration file to use for Coherence caching	
Not Set	By default, Coherence uses the `coherence-cache-config.xml` file bundled in `%COHERENCE_HOME%\lib\coherence.jar`.
`filename`	Specifies the filename of the configuration file to use

Table 4.1

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Configure a local cache and run in a Coherence cluster.
2	Analyze local cache execution results.

Hints:

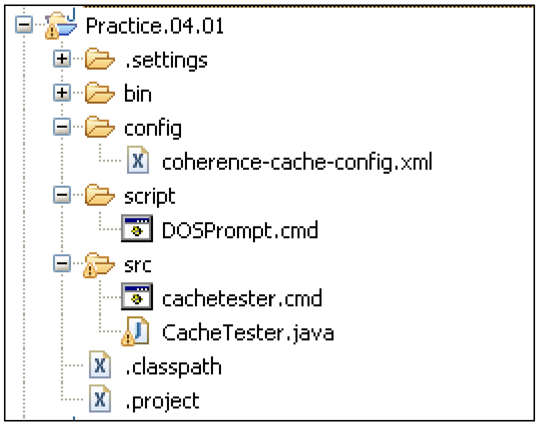
- A local cache is configured using the `<local-scheme>` element within the `<caching-schemes>` element.
- A cache is mapped to a particular scheme using the `<cache-name>` and `<scheme-name>` elements within the `<cache-mapping>` element.
- You specify to Coherence which cache configuration file to use by setting the system property `-Dtangosol.coherence.cacheconfig=`*`your_file_name.xml`*.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to make. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Detailed Instructions

Configure a local cache and run in a Coherence cluster

4-1.1 Within Eclipse, expand the **Practice.04.01** project and expand the **config**, **script**, and **src** folders to display their contents.



4-1.2 Open the `coherence-cache-config.xml` file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

4-1.3	Locate the text, “TODO: Place local cache scheme here” within the `<caching-schemes>` element, and replace with the code for a Coherence local caching scheme. **Note:** For convenience, the code snippets for this exercise can be found in the resources subdirectory for **Practice.04.01** in the **code.snippet.txt** file. Your code should resemble:
	
	`<local-scheme>`
	`  <scheme-name>MyLocalCachingScheme</scheme-name>`
	`</local-scheme>`

4-1.4	Locate the text, “TODO: Place cache-mapping here” within the `<caching-scheme-mapping>` element, and replace with the code to map a cache of any name to the local cache scheme defined in the last step.
	Your code should resemble:
	
	`<cache-mapping>`
	`  <cache-name>*</cache-name>`
	`  <scheme-name>MyLocalCachingScheme</scheme-name>`
	`</cache-mapping>`
	
	Save your changes.

4-1.5	Within the **src** folder, open the **`cachetester.cmd`** file for editing. Add the system property to the command line that instructs Coherence to use your `coherence-cache-config.xml` file instead of the default. Your code should resemble: `set java_opts="-Xms%memory% -Xmx%memory% -Dtangosol.coherence.ttl=0 -`**`Dtangosol.coherence.cacheconfig=D:\Oracle\student\practices\Practice.04.01\config\coherence-cache-config.xml`**`"` Save your changes. Now that the configuration is done, you are ready to run your cluster.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-1.6 Within the **src** folder, open the **CacheTester.java** file and examine its code while keeping the following in mind:

CacheTester.java accepts two arguments:

- ***-s startIndex*** (default is 0)
- ***-f finishIndex*** (default is 10)

The program creates a `NamedCache` called `Customers`, and uses the entered indexes to load some test data into the cache. It uses the text "customer" plus the index to create the keys for the data. For example, for index 0, the key will be customer0. Likewise, a similar tactic is used for the data that is placed in the cache, using the text "`Mark`" combined with the index. For example, for index 0, the data will be Mark0. The program will loop from the `-s` index value to the `-f` index value, and load the local cache with data.

This lab uses two cluster nodes to demonstrate how a local cache works across multiple nodes. Both nodes must load the cache with different data, so the first node will load the cache with indexes 0-9, and the second node will load the cache with indexes 10-19.

After the caches are loaded, the program sleeps for 15 seconds to ensure that the two nodes have formed a cluster and properly loaded their caches. When the program stops sleeping, it tells the NamedCache to return all of its cached entries and prints them out on the console screen.

The next few steps will instruct you how to run this scenario.

4-1.7 From within Eclipse:

- Make sure the **Practice.04.01** project is selected.
- Double-click the **DOSPrompt.cmd** file in the **script** folder to open a DOS command shell.

4-1.8 Repeat the previous step to open a second command window.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-1.9 In both command windows type in `cachetester.cmd` to prepare them to run the cluster. **DO NOT EXECUTE THE SCRIPTS YET!**
In the first command window, the defaults are used, so no command line arguments are needed. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd
```

In the second command window, set the index range to use by specifying the arguments: **`-s 10 -f 20`**. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd -s 10 -f 20
```

When ready, quickly execute the command in both command windows, and wait for them to exit. You should see output similar to:

Command Window 1:

```
Oracle Coherence Version 3.6.0.0 DPR2 Build 14713
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

2010-06-24 15:14:51.531/0.516 Oracle Coherence GE 3.6.0.0 DPR2 <Info>
(thread=main, member=n/a): Loaded cache configuration from
"file:/D:/Oracle/student/practices/Practice.04.01/config/coherence-
cache-config.xml"
Loading data in cache: 0-10
Pausing for 15 seconds...
customer0 = Mark 0
customer1 = Mark 1
customer2 = Mark 2
customer3 = Mark 3
customer4 = Mark 4
customer5 = Mark 5
customer6 = Mark 6
customer7 = Mark 7
customer8 = Mark 8
customer9 = Mark 9
```

Command Window 2:

```
Oracle Coherence Version 3.6.0.0 DPR2 Build 14713
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

2010-06-24 15:14:52.390/0.531 Oracle Coherence GE 3.6.0.0 DPR2 <Info>
(thread=main, member=n/a): Loaded cache configuration from
"file:/D:/Oracle/student/practices/Practice.04.01/config/coherence-
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

```
cache-config.xml"
Loading data in cache: 10-20
Pausing for 15 seconds...
customer10 = Mark 10
customer11 = Mark 11
customer12 = Mark 12
customer13 = Mark 13
customer14 = Mark 14
customer15 = Mark 15
customer16 = Mark 16
customer17 = Mark 17
customer18 = Mark 18
customer19 = Mark 19
```

You have successfully configured a local cache and ran it in a Coherence cluster. Are you sure? How do you know?
The next section explores these questions with you.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze local cache execution results	
4-1.10	Take a closer look at the output in each command window. The first thing you should find in each window is a line similar to the following: `2010-06-24 15:14:52.390/0.531 Oracle Coherence GE 3.6.0.0 DPR2 <Info> (thread=main, member=n/a): Loaded cache configuration from` **`"file:/D:/Oracle/student/practices/Practice.04.01/config/coherence-cache-config.xml"`** If either of them does not have a line pointing to your `coherence-cache-configuration.xml` file, then you may want to check the `-D` setting you made in the `cachetester.cmd` script. Otherwise, congratulations, Coherence used your configuration file. Now let's see what else you can figure out from the console output.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

4-1.11 The next thing you might notice while looking at both console windows is that command window 1 and command window 2 report having different data in their caches:

Command Window 1:

```
customer0 = Mark 0
customer1 = Mark 1
customer2 = Mark 2
customer3 = Mark 3
customer4 = Mark 4
customer5 = Mark 5
customer6 = Mark 6
customer7 = Mark 7
customer8 = Mark 8
customer9 = Mark 9
```

Command Window 2:

```
customer10 = Mark 10
customer11 = Mark 11
customer12 = Mark 12
customer13 = Mark 13
customer14 = Mark 14
customer15 = Mark 15
customer16 = Mark 16
customer17 = Mark 17
customer18 = Mark 18
customer19 = Mark 19
```

Keep in mind that command window 1 ran the program with indexes 0-9, and command window 2 ran the program with indexes 10-19. What does this tell you about how a local cache works in Coherence? Why? The answer is because a local cache in a Coherence node is completely unaware of the data in a cache on another node. Each cache stores its data locally, so each cache, or node, has a completely different view of data.

Now that you have run a Coherence local cache, and verified its behavior, let's take a look at working with replicated caches.

Note: Close all command windows before proceeding.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace and open **Practice.04.01.solution**. Follow the directions for testing the application from steps **4-1.6**, ignoring any steps to add code, all of which is in the solution.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 4-2: Configure, Run, and Review a Replicated Cache Topology

Skills Learned

At the end of this practice, you should be able to:

- Specify command line arguments to the Coherence server start script to override the default `coherence-cache-config.xml` file.
- Configure a replicated cache scheme and map it to a cache.
- Start a Coherence two-node cluster that uses the replicated cache.
- Identify that Coherence is using the specific configuration.
- Identify that Coherence starts the services related to the configuration.
- Review and explain where data is stored, and why it proves a replicated cache is in use.

Problem Statement

From:	Tom.Terrific@retail-is-us.com
Sent:	Thursday, July 7th, 4:02pm
To:	you@acmedev.com
Subject:	Configuring a Coherence Replicated Cache

Hello!

We have been using the Coherence local cache configuration you provided us earlier. We have found that the data stored in a local cache is not consistent across cache instances. We would like data stored in the cache to be consistent across every instance of the cache using a Coherence replicated cache, which provides this capability.

As we understand it, the replicated cache is good for keeping data in-sync across every cache instance while providing high-performance cache reads. However, because the data in a replicated cache is replicated to every cache instance in the cluster, it consumes a lot of memory quickly, limits the overall data size to the size of a single JVM, and scales poorly for cache writes when the number of caching nodes increases.

Our engineers say that you can easily modify our cache configuration to change our local cache scheme to a replicated cache scheme. We look forward to seeing an example of a Coherence replicated cache configuration, and understanding how to tell our configuration is being used.

Regards,

Tom Terrific
Retail Is Us
Tom.Terrific@retail-is-us.com

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Design

In this practice you will add a replicated cache configuration to a `coherence-cache-config.xml` file. This file is used by Coherence to construct the various caches that the node will use. You will use the `<replicated-scheme>` element to configure the replicated cache, and the `<cache-mapping>` element to map all caches to your replicated scheme. The diagram below shows two Coherence nodes, called CacheTester, each running with the replicated cache configured.

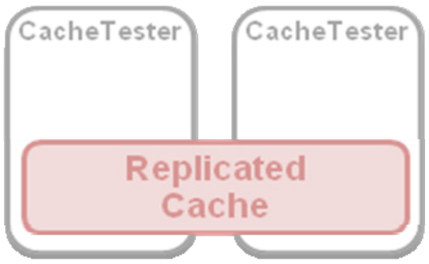


Figure 2 Replicated Cache Cluster

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources`**`\`	Contains a set of practice specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Coherence Cache Override (`-Dtangosol.coherence.cacheconfig`**)** Specifies which configuration file to use for Coherence caching	
Not Set	By default, Coherence uses the `coherence-cache-config.xml` file bundled in `%COHERENCE_HOME%\lib\coherence.jar`.
`filename`	Specifies the filename of the configuration file to use

Table 4.2

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Configure a replicated cache and run in a Coherence cluster.
2	Analyze replicated cache execution results.

Hints:

- A replicated cache is configured using the `<replicated-scheme>` element within the `<caching-schemes>` element.
- A cache is mapped to a particular scheme using the `<cache-name>` and `<scheme-name>` elements within the `<cache-mapping>` element.
- A local cache is configured as a backing map for a replicated cache. Use the `<scheme-ref>` element to reuse a local cache configuration as the backing map for the replicated cache.
- You specify to Coherence which cache configuration file to use by setting the system property `-Dtangosol.coherence.cacheconfig=`*`your_file_name.xml`*.

Detailed Instructions

Configure a replicated cache and run in a Coherence cluster	
4-2.1	Within Eclipse, expand the **Practice.04.02** project and expand the **config**, **script**, and **src** folders to display their contents.
4-2.2	Open the `coherence-cache-config.xml` file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-2.3 Locate the text, "TODO: Place replicated cache scheme here" within the `<caching-schemes>` element, and replace with the code for a Coherence replicated caching scheme. **Note:** For convenience, the code snippets for this exercise can be found in the resources subdirectory for **Practice.04.02** in the **code.snippet.txt** file. Your code should resemble:

```
<replicated-scheme>
  <scheme-name>MyReplicatedScheme</scheme-name>
  <service-name>ReplicatedCache</service-name>

  <backing-map-scheme>
    <local-scheme>
      <scheme-ref>MyLocalCachingScheme</scheme-ref>
    </local-scheme>
  </backing-map-scheme>

  <autostart>true</autostart>
</replicated-scheme>
```

4-2.4 Locate the text, "TODO: Place cache-mapping here" within the `<caching-scheme-mapping>` element, and replace with the code to map a cache of any name to the replicated cache scheme defined in the last step. Your code should resemble:

```
<cache-mapping>
  <cache-name>*</cache-name>
  <scheme-name>MyReplicatedScheme</scheme-name>
</cache-mapping>
```

Save your changes.

4-2.5 Within the **src** folder, open the **`cachetester.cmd`** file for editing. Add the system property to the command line that instructs Coherence to use your `coherence-cache-config.xml` file instead of the default.
Save your changes. Now that the configuration is done, you are ready to run your cluster.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-2.6	**Note:** This version of the CacheTester program works a little differently than the last version, so make sure to read through this description. Within the **src** folder, open the **`CacheTester.java`** file and examine its code while keeping the following in mind: The program creates a `NamedCache` called `Customers`, and uses the entered indexes to load some test data into the cache. It uses the `Customer` and `Address` objects from the retail object model found in the `DomainExamples` Eclipse project. The index serves as the `ID` of the `Customer` objects, as well as the key for storing the objects into the cache. The data that comprises the `Customer` and `Address` objects is hardcoded in the **`CacheTester.java`** class, and built using the index counter. The program will loop from the `-s` index value to the `-f` index value, and load the replicated cache with data. This lab uses two cluster nodes to demonstrate how a replicated cache works across multiple nodes. Both nodes must load the cache with different data, so the first node will load the cache with indexes 0-4, and the second node will load the cache with indexes 5-9. After the caches are loaded, the program sleeps for 15 seconds to ensure that the two nodes have formed a cluster and properly loaded their caches. When the program running in command window 1 (or whichever program runs with a start parameter of 0) stops sleeping, it tells the `NamedCache` to execute some code on every customer entry in the cache to print out a “**Here I Am**” message which displays on the command window where the cache entry exists. This “**Here I Am**” message is invoked as part of the `HereIAmProcessor.java` file. This reveals how a replicated cache stores and retrieves its cache entries. The next few steps will instruct you how to run this scenario.
4-2.7	From within Eclipse, open two DOS prompt command windows for the **Practice.04.02** project.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-2.8 In both command windows type in `cachetester.cmd` to prepare them to run the cluster. **DO NOT EXECUTE THE SCRIPTS YET!**

In the first command window, the defaults are used so no command line arguments are needed. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd
```

In the second command window, set the index range to use by specifying the arguments: **`-s 5 -f 10`**. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd -s 5 -f 10
```

When ready, quickly execute the command in both command windows, and wait for them to exit. You should see output similar to:

Command Window 1:

```
Loading data in cache: 0-5
Pausing for 15 seconds...
2010-07-23 10:34:19.187/4.469 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member(Id=2, Timestamp=2010-07-23
10:34:19.187, Address=139.185.35.126:8089, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:2524,
Role=CacheTesterCacheTester) joined Cluster with senior member 1
2010-07-23 10:34:19.312/4.594 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member 2 joined Service Management with
senior member 1
2010-07-23 10:34:19.453/4.735 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member 2 joined Service ReplicatedCache
with senior member 1
2010-07-23 10:34:20.078/5.391 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=TcpRingListener, member=1): Connected to Member(Id=2,
Timestamp=2010-07-23 10:34:19.187, Address=139.185.35.126:8089,
MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:2524,
Role=CacheTesterCacheTester)
Here I am: Customer:
      ID:1
      Name:James Jones
      Address:    Address:
            Street:38 Clark Place
            City:San Francisco
            State:CA
            Zip:94121

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:2
      Name:Scott Riley
      Address:    Address:
            Street:44 Pumpkin Ct
            City:Shirley
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

```
            State:MA
            Zip:01464

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:3
      Name:Johnny Greif
      Address:    Address:
            Street:76 Nile Ct
            City:Arvata
            State:CO
            Zip:80001

      Credit Card IDs:[]
            Order IDs:[]
      . . .
```

Command Window 2:

```
Loading data in cache: 5-10
Pausing for 15 seconds...
2010-07-23 10:34:20.109/3.250 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=TcpRingListener, member=2): Connected to Member(Id=1,
Timestamp=2010-07-23 10:34:15.687, Address=139.185.35.126:8088,
MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:3424,
Role=CacheTesterCacheTester)
2010-07-23 10:34:32.609/15.750 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=2): Member 1 left service Management with
senior member 2
2010-07-23 10:34:32.625/15.766 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=2): Member 1 left service ReplicatedCache
with senior member 2
2010-07-23 10:34:32.640/15.781 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=2): MemberLeft notification for Member 1
received from Member(Id=1, Timestamp=2010-07-23 10:34:15.687,
Address=139.185.35.126:8088, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:3424,
Role=CacheTesterCacheTester)
2010-07-23 10:34:32.640/15.781 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=2): Member(Id=1, Timestamp=2010-07-23
10:34:32.64, Address=139.185.35.126:8088, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:3424,
Role=CacheTesterCacheTester) left Cluster with senior member 2
2010-07-23 10:34:32.718/15.859 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=TcpRingListener, member=2): Disconnected from Member(Id=1,
Timestamp=2010-07-23 10:34:32.64, Address=139.185.35.126:8088,
MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:3424,
Role=CacheTesterCacheTester) to maintain ring
2010-07-23 10:34:34.578/17.719 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Invocation:Management, member=2): Service Management left the
cluster
2010-07-23 10:34:34.578/17.719 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=ReplicatedCache, member=2): Service ReplicatedCache left the
```

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

```
cluster
2010-07-23 10:34:34.656/17.797 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=2): Service Cluster left the cluster
```

You have successfully configured a replicated cache and ran it in a Coherence cluster. How do you know this time?

The next section explores these questions with you.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze local cache execution results	
4-2.9	Take a closer look at the output in each command window. The first thing you should find in each window is the line showing that your `coherence-cache-configuration.xml` file was loaded.
4-2.10	Next, you want to look for messages that have the word "ReplicatedCache" in them. This shows that the replicated cache service was started, and that there were two cluster members that joined the service. This is a positive indication that a replicated cache is configured and in use.
4-2.11	The next thing you might notice while looking at both console windows is that command window 1 and command window 2 have completely different application output. Command window 1 prints out all of the "**Here I Am**" messages, and command window 2 does not print anything. You may remember when learning how the programs run, that only the first command window was going to call the `NamedCache` method that causes the "**Here I Am**" message to get printed out by all entries in the cluster. What does this tell you about how a replicated cache works in Coherence? Why? The reason that command window 1 shows all the results is because a replicated cache contains all of the cache entries by default, so Coherence never has to leave the current node in order to find every entry. Although each cluster node only put half of the data into the cache, the replicated cache automatically replicated the data to the caches on both nodes. If you changed the `CacheTester.java` code to allow the `customers.invokeAll()` method to run on both command windows, then you would see all the output in both windows because the data exists on both nodes. Now that you have run a Coherence replicated cache, and verified its behavior, let's take a look at working with partitioned caches. **Note:** Close all command windows before proceeding.

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace and open **Practice.04.02.solution**. Follow the directions for testing the application from steps **4-2.6**, ignoring any steps to add code, all of which is in the solution.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 4-3: Configure, Run, and Review a Partitioned Cache Topology

Skills Learned

At the end of this practice, you should be able to:

- Specify command line arguments to the Coherence server start script to override the default `coherence-cache-config.xml` file.
- Configure a partitioned cache scheme and map it to a cache.
- Start a Coherence two-node cluster that uses the partitioned cache.
- Identify that Coherence is using the specific configuration.
- Identify that Coherence starts the services related to the configuration.
- Review and explain where data is stored, and why it proves a partitioned cache is in use.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Problem Statement

From:	Becky.Slick@retail-is-us.com
Sent:	Friday, July 8th, 1:00pm
To:	you@acmedev.com
Subject:	Configuring a Coherence Partitioned Cache

Hello!

We have been using the Coherence replicated cache configuration you provided us earlier. At first, this was working great for us, but as our business grew, our caching requirements grew. The replicated cache was able to handle the smaller data size, but now the size of our cached data is much larger than the size of a single JVM heap. We also noticed that as we added more Coherence nodes to handle the growth of our business, the writes to the cache were taking much longer.

We have found that a replicated cache is good for providing consistent data and high-performance read access to smaller data set sets in the 1-4 GB range, depending on the size of the JVM. If a replicated cache is backed by off-heap storage, then it can be argued that larger data sets could be supported. However, the real shortfall of the replicated cache occurs exponentially as the number of cache nodes and the need to write data to the cache increases. Because a replicated cache automatically replicates the data to every node in the cluster, each and every write to the cache results in ***1 * # of nodes in cluster*** writes to caches across the cluster. This makes it very difficult, to impossible, to scale the replicated cache to support extremely large and write-intensive data sets. We would like to support our growing data set size and improve the write scalability of our cache by using a Coherence partitioned cache, which provides this capability.

As we understand it, the partitioned cache also keeps the data consistent across the cluster, but instead of replicating the data to every node in the cluster it stores a primary and a backup copy of the data within the cluster. This preserves memory on a per JVM basis and allows for highly predictive scalable reads and writes to extremely large data sets, with a small tradeoff in read performance.

Our engineers say that you can easily modify our cache configuration to change our replicated cache scheme to a partitioned cache scheme. We look forward to seeing an example of a Coherence partitioned cache configuration, and understanding how to tell if our configuration is being used.

Regards,

Becky Slick
Retail Is Us
Becky.Slick@retail-is-us.com

Design

In this practice you will add a partitioned cache configuration to a `coherence-cache-config.xml` file. This file is used by Coherence to construct the various caches that the node

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

will use. You will use the `<distributed-scheme>` element to configure the partitioned cache, and the `<cache-mapping>` element to map all caches to your partitioned scheme. The diagram below shows two Coherence nodes, called CacheTester, each running with the partitioned cache configured.

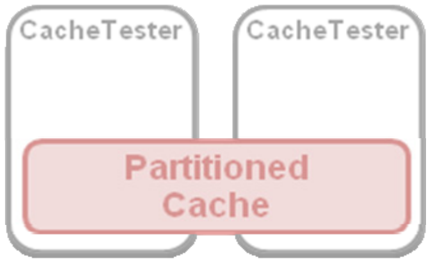


Figure 3 Partitioned Cache Cluster

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources\`**	Contains a set of practice specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Coherence Cache Override (`-Dtangosol.coherence.cacheconfig`) Specifies which configuration file to use for Coherence caching	
Not Set	By default, Coherence uses the `coherence-cache-config.xml` file bundled in `%COHERENCE_HOME%\lib\coherence.jar`.
`filename`	Specifies the filename of the configuration file to use

Table 4.3

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Configure a partitioned cache and run in a Coherence cluster.
2	Analyze partitioned cache execution results.

Hints:

- A partitioned cache is configured using the `<distributed-scheme>` element within the `<caching-schemes>` element.
- A cache is mapped to a particular scheme using the `<cache-name>` and `<scheme-name>` elements within the `<cache-mapping>` element.
- A local cache is configured as a backing map for a partitioned cache. Use the <scheme-ref> element to reuse a local cache configuration as the backing map for the partitioned cache.
- You specify to Coherence which cache configuration file to use by setting the system property `-Dtangosol.coherence.cacheconfig=`*`your_file_name.xml`*.

Detailed Instructions

Configure a partitioned cache and run in a Coherence cluster	
4-3.1	Within Eclipse, expand the **Practice.04.03** project and expand the **config**, **script**, and **src** folders to display their contents.
4-3.2	Open the `coherence-cache-config.xml` file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-3.3	Locate the text, "TODO: Place partitioned cache scheme here" within the `<caching-schemes>` element, and replace with the code for a Coherence partitioned caching scheme. **Note:** For convenience the code snippets for this exercise can be found in the resources subdirectory for **Practice.04.03** in the **code.snippet.txt** file. Your code should resemble:

```
<distributed-scheme>
  <scheme-name>MyPartitionedScheme</scheme-name>
  <service-name>DistributedCache</service-name>

  <backing-map-scheme>
    <local-scheme>
      <scheme-ref>MyLocalCachingScheme</scheme-ref>
    </local-scheme>
  </backing-map-scheme>

  <autostart>true</autostart>
</distributed-scheme>
```

4-3.4	Locate the text, "TODO: Place cache-mapping here" within the `<caching-scheme-mapping>` element, and replace with the code to map a cache of any name to the partitioned cache scheme defined in the last step. Your code should resemble:

```
<cache-mapping>
  <cache-name>*</cache-name>
  <scheme-name>MyPartitionedScheme</scheme-name>
</cache-mapping>
```

Save your changes.

4-3.5	Within the **src** folder, open the **`cachetester.cmd`** file for editing. Add the system property to the command line that instructs Coherence to use your `coherence-cache-config.xml` file instead of the default. Save your changes. Now that the configuration is done, you are ready to run your cluster.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

4-3.6	**Note:** This version of the CacheTester program works exactly the same as in the replicated cache lab, so the text is not repeated here. The general concept is that the cluster members will call the "**Here I Am**" processor, revealing where the cache entries are located in the cluster. This lab reveals how a partitioned cache stores and retrieves its cache entries. The next few steps will instruct you how to run this scenario.
4-3.7	From within Eclipse, open two DOS prompt command windows for the **Practice.04.03** project.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-3.8 In both command windows type in `cachetester.cmd` to prepare them to run the cluster. **DO NOT EXECUTE THE SCRIPTS YET!**

In the first command window, the defaults are used, so no command line arguments are needed. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd
```

In the second command window, set the index range to use by specifying the arguments: **`-s 5 -f 10`**. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd -s 5 -f 10
```

When ready, quickly execute the command in both command windows, and wait for them to exit. You should see output similar to:

Command Window 1:

```
Loading data in cache: 0-5
Pausing for 15 seconds...
2010-07-23 10:17:14.515/4.250 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member(Id=2, Timestamp=2010-07-23
10:17:14.515, Address=139.185.35.126:8089, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:1048,
Role=CacheTesterCacheTester) joined Cluster with senior member 1
2010-07-23 10:17:14.578/4.313 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=TcpRingListener, member=1): Connected to Member(Id=2,
Timestamp=2010-07-23 10:17:14.515, Address=139.185.35.126:8089,
MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:1048,
Role=CacheTesterCacheTester)
2010-07-23 10:17:14.656/4.391 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member 2 joined Service Management with
senior member 1
2010-07-23 10:17:14.953/4.688 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member 2 joined Service DistributedCache
with senior member 1
2010-07-23 10:17:14.984/4.719 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=DistributedCache, member=1): 3> Transferring 128 out of 257
primary partitions to member 2 requesting 128
2010-07-23 10:17:15.140/4.875 Oracle Coherence GE 3.6.0.0 DPR2 <D4>
(thread=DistributedCache, member=1): 1> Transferring 129 out of 129
partitions to a node-safe backup 1 at member 2 (under 129)
2010-07-23 10:17:15.156/4.891 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=DistributedCache, member=1): Transferring 0KB of backup[1]
for PartitionSet{128, 129, 130, 131, 132, 133, 134, 135, 136, 137,
138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151,
152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165,
166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179,
180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193,
194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221,
222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235,
236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249,
250, 251, 252, 253, 254, 255, 256} to member 2
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

```
Here I am: Customer:
      ID:8
      Name:Gina Smith
      Address:    Address:
            Street:1224 Van Ness Dr
            City:Torrance
            State:CA
            Zip:90501

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:6
      Name:Max Xavier
      Address:    Address:
            Street:335 Woodruff Ave
            City:Tampa
            State:FL
            Zip:33066

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:9
      Name:Leslie Green
      Address:    Address:
            Street:14789 Prospect Ave
            City:Jim Thorpe
            State:PA
            Zip:18229

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:10
      Name:Stephanie Kramer
      Address:    Address:
            Street:100 Remsen Ave
            City:Columbia
            State:MD
            Zip:21044

      Credit Card IDs:[]
      Order IDs:[]
```

Command Window 2:

```
Loading data in cache: 5-10
2010-07-23 10:17:14.968/2.593 Oracle Coherence GE 3.6.0.0 DPR2 <D4>
(thread=DistributedCache, member=2): Asking member 1 for 128 primary
partitions
Pausing for 15 seconds...
Here I am: Customer:
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

```
      ID:1
      Name:James Jones
      Address:    Address:
            Street:38 Clark Place
            City:San Francisco
            State:CA
            Zip:94121

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:2
      Name:Scott Riley
      Address:    Address:
            Street:44 Pumpkin Ct
            City:Shirley
            State:MA
            Zip:01464

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:3
      Name:Johnny Greif
      Address:    Address:
            Street:76 Nile Ct
            City:Arvata
            State:CO
            Zip:80001

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:7
      Name:Logan Pollard
      Address:    Address:
            Street:20 Pine St
            City:Dolan Springs
            State:AZ
            Zip:86441

      Credit Card IDs:[]
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

```
      Order IDs:[]

Here I am: Customer:
      ID:5
      Name:Chris Sandow
      Address:    Address:
            Street:865 Seminole Tr
            City:Avenel
            State:NJ
            Zip:07001

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:4
      Name:Tobey Sandor
      Address:    Address:
            Street:1920 Voss Rd
            City:Houston
            State:TX
            Zip:77057

      Credit Card IDs:[]
      Order IDs:[]
```

You have successfully configured a partitioned cache and ran it in a Coherence cluster. How do you know this time?

The next section explores these questions with you.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze local cache execution results	
4-3.9	Take a closer look at the output in each command window. The first thing you should find in each window is the line showing that your `coherence-cache-configuration.xml` file was loaded.
4-3.10	Next, you want to look for messages that have the word "DistributedCache" in them. This shows that the partitioned cache service was started, and that there were two cluster members that joined the service. This is a positive indication that a partitioned cache is configured and in use.
4-3.11	The next thing you might notice while looking at both console windows is that command window 1 and command window 2 have completely different application output. Command window 1 prints out "**Here I Am**" messages for some of the entries, and command window 2 prints out "**Here I Am**" for the other entries. You may remember when describing how these programs run, that only the first command window was going to call the `NamedCache` method that causes the "**Here I Am**" message to get printed out by all entries in the cluster. This is still true with this lab. What does this tell you about how a partitioned cache works in Coherence? Why? The reason that command window 1 shows some of the results and command window 2 shows some of the results is because although a partitioned cache contains all of the cache entries by default, it stores them in primary and backup copies across the cluster, and makes them available to every node in the cluster. This means that some entries are located on node 1, and some are on node 2. **Note:** Coherence uses a feature in this lab called an `EntryProcessor`. It may be confusing to understand at this point in the course. Do not concern yourself with understanding it at this time. Just know that it causes each entry in the cluster to print its "**Here I Am**" message in the console where the entry exists. The paragraph below explains a little more about what is happening. The call to `customers.invokeAll()` causes the `HereIAmProcessor` code to get executed where every cache entry exists in the cluster. In this case, Coherence has to call outside of the current node in order to find every entry. Although each cluster node only put half of the data into the cache, the partitioned cache automatically makes the data available to both nodes. The program that calls `invokeAll()` is the instance that has a `start` argument equal to 0 (`-s 0`). If you changed the `CacheTester.java` code to allow the `customers.invokeAll()` method to run on *both* command windows, then you would see the same output repeat in both windows, because the `HereIAmProcessor` code would be executed on every entry in the cache twice. Now that you have run a Coherence partitioned cache, and verified its behavior, let's take a look at working with near caches. **Note:** Close all command windows before proceeding.

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace and open **Practice.04.03.solution**. Follow the directions for testing the application from steps **4-3.6**, ignoring any steps to add code, all of which is in the solution.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 4-4: Configure, Run, and Review a Near Cache Topology

Skills Learned

At the end of this practice, you should be able to:

- Specify command line arguments to the Coherence server start script to override the default `coherence-cache-config.xml` file.
- Configure a near cache scheme and map it to a cache.
- Start a Coherence two-node cluster that uses the near cache.
- Identify that Coherence is using the specific configuration.
- Identify that Coherence starts the services related to the configuration.
- Review and explain where data is stored, and why it proves a near cache is in use.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Problem Statement

From:	Craig.S.List@retail-is-us.com
Sent:	Friday, July 8th, 4:32pm
To:	you@acmedev.com
Subject:	Configuring a Coherence Near Cache

Hello!

We have been using the Coherence partitioned cache configuration you provided to us earlier. So far, this is working great for us. As we mentioned earlier, a partitioned cache can support extremely large data sets, and can scale predictably for reads and writes as the number of Coherence nodes increases. We also mentioned that this came with a slight tradeoff for read performance when compared to the replicated cache. We would like to improve the read performace of our partitioned cache by using a near cache.

As we understand it, a near cache is a composite cache, or a cache that consists of two caching strategies that work together to provide the benefits of local performance, while simultaneously providing the benefits of a scalable and highly available larger distributed data set. The most common usage of the near cache is combining the local cache for the front tier, and the partitioned cache for the back tier. This allows applications to keep frequently used data close in a local cache for high-performance reading, while allowing for all the benefits of a partitioned cache; thus, effectively removing the tradeoff for read performance that comes with the partitioned cache.

Our engineers say that you can easily modify our cache configuration to change our partitioned cache scheme to a near cache scheme. We look forward to seeing an example of a Coherence near cache configuration, and understanding how to tell if our configuration is being used.

Regards,

Craig List
Retail Is Us
Craig.S.List@retail-is-us.com

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Design

In this practice you will add a near cache configuration to a `coherence-cache-config.xml` file. This file is used by Coherence to construct the various caches that the node will use. You will use the `<near-scheme>` element to configure the near cache, and the `<cache-mapping>` element to map all caches to your near scheme. The diagram below shows two Coherence nodes, called CacheTester, each running with the near cache configured.

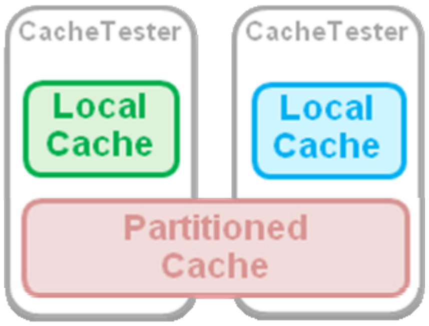


Figure 4 Near Cache Cluster

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\`**`resources`**`\`	Contains a set of practice specific resource directories
Support	`D:\oracle\student\`**`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\`**`software`**	Contains installable Coherence and other files

Coherence Cache Override (`-Dtangosol.coherence.cacheconfig`) Specifies which configuration file to use for Coherence caching	
Not Set	By default, Coherence uses the `coherence-cache-config.xml` file bundled in `%COHERENCE_HOME%\lib\coherence.jar`.
`filename`	Specifies the filename of the configuration file to use

Table 4.3

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Configure a near cache and run in a Coherence cluster.
2	Analyze near cache execution results.

Hints:

- A near cache is configured using the `<near-scheme>` element within the `<caching-schemes>` element.
- A cache is mapped to a particular scheme using the `<cache-name>` and `<scheme-name>` elements within the `<cache-mapping>` element.
- A local cache is configured as a front cache for a near cache. Use the <local-scheme> element to configure a local cache as the front cache for the near cache.
- A partitioned cache is configured as the back cache for a near cache. Use the <scheme-ref> element to reuse a partitioned cache configuration as the back cache for the near cache.
- You specify to Coherence which cache configuration file to use by setting the system property `-Dtangosol.coherence.cacheconfig=`*`your_file_name.xml`*.

Detailed Instructions

Configure a near cache and run in a Coherence cluster	
4-4.1	Within Eclipse, expand the **Practice.04.04** project and expand the **config**, **script**, and **src** folders to display their contents.
4-4.2	Open the `coherence-cache-config.xml` file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-4.3	Locate the text, "TODO: Place near cache scheme here" within the `<caching-schemes>` element, and replace with the code for a Coherence near caching scheme. **Note:** For convenience the code snippets for this exercise can be found in the resources subdirectory for **Practice.04.04** in the **code.snippet.txt** file. Your code should resemble:

```
<near-scheme>
  <scheme-name>MyNearScheme</scheme-name>

  <front-scheme>
    <local-scheme>
      <eviction-policy>HYBRID</eviction-policy>
      <high-units>100</high-units>
      <expiry-delay>1m</expiry-delay>
    </local-scheme>
  </front-scheme>

  <back-scheme>
    <distributed-scheme>
      <scheme-ref>MyPartitionedScheme</scheme-ref>
    </distributed-scheme>
  </back-scheme>

  <invalidation-strategy>present</invalidation-strategy>
  <autostart>true</autostart>
</near-scheme>
```

4-4.4	Locate the text, "TODO: Place cache-mapping here" within the `<caching-scheme-mapping>` element, and replace with the code to map a cache of any name to the near cache scheme defined in the last step. Your code should resemble:

```
<cache-mapping>
  <cache-name>*</cache-name>
  <scheme-name>MyNearScheme</scheme-name>
</cache-mapping>
```

Save your changes.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

4-4.5	Within the **src** folder, open the **`cachetester.cmd`** file for editing. Add the system property to the command line that instructs Coherence to use your `coherence-cache-config.xml` file instead of the default. Save your changes. Now that the configuration is done, you are ready to run your cluster.
4-4.6	**Note:** This version of the CacheTester program works almost the same as in the partitioned cache lab, so the text is not repeated here. The general concept is that the cluster members will call the "**Here I Am**" processor, revealing where the cache entries are located in the cluster. The main difference in this lab is the introduction of the `StopWatch` class. This class is used to time multiple reads from the cache. In addition to causing all cache entries to announce, "**Here I Am**", the code will read all cache entries five times and time each read. As it reads each entry, it prints out the timing results for that entry, then moves on to the next entry. Both cluster nodes will read (or `get`) all entries from the cache. This lab reveals how a near cache stores and retrieves its cache entries. The next few steps will instruct you how to run this scenario.
4-4.7	From within Eclipse, open two DOS prompt command windows for the **Practice.04.04** project.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

4-4.8 In both command windows type in `cachetester.cmd` to prepare them to run the cluster. **DO NOT EXECUTE THE SCRIPTS YET!**

In the first command window, the defaults are used, so no command line arguments are needed. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd
```

In the second command window, set the index range to use by specifying the arguments: **`-s 5 -f 10`**. Your command line should resemble:

```
D:\Oracle…> cachetester.cmd -s 5 -f 10
```

When ready, quickly execute the command in both command windows, and wait for them to exit. You should see output similar to:

Command Window 1:

```
Loading data in cache: 0-5
Pausing for 15 seconds...


2010-07-23 08:43:28.937/15.516 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member(Id=2, Timestamp=2010-07-23
08:43:28.937, Address=139.185.35.126:8089, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:3752,
Role=CacheTesterCacheTester) joined Cluster with senior member 1
2010-07-23 08:43:29.000/15.579 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=TcpRingListener, member=1): Connected to Member(Id=2,
Timestamp=2010-07-23 08:43:28.937, Address=139.185.35.126:8089,
MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:3752,
Role=CacheTesterCacheTester)
2010-07-23 08:43:29.046/15.625 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member 2 joined Service Management with
senior member 1
2010-07-23 08:43:29.328/15.907 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=Cluster, member=1): Member 2 joined Service DistributedCache
with senior member 1
2010-07-23 08:43:29.359/15.938 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=DistributedCache, member=1): 3> Transferring 128 out of 257
primary partitions to member 2 requesting 128
2010-07-23 08:43:29.484/16.063 Oracle Coherence GE 3.6.0.0 DPR2 <D4>
(thread=DistributedCache, member=1): 1> Transferring 129 out of 129
partitions to a node-safe backup 1 at member 2 (under 129)
2010-07-23 08:43:29.531/16.110 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=DistributedCache, member=1): Transferring 0KB of backup[1]
for PartitionSet{128, 129, 130, 131, 132, 133, 134, 135, 136, 137,
138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151,
152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165,
166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179,
180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193,
194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221,
222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235,
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

```
236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249,
250, 251, 252, 253, 254, 255, 256} to member 2
Timing reads from the cache:
==================================================
Customer: ID=10 Name=Stephanie Kramer Read
time=StopWatch{startTime=1369996208503408,
finishTime=1369996212038041, elapsedTime=3534633 ns (0.003535 sec)}
Customer: ID=10 Name=Stephanie Kramer Read
time=StopWatch{startTime=1369996230492404,
finishTime=1369996230520470, elapsedTime=28066 ns (0.000028 sec)}
Customer: ID=10 Name=Stephanie Kramer Read
time=StopWatch{startTime=1369996231348218,
finishTime=1369996231353932, elapsedTime=5714 ns (0.000006 sec)}
Customer: ID=10 Name=Stephanie Kramer Read
time=StopWatch{startTime=1369996231547030,
finishTime=1369996231551757, elapsedTime=4727 ns (0.000005 sec)}
Customer: ID=10 Name=Stephanie Kramer Read
time=StopWatch{startTime=1369996231739919,
finishTime=1369996231744629, elapsedTime=4710 ns (0.000005 sec)}
==================================================
Customer: ID=6 Name=Max Xavier Read
time=StopWatch{startTime=1369996231963813,
finishTime=1369996234259822, elapsedTime=2296009 ns (0.002296 sec)}
Customer: ID=6 Name=Max Xavier Read
time=StopWatch{startTime=1369996234465527,
finishTime=1369996234470952, elapsedTime=5425 ns (0.000005 sec)}
Customer: ID=6 Name=Max Xavier Read
time=StopWatch{startTime=1369996234660396,
finishTime=1369996234665093, elapsedTime=4697 ns (0.000005 sec)}
Customer: ID=6 Name=Max Xavier Read
time=StopWatch{startTime=1369996234851457,
finishTime=1369996234856125, elapsedTime=4668 ns (0.000005 sec)}
Customer: ID=6 Name=Max Xavier Read
time=StopWatch{startTime=1369996235042482,
finishTime=1369996235047081, elapsedTime=4599 ns (0.000005 sec)}
==================================================
Customer: ID=9 Name=Leslie Green Read
time=StopWatch{startTime=1369996235261844,
finishTime=1369996236993637, elapsedTime=1731793 ns (0.001732 sec)}
Customer: ID=9 Name=Leslie Green Read
time=StopWatch{startTime=1369996237210689,
finishTime=1369996237216039, elapsedTime=5350 ns (0.000005 sec)}
Customer: ID=9 Name=Leslie Green Read
time=StopWatch{startTime=1369996237400944,
finishTime=1369996237405627, elapsedTime=4683 ns (0.000005 sec)}
Customer: ID=9 Name=Leslie Green Read
time=StopWatch{startTime=1369996237591783,
finishTime=1369996237596348, elapsedTime=4565 ns (0.000005 sec)}
Customer: ID=9 Name=Leslie Green Read
time=StopWatch{startTime=1369996237782754,
finishTime=1369996237787455, elapsedTime=4701 ns (0.000005 sec)}
==================================================
. . .
```

Command Window 2:

```
Loading data in cache: 5-10
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

```
2010-07-23 08:43:29.328/2.485 Oracle Coherence GE 3.6.0.0 DPR2 <D4>
(thread=DistributedCache, member=2): Asking member 1 for 128 primary
partitions
Pausing for 15 seconds...
Here I am: Customer:
      ID:1
      Name:James Jones
      Address:    Address:
            Street:38 Clark Place
            City:San Francisco
            State:CA
            Zip:94121

      Credit Card IDs:[]
      Order IDs:[]

Here I am: Customer:
      ID:2
      Name:Scott Riley
      Address:    Address:
            Street:44 Pumpkin Ct
            City:Shirley
            State:MA
            Zip:01464

      Credit Card IDs:[]
      Order IDs:[]
. . .
Timing reads from the cache:
==================================================
Customer: ID=5 Name=Chris Sandow Read
time=StopWatch{startTime=1370009559227922,
finishTime=1370009560969204, elapsedTime=1741282 ns (0.001741 sec)}
Customer: ID=5 Name=Chris Sandow Read
time=StopWatch{startTime=1370009565166921,
finishTime=1370009565187248, elapsedTime=20327 ns (0.000020 sec)}
Customer: ID=5 Name=Chris Sandow Read
time=StopWatch{startTime=1370009565388439,
finishTime=1370009565393248, elapsedTime=4809 ns (0.000005 sec)}
Customer: ID=5 Name=Chris Sandow Read
time=StopWatch{startTime=1370009565586862,
finishTime=1370009565591587, elapsedTime=4725 ns (0.000005 sec)}
Customer: ID=5 Name=Chris Sandow Read
time=StopWatch{startTime=1370009565781244,
finishTime=1370009565785927, elapsedTime=4683 ns (0.000005 sec)}
================================================
```

You have successfully configured a near cache and ran it in a Coherence cluster. How do you know this time?

The next section explores these questions with you.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze near cache execution results	
4-4.9	Take a closer look at the output in each command window. The first thing you should find in each window is the line showing that your `coherence-cache-configuration.xml` file was loaded.
4-4.10	Next, you want to look for messages that have the word "DistributedCache" in them. This shows that the partitioned cache service was started, and that there were two cluster members that joined the service. This is a positive indication that a partitioned cache is configured and in use. However, there is no way to indicate that a near cache is in use this way. At this point, you cannot tell if this partitioned cache is part of a near caching scheme or not.
4-4.11	The next thing you might notice while looking at both console windows is that command window 1 and command window 2 have similar output to Practice 4-3 as far as the "**Here I Am**" messages. This is because the `customers.invokeAll()` method called on a near caching scheme runs on the back tier, which in this case is a partitioned scheme, so the results should be similar to the last lab. Now you should look for the printout that shows the reads and the timing of the reads. Each entry is read from the cache five times on each node, and the time it took to read the data is printed along with each read result. Examine the sets of read entries and their timings to see if you notice any trends in the performance numbers. What does this tell you about how a near cache works in Coherence? Why? What you should see is that the first read for each entry takes much longer to process than the subsequent reads. You should also notice that all subsequent reads take about the same amount of time. This is because the first read involved going to the back tier to get the data from the partitioned cache then loading the near cache with the data. All susequent calls for that data item were handled solely by the local near cache, which avoided any network and serialization processing. Now you have run a Coherence near cache, and verified its behavior. You should now feel comfortable working with the different types of Coherence caching topologies. **Note:** Close all command windows before finishing.

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace and open **Practice.04.04.solution**. Follow the directions for testing the application from steps **4-4.6**, ignoring any steps to add code, all of which is in the solution.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 5

Chapter 6

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 5

Practices Overview

Map Listeners, map triggers, and backing map listeners provide support for handing events at a variety of points in an object's lifecycle, as well as at different locations within a Coherence cluster.

In this practice, you will perform the following tasks:

- Create and register a map listener programmatically.
- Create and register a map trigger both programmatically and declaratively.
- Create and register a backing map listener.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 5-1: Working with MapListeners

Skills Learned

At the end of this practice, you should be able to:

- Extend an existing class to implement the `MapListener` interface to manage an additional cache
- Add a map listener programmatically to a cache
- Test a map listener registered programmatically

Problem Statement

From:	Billi.Onair@retail-is-us.com
Sent:	Monday, July 11th, 6:05am
To:	you@acmedev.com
Subject:	Managing a state cache using events

Hello!

Our Coherence project continues, and we understand there is some sort of event model, which can listen for inserts, updates, and deletes. Coherence event support might solve an issue we are struggling with.

As you know, our data loader stores customers into an appropriate cache. Along with each customer is an address. We would like to maintain a state list as well. But rather than add a new loader, we understand that we can listen for inserts to the customer table, and perhaps extract the address and create the states that way. In this way, we would be letting the customer data drive the state data. When the load is complete, any states should be in the new cache.

Our engineers say this sounds like an excellent application of a `MapListener`, and that you have the expertise to create a class that can extract address data on inserts and add it to a new cache.

Regards,

Billi Onair
Retail Is Us
Billi.Onair @retail-is-us.com

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Design

In this practice, there are several classes that have not been encountered before. The `State` class encapsulates a simple state code as a `String`. Additionally, a partial implementation of `AddStateMapListener` is provided which must be completed to be used as an actual `MapListener`. The `CoherenceStateRepository` class is used to externalize the name of the state cache, and follows the pattern described in earlier lessons, but is otherwise a convenience only. Finally, the `TestMapListener` class must be updated to register the map listener against the customer cache.

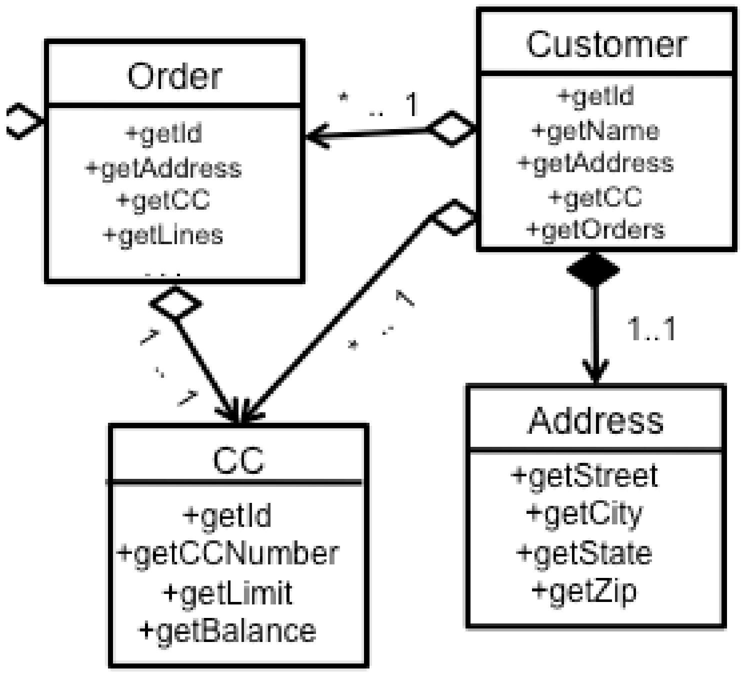


Figure 1 Partial Retail Model

Oracle University and ORACLE CORPORATION use only

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\ resources\`	Contains a set of practice specific resource directories, including code snippets for this practice

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Open Eclipse, and if required, import the project.
2	Modify a class to implement the `MapListener` interface.
3	Register the listener programmatically against the Customer cache.
4	Test using the provided **TestMapListener** class and run configuration.
5	Analyze execution results.

Hints:

- `MapListener`s must implement `com.tangosol.util.MapListener`.
- Map listeners can be registered against a `NamedCache` instance using the `AddMapListner(MapListener listener)` method.
- The `com.tangosol.util.MapEvent.getNewValue()` method can be used on inserts to obtain the `Object` which was inserted into the cache.
- The `com.oracle.education.retail.Customer` class is the class being inserted. Customer addresses can be retrieved via the `getAddress()` method.
- `com.oracle.education.retail.Address` contains a state which can be retrieved via the `getState()` method.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to make. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects except the current one to minimize the number of tasks displayed.

Detailed Instructions

Open Eclipse and import the project if required	
5-1.1	Within Eclipse, expand the **Practice.05.01** project and expand the **src** folder to display its contents.
5-1.2	If asked to open dependent projects, choose Yes. **Practice.05.01** depends on **DomainExamples**.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

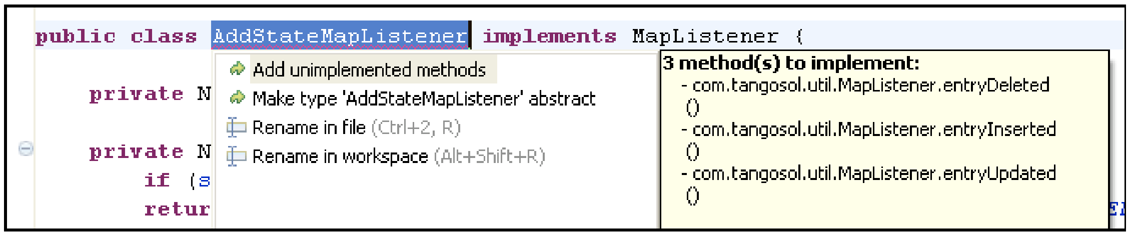
Modify a class to implement the MapListener interface

5-1.3 Navigate to the **retail.listener** package and open the **AddStateMapListener.java** file. **Note**: This class has been partially developed.

5-1.4 Locate the text “TODO: Must implement the MapListener” interface, and modify the class to implement the map listener interface. The modified code should resemble:

```
public class AddStateMapListener implements MapListener {
. . .
```

5-1.5 The class will now show errors. Using either quick fix or modifying by hand, add the three required methods:



```
public class AddStateMapListener implements MapListener {

    private N
    private N
        if (s
        retur
```

Add unimplemented methods
Make type 'AddStateMapListener' abstract
Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)

3 method(s) to implement:
- com.tangosol.util.MapListener.entryDeleted ()
- com.tangosol.util.MapListener.entryInserted ()
- com.tangosol.util.MapListener.entryUpdated ()

Note that the required imports, shown below, have been previously added to the class:

```
import com.tangosol.util.MapEvent;
import com.tangosol.util.MapListener;
```

The three generated methods should resemble:

```
public void entryDeleted(MapEvent arg0) {
   // TODO Auto-generated method stub
}
public void entryInserted(MapEvent evt) {
   // TODO Auto-generated method stub
}
public void entryUpdated(MapEvent evt) {
   // TODO Auto-generated method stub
}
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

5-1.6 *Optionally*, you may modify the **entryDeleted** method to resemble:

```
public void entryDeleted(MapEvent evt) {
  System.out.println("\n\n+++entryDeleted++++\n");
  printEvent(evt);
}
```

The **printEvent** method is provided for you, and dumps the event contents to standard out.

5-1.7 *Optionally*, you may modify the **entryUpdated** method to resemble:

```
public void entryUpdated(MapEvent evt) {
  System.out.println("\n\n+++entryUpdated++++\n");
  printEvent(evt);
}
```

5-1.8 Find the **entryInserted** method and modify it to resemble the code shown below. **Note:** For convenience, the code snippets for this exercise can be found in the resources subdirectory for **Practice.05.01** in the **code.snippet.txt** file. The completed code should resemble:

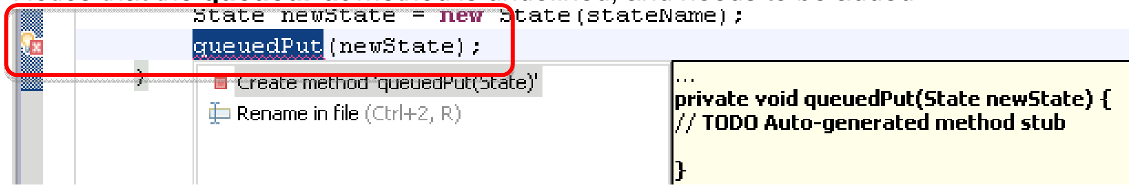
```
public void entryInserted(MapEvent evt) {
    System.out.println("\n\n+++entryInserted++++\n");
    printEvent(evt);
    Object o = evt.getNewValue();
    if (! ( o instanceof Customer ) ) {
       return;
    }
    Customer customer = (Customer)o;
    System.out.println("Received insert of customer " +
                      customer);
    Address address = customer.getAddress();
    String stateName = address.getState();
    State newState = new State(stateName);
    queuedPut(newState);
}
```

Note: The provided code uses a provided **StateWorker** runnable class via the **queuePut** method. **StateWorker** decouples the event method from the put method on the state cache, which could cause deadlock.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

5-1.9 Notice that the **queuedPut** method is undefined, and needs to be added.



```
State newState = new State(stateName);
queuedPut(newState);
}
```

```
...
private void queuedPut(State newState) {
// TODO Auto-generated method stub

}
```

Using the Eclipse quickfix feature or by hand add the **queuedPut** method. Note that this method uses a state variable that must also be specified. Your code should resemble:

```
StateWorker worker = null;
private void queuedPut (State state) {
  if ( worker == null) {
        worker = new StateWorker();
        new Thread(worker).start();
  }
  worker.storeState(state);
}
```

Note: For convenience, this code snippet can be found in the resources subdirectory for **Practice.05.01** in the **code.snippet.queuePut.txt** file.

5-1.10 Save your changes.

Register the listener programmatically against the Customer **cache**

5-1.11 In the `retail.test` package, open the `TestMapListener.java` file.

5-1.12 Find the TODO resembling:

```
// TODO: Create a customer cache using the name
//       CoherenceCustomerRepository.CACHENAME
```

5-1.13 Add code to create a customer cache instance, the map listener will be registered against this instance. Your code should resemble:

```
NamedCache customerCache=
   CacheFactory.getCache(CoherenceCustomerRepository.CACHENAME);
```

Note: The import for `CoherenceCustomerRepository` has been added for you.

5-1.14 Register the map listener against the new cache using the `AddMapListener()` method. Your code should resemble:

```
NamedCache customerCache=
   CacheFactory.getCache(CoherenceCustomerRepository.CACHENAME);
customerCache.addMapListener(new AddStateMapListener());
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

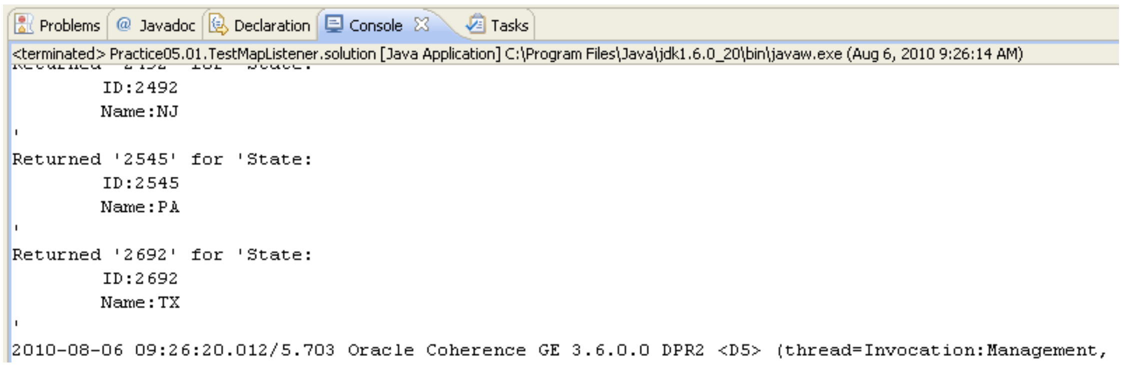
5-1.15 At the end of the main method, examine the code for dumping the cache data. Note the use of the `entrySet` method to return all the key/value pairs in the state cache.

Test using the provided TestMapListener class and run configuration

5-1.16 Make sure the **Practice.05.01** project is selected, or its solution if testing the solution.

- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the Eclipse.
- If **Practice.05.01[.solution]** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice.05.01[.solution]** in the left-hand window.

This starts the test and displays its results in the console window.



```
<terminated> Practice05.01.TestMapListener.solution [Java Application] C:\Program Files\Java\jdk1.6.0_20\bin\javaw.exe (Aug 6, 2010 9:26:14 AM)
        ID:2492
        Name:NJ
'
Returned '2545' for 'State:
        ID:2545
        Name:PA
'
Returned '2692' for 'State:
        ID:2692
        Name:TX
'
2010-08-06 09:26:20.012/5.703 Oracle Coherence GE 3.6.0.0 DPR2 <D5> (thread=Invocation:Management,
```

Analyze execution results

5-1.17 Examine the console results closely, scrolling both windows to find a statement similar to: "**Loading 10 customers in cache**".
Each customer being added should have triggered an insert which was hooked by the map listener, generating output similar to:

```
MapEvent:id:1'ENTRY_INSERTED'
Key:1
New value:Customer:
    ID:1
    Name:James Jones
    Address:     Address:
      Street:38 Clark Place
      City:San Francisco
      State:CA
      Zip:94121

    Credit Card IDs:[1]
    Order IDs:[1, 2]

Old value:null
. . .
```

5-1.18 Each insert should have generated a corresponding "**Adding State**" text, output beneath

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

	the original insert which resembles:
	
	`Adding State 'State:`
	`    ID:2142`
	`    Name:CA`
	`'.`

5-1.19	Search for a statement similar to: “**Cache is loaded**”, which should be near the bottom of the console output.
5-1.20	Analyze the dump of the state cache. If the map listener was coded successfully, a set of states should have been added, and the entire set is output in ascending order by state code. The complete results should resemble: Returned '2105' for 'State: ID:2105 Name:AZ ' . . . Returned '2692' for 'State: ID:2692 Name:TX ' Note that the states are show in ascending order. This is a result of the `State` object implementing the `Comparable` interface.
5-1.21	Close any open projects and editor windows and proceed to the next exercise.

Solution Instructions

To run the solution, start Eclipse and open the **D:\Oracle\student\practices** workspace and open **Practice.05.01.solution**.

Follow the directions for testing the application from steps 5-1.15.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 5-2: Working with MapTriggers

Skills Learned

At the end of this practice, you should be able to:

- Extend an existing class to implement the `MapTrigger` interface
- Extract an object on a MapTrigger call, and modify its contents
- Add a map listener programmatically to a cache
- Test a map listener registered programmatically

Oracle University and ORACLE CORPORATION use only

Problem Statement

From:	Tom.Foolery@retail-is-us.com
Sent:	Tuesday, July 12th, 11:38am
To:	you@acmedev.com
Subject:	Modifying data prior to inserts

Good morning,

We are running into a minor snag, and understand you can help.

We have an application that is working with items. However, it appears the item SKU field is always in upper case when it should be in lower case. We understand that by using a map trigger, we can intercept the inserts into the item cache, and replace the SKU value with its lower case equivalent.

Since we had such success with your implementation of a map listener, we heard you might be able to help here as well. We look forward to your work!

Regards,

Tom Foolery
Retail Is Us
Tom.Foolery@retail-is-us.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Design

In this practice, there are several classes that have not been encountered before. The `Item` class encapsulates a variety of item-specific information as strings. Additionally, a partial implementation of `ToLowerMapTrigger` exists, which must be extended to be used as an implementation of a `MapTriggerener`. The `TestMapTrigger` class is provided, but must have its main method updated to register the map trigger, via a `MapTriggerListener`.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Overview Instructions

1	Start Eclipse and open the provided project.
2	Modify the **`retail.trigger.ToLowerMapTrigger`** class to complete the **`MapTrigger`** implementation.
3	Register the `MapTrigger` programmatically against the **`Item`** cache.
4	Test using the provided TestMapTrigger class and run configuration.
5	Analyze the execution results.

Hints:

- `MapListener`s must implement `com.tangosol.util.MapTrigger`.
- Map triggers can be registered against a `NamedCache` instance using the `AddMapListner(MapListener listener)` method and wrapping the trigger in a `MapTriggerListener` instance.
- The `com.tangosol.util.MapEvent.getNewValue()` can be used on inserts to obtain the `Object` which was inserted into the cache.
- The `com.oracle.education.retail.Item` class is the class being inserted. Item SKUs can be retrieved via the `getSku`() method.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Detailed Instructions

Start Eclipse and open the provided project

5-2.1 Within Eclipse, expand the **Practice.05.02** project and expand the **src** folder to display its contents. **Note: Practice.05.02** depends on **DomainExamples**.

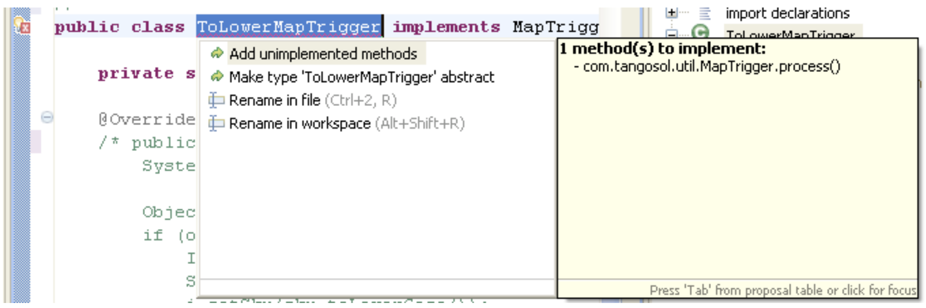
Modify the `retail.trigger.ToLowerMapTrigger` to complete the MapTrigger implementation

5-2.2 Navigate to the **`retail.trigger`** package and open the **`ToLowerMapTrigger.java`** file.
Note: This class has been partially developed.

5-2.3 Locate the text "TODO: Must implement the MapLister interface" and modify the class to implement the `MapListener` interface. The modified code should resemble:

```
public class ToLowerMapTrigger implements MapTrigger {
. . .
```

5-2.4 The class will now show errors. Using either quick fix or modifying by hand, add the required *process* method:


Add unimplemented methods
Make type 'ToLowerMapTrigger' abstract
Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)
import declarations
ToLowerMapTrigger
Press 'Tab' from proposal table or click for focus

```
public class ToLowerMapTrigger implements MapTrigg
    private s
    @Override
    /* public
        Syste
        Objec
        if (o
```

1 method(s) to implement:
- com.tangosol.util.MapTrigger.process()

Note that the required imports, shown below, have been previously added to the class:

```
import com.tangosol.util.MapEvent;
import com.tangosol.util.MapTrigger;
```

The generated method should resemble:

```
public void process(Entry entry) {
    // TODO Auto-generated method stub
}
```

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

5-2.5	Find the **`process`** method and modify it to resemble the code shown below. **Note:** For convenience, the code snippets for this exercise can be found in the resources subdirectory for **Practice.05.02** in the **`code.snippet.txt`** file. Note that the imports for the **`Item`** object were provided. The completed code should resemble:

```
 public void process(Entry entry) {
    System.out.println("\n\nProcess('" + entry + "')");

    Object o = entry.getValue();
    if (o != null && (o instanceof Item )) {
       Item i= (Item)o;
       String sku = i.getSku();
       i.setSku(sku.toLowerCase());
       entry.setValue(i);
    }
}
```

5-2.6 Save your changes.

Register the `MapTrigger` programmatically against the `Item` cache

5-2.7 In the **`retail.test`** package, open the **`TestMapTrigger.java`** file.

5-2.8 Find the TODO resembling:

```
// TODO: Create an instance of a map trigger
```

5-2.9 Register the map listener against the new cache using the **`AddMapListener()`** method, however you will need to wrap the map trigger in a `MapTriggerListener` instance. Your code should resemble:

```
ToLowerMapTrigger trigger = new ToLowerMapTrigger();
MapTriggerListener listener = new MapTriggerListener(trigger);
itemCache.addMapListener(listener);
```

5-2.10 Note that this method also contains a loop for displaying the results similar to the last practice but using the item cache.

5-2.11 Save your changes.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Test using the provided TestMapTrigger class and run configuration

5-2.12 Make sure the **Practice.05.02** project is selected, or its solution if testing the solution.

- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the Eclipse.
- If **Practice.05.02[.solution]** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice.05.02[.solution]** in the left-hand window.

Analyze execution results

5-2.13 Examine the console window, scrolling to the bottom if required.

5-2.14 Examine the dump of the items. Is the SKU field in the format expected?

5-2.15 *Optionally*, comment out the code line:

```
itemCache.addMapListener(listener);
```

5-2.16 Rerun the test. Did you see the expected result?

Solution Instructions

To run the solution, start Eclipse and open the **D:\Oracle\student\practices** workspace, then open **Practice.05.01.solution**.

Follow the directions for testing the application from steps 5-2.12.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 5-3: Working with backing map listeners

Skills Learned

At the end of this practice, you should be able to:

- Complete a backing map listener, adding code to convert data from serialized to deserialized form.
- Register a backing map listener within a cache scheme.
- Map a cache to a cache scheme.
- Test a backing map listener.

Problem Statement

From:	Tom.Foolery@retail-is-us.com
Sent:	Tuesday, July 12th, 3:17pm
To:	you@acmedev.com
Subject:	Backing map listeners

Good morning,

I've heard that we can save a lot of time and bandwidth by using a "backing map listener." Since you've been so successful with all the other event based projects we were wondering if you could write a simple backing map listener which echoes events as they come in, and provide it, with an example of how to register it?

Thanks in advance!

Regards,

Tom Foolery
Retail Is Us
Tom.Foolery@retail-is-us.com

Design

This practice requires two tasks: completion of a backing map listener and registration of the listener via a cache-scheme, as well as mapping of the cache name to a cache scheme.

An `AbstractBackingMap` listener class has been provided which is extended by `EchoingBackingMapListener`. The echoing listener is then added to a cache configuration and used by the customer cache to show a backing map listener in action.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects except the current one to minimize the number of tasks displayed.

Resource Directories

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Resource Directories		
Practice Resources	`D:\oracle\student\resources\`	Contains a set of practice specific resource directories, including code snippets for this practice

Overview Instructions

1	Open Practice.06.03.
2	Complete the **AbstractBackingMap** class implementation.
3	Modify **`coherence-cache-config.xml`** to define a cache-scheme with a BML.
4	Test using the Practice.06.03 run configuration.
5	Analyze execution results.

Hints:

- The `BackingMapManager` provided as a constructor argument to a backing map listener contains methods for converting/returning data. For example, to convert from an interval, code such as:
```
Object o;
o =
context.getValueFromInternalConverter().convert(entry.getNewValue
());
```
might be used.
- A cache name is bound to a cache scheme using a `<cache-mapping>` element within the `<caching-scheme-mapping>` element of `coherence-cache.config.xml`. For example:
```
    <cache-mapping>
         <cache-name>SomeCacheName</cache-name>
         <scheme-name> SomeCacheName </scheme-name>
    </cache-mapping>
```
Maps a cache named *`SomeCacheName`* against a scheme named *`SomeCacheName.`*
- A caching scheme is defined within the `<caching-schemes>` element of the Coherence cache configuration XML file.
- A Backing map listener is registered as a listener element within scheme element itself within a backing-map-scheme element such as:

```
<backing-map-scheme>
    <local-scheme>
        <listener>
            <class-scheme>
                <class-name>
                   fully.qualified.path.to.listener</class-name>
                     <init-params>
                        <init-param>
                            <param-type>
               com.tangosol.net.BackingMapManagerContext
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

```
                                </param-type>
                                <param-value>
                                 {manager-context}
                            </param-value>
                            </init-param>
                        </init-params>
                 </class-scheme>
             </listener>
           </local-scheme>
       </backing-map-scheme>
```

Start Eclipse and open the provided project	
5-3.1	Within Eclipse, expand **Practice.05.03** project and expand the **src** folder to display its contents. **Note: Practice.05.03** depends on **DomainExamples**.
Complete the AbstractBackingMap **class implementation**	
5-3.2	Navigate to the **`retail.listener`** package and open the **`AbstractBackingMapListener.java`** file. **Note:**The bulk of this class has been implemented.

5-3.3	Locate: “TODO: Insert a implementation for this method “.
	Complete the implementation of the **`getConvertedValue`** method by adding code that uses a backing map context manager to transform an object from serialized to deserialized form. The completed method should resemble:
	
	`private Object getConvertedValue (Object internal) {`
	`   if (internal == null) return null;`
	`   return`
	`context.getValueFromInternalConverter().convert(internal);`
	`    return o;`
	`   }`
	
	For your convenience, a complete implementation can be found in the **resources\Practice.05.03 directory. Note:** The code snippet includes debug statements, which are optional and not included in the listing above.

5-3.4	Save your work.

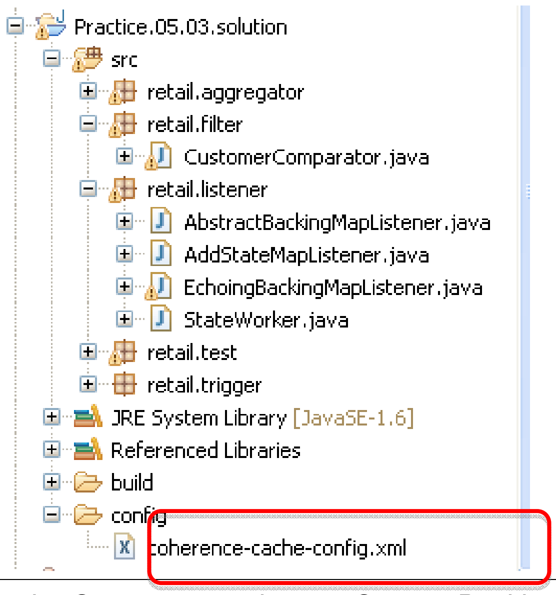
Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Modify `coherence-cache-config.xml` to define a cache-scheme with a BMP.

5-3.5 Navigate to the `config` directory and open the `coherence-cache-config.xml` file.



5-3.6 Find the TODO resembling: "TODO: Map the Customers cache to a Custom Backing Map Scheme which uses the retail.listener.EchoingBackingMapListener."
And add a new **`cache-mapping`** for the Customer cache.
When complete, the new mapping should resemble: Note that the mapping maps to a scheme that has not been added yet.

```
    . . .
    <cache-mapping>
      <cache-name>Customers</cache-name>
      <scheme-name>CustomBackingMapScheme</scheme-name>
    </cache-mapping>

</caching-scheme-mapping>
. . .
```

5-3.7 Find the TODO: resembling: "TODO: Add a new caching scheme which as its backing map scheme uses a local scheme which references a listener."
Note: You are to insert a *new* scheme here, not update the existing scheme!

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

5-3.8	Add a new custom scheme based on the `DistributedCache` service. The basic scheme should resemble:
	`<distributed-scheme>`
	`      <scheme-name>`**`CustomBackingMapScheme`**` </scheme-name>`
	`      <service-name>DistributedCache</service-name>`
	
	`     <autostart>true</autostart>`
	`</distributed-scheme>`

5-3.9	Extend the new scheme to add a **`backing-map-scheme`** element, inside of which is nested a **`local-scheme`** element. These two elements should resemble:
	`<distributed-scheme>`
	`. . .`
	`    <backing-map-scheme>`
	`        <local-scheme>`
	`        </local-scheme>`
	`    </backing-map-scheme>`
	`</distributed-scheme>`
	
	Note that the entire element can be found in the resources directory as **`snippet.CustomBackingMapScheme.txt`**. Since this element is rather long, it is suggested that students cut and paste the XML.

5-3.10	Inside the local scheme, add a **`listener`** based on the **`EchoingBackingMap`** listener which was provided, and extends **`AbstractBackingMapListener`**. The **`listener`** element should resemble:
	`<backing-map-scheme>`
	`    <local-scheme>`
	`        <listener>`
	`        </listener>`
	`    </local-scheme>`
	`</backing-map-scheme>`
	
	>

5-3.11	Inside the listener, add a **`class-scheme`** which itself contains a **`class-name`** element. The **`class-scheme`** should resemble:
	`<listener>`
	`    <class-scheme>`
	`       <class-`
	`name>retail.listener.EchoingBackingMapListener</class-name>`
	`    </class-scheme>`
	`</listener>`

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

5-3.12 After the class-name element, add the required **`init-params`** element to correctly construct the backing map listener. The `init-params` element should resemble:

```
<class-scheme>
    <class-name>. . . </class-name>
    <init-params>
        <init-param>
            <param-type>
            com.tangosol.net.BackingMapManagerContext
            </param-type>
           <param-value>{manager-context}</param-value>
        </init-param>
    </init-params>
</class-scheme>
```

5-3.13 The completed distributed scheme element should resemble:

```
<caching-schemes>
 <distributed-scheme>
  <scheme-name>CustomBackingMapScheme</scheme-name>
  <service-name>DistributedCache</service-name>
  <backing-map-scheme>
   <local-scheme>
    <listener>
     <class-scheme>
      <class-name>
        retail.listener.EchoingBackingMapListener</class-name>
      <init-params>
      <init-param>
      <param-type>
       com.tangosol.net.BackingMapManagerContext</param-type>
      <param-value>{manager-context}</param-value>
       </init-param>
              </init-params>
            </class-scheme>
   </listener>
  </local-scheme>
 </backing-map-scheme>
 <autostart>true</autostart>
</distributed-scheme>.
```

5-3.14 Save your work.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Test using the Practice.05.03 run configuration

5-3.15 Navigate to the **`retail.test`** package and open the **`TestBackingMapListener`**. Class. Examine the class, noticing that no specialized code exists for loading the cache store.

5-3.16 Make sure **Practice.05.03** project is selected, or its solution if testing the solution.

- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the Eclipse.
- If **Practice.05.03[.solution]** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice.05.03[.solution]** in the left-hand window.

This starts the test and displays its results in the console window.

HINT: Use the Console tab's Display Selected Console icon to quickly switch between command windows.

Analyze execution results

5-3.17 Examine the console results closely. Notice the console has many statements similar to:

```
+++entryInserted++++
MapEvent:id:1'ENTRY_INSERTED'
Key:8
New value:Customer:
    ID:8
    Name:Gina Smith
    Address:     Address:
      Street:1224 Van Ness Dr
      City:Torrance
      State:CA
      Zip:90501

    Credit Card IDs:[13, 14]
    Order IDs:[16]

Old value:null
```

Each of those entries represents an add to the underlying **`Customer`** cache, and is showing that the backing map listener was being called for each operation performed by the **`Loader.load()`** method.

Solution Instructions

To run the solution, start Eclipse and open the **D:\Oracle\student\practices** workspace, then open **Practice.05.03.solution**. Follow the directions for testing the application from steps 5-3.15.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 6

Chapter 7

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 6

Practices Overview

Filtering, sorting, and aggregating data are commonplace activities within any data-rich application, and Coherence applications are no exception.

In this practice, you will perform the following tasks:

- Create and use Filters and comparators to subset and sort data.
- Create and use aggregators to perform data aggregation.
- Simplify data subsetting using QueryHelpers and the Coherence Query Language.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 6-1: Filtering and Sorting Data

Skills Learned

At the end of this practice, you should be able to:

- Developer a `Comparator` to use in sorting cache data.
- Use and `EqualsFilter` to subset data.
- Test a filter programmatically.

Problem Statement

From:	V.Key@retail-is-us.com
Sent:	Wednesday,July 13th, 7:12pm
To:	you@acmedev.com
Subject:	Filtering data

Hi,

I understand it is a relatively simple process to filter Coherence Cache Data.

What we were hoping you could do is to create some sort of filter for the data to select any address in the state of California. Apparently, all of our customer data has an address which contains a state. We were also hoping the data could then be sorted by customer name. I look forward to seeing how this small project comes together!

Regards,

Veronica Key
Retail Is Us
V.Key @retail-is-us.com

Design

In this practice, there is an instance of `java.util.Comparator` which we haven't encountered yet. Data ordering can happen in either of two ways with Coherence: via the domain object implementing `Comparable`, or by writing custom comparators In fact, if you want to use the `entrySet` methods on caches, one or the other of these methods is required.

For reference, the entry set method resembles:

- QueryMap.entrySet(Filter filter, java.util.Comparator comparator);

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\ resources\`	Contains a set of practice specific resource directories, including code snippets for this practice

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Open Eclipse, and, if required, import the project.
2	Modify the **`CustomerComparator`** class to implement the `Comparator` interface.
3	Modify **`TestMapFilter`** to create a filter based on a customers state field.
4	Test using the Practice.06.01 run configuration.
5	Analyze execution results.

Hints:

- The class **`CustomerComparator`** must implement the `java.util.Comparator` interface and its single **`compare`** method.
- An `EqualsFilter` can be used to extract a field's value and compare it to a String.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Detailed Instructions

Open Eclipse and import the project if required

6-1.1 Within Eclipse, expand project **Practice.06.01** and expand the **src** folder to display its contents.

Modify the `CustomerComparator` class to implement the `Comparator` interface.

6-1.2 Navigate to the **`retail.filter`** package and open the class **`CustomerComparator.java`**.
Note: This class has been partially developed .

6-1.3 Locate the text TODO, and modify the class to implement the `Comparator` interface. The modified code should resemble:

```
public class CustomerComparator implements Comparator {
...
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

6-1.4 The class will not show any errors, but has a dummy compare method. Complete the compare method. The method should resemble:

```
public int compare(Object first, Object second) {
  if (!( first instanceof Customer) ||
      !( second instanceof Customer)){
   throw new IllegalArgumentException("object!= Customer");
  }

  Customer customerOne = (Customer) first;
  Customer customerTwo = (Customer) second;

return customerOne.getName().compareTo(customerTwo.getName());
}
```

Note that for convenience, a snippet of code can be cut and pasted from directory **`Resources\Practice06.01`**.

6-1.5 Save your changes.

Modify `TestMapFilter` to create a filter based on a customers state field

6-1.6 In the **`retail.test`** package, open the class **`TestMapFilter.java`**.

6-1.7 Find the TODO resembling: "`// TODO: Add a simple Equals filter…`"

6-1.8 Add an `EqualsFilter` which should filter on the address' state field. The complete code should resemble:

```
Filter stateFilter =
        new EqualsFilter("getAddress.getState","CA");
```

6-1.9 Find the TODO resembling: "`TODO: Complete the Customer comparator.` "

6-1.10 Modify the code to provide as the second argument to the `entrySet` method the newly-developed comparator method. When complete, the code should resemble:

```
Set<Map.Entry>  entries =
    customerCache.entrySet(null,new CustomerComparator());
```

6-1.11 Find the TODO resembling:
`TODO: Supply the new filter as the first argument and the comparator as the 2nd`

6-1.12 Update the statement to specify the new filter as the first argument to the `entrySet` method and the comparator as the second. When complete, the code should resemble:

```
entries = customerCache.entrySet(
                 stateFilter,new CustomerComparator());
```

6-1.13 Save your changes.

Test using the Practice.06.01 run configuration

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

6-1.14 Make sure project **Practice.06.01** is selected, or its solution if testing the solution.

- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the Eclipse.
- If **Practice.06.01[.solution]** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice.06.01 [.solution]** in the left-hand window.

This starts the test and displays its results in the console window.

HINT: Use the Console tab's Display Selected Console icon to quickly switch between command windows.

Analyze execution results

6-1.15 Examine the console results closely, scrolling towards the bottom. Find a statement similar to: **All entries = '10', filtered size '2'**.
After the filter operation there should be only a few Customers remaining, specifically two:

```
Returned '8' for 'Customer:
   ID:8
   Name:Gina Smith
   Address:     Address:
     Street:1224 Van Ness Dr
     City:Torrance
     State:CA
     Zip:90501

   Credit Card IDs:[17, 16]
   Order IDs:[16, 14, 15]
'
Returned '1' for 'Customer:
   ID:1
   Name:James Jones
   Address:     Address:
     Street:38 Clark Place
     City:San Francisco
     State:CA
     Zip:94121

   Credit Card IDs:[1, 2, 3]
   Order IDs:[1, 2]. . .
```

6-1.16 Are the two customers from the state you expected? Are they sorted in the order you expected? Does the order match the IDs? What happens if you change the state to something else, such as 'MA'?

6-1.17 Close any open projects and editor windows, and proceed to the next exercise.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Solution Instructions

To run the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace and open the **Practice.06.01.solution**.

Follow the directions for testing the application from steps 6-1.14.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 6-2: Developing and using Aggregators

Skills Learned

At the end of this practice, you should be able to:

- Implement a custom Aggregator class.
- Execute an Aggregator and examine the result.

Problem Statement

From:	V.Key@retail-is-us.com
Sent:	Thursday, July-14th, 5:55am
To:	you@acmedev.com
Subject:	Creating data rollups

Good evening,

Sorry for the late email, but on additional thing. Apparently one of the members of our groups needs to create some data rollup support. We understand that Coherence provides a feature known as Aggregators, and that you would be the resource for implementing an example. We were hoping you could create a simple string concatenating aggregator that we could then use as an example.

I look forward to seeing the end product of this effort.

Oh, and one other thing: we understand that filters can be replaced with Coherence Query Language statements. Can you replace the Filter with a parameterized query which takes a single argument, for example, the state code for the select?

Kind regards
Veronica Key
Retail Is Us
V.Key @retail-is-us.com

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Start Eclipse and open the provided project.
2	Complete the partial implementation of the **`StringAggregator`** class.
3	Modify the **`TestAggregator`** class to use **`QueryHelper`** and the new aggregator.
4	Test using the provided `TestMapTrigger` class and run configuration.
5	Analyze the execution results.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Hints

Aggregators are conceptually simple, but difficult to develop correctly due to their lifecycle and use. Things to remember when developing an `Aggregator` include:

- Most aggregators will extend `AbstractAggregator`.
- Always implement a `ValueExtractor` based constructor.
- The `process` method is called with either the roll up of an aggregator (`isFinal == true`) or to aggregate individual data elements (`isFinal == false`).
- The `finalizeResult` and `init` methods also either rollup or aggregate.

Aggregators are registered using one of the `aggregate` methods inherited from the `InvocableMap` interface. For example:

- Object InvocableMap.aggregate(Filter filter, InvocableMap.EntryAggregator aggregator);

The `QueryHelper` class can be used to replace a `Filter` with the `WHERE` clause of a Coherence QL `SELECT` statement. Method of `QueryHelper` include:

- `Filter createFilter(String s)` – create a filter from a simple CohQL where clause. Remember the '`SELECT * FROM 'cachename' WHERE`' portion is implied.
- `Filter createFilter(String s, Object[] aBindings)` – create a filter from a CohQL where clause, using numeric `?#` style bindings
- `Filter createFilter(String s, Map mapBindings)` – create a filter from a CohQL where clause, using `:name` style bindings

Detailed Instructions

Start Eclipse and open the provided project	
6-2.1	Within Eclipse, expand project **Practice.06.02** and expand the **src** folder to display its contents.
Complete the partial implementation of the StringAggregator **class**	
6-2.2	Navigate to the **`retail.aggregator`** package and open the **`StringAggregator.java`** class.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

6-2.3 Locate the text "TODO: Must extend AbstractAggregator" and modify the class to extend `AbstractAggregator`. The modified code should resemble:

```
public class StringAggregator extends AbstractAggregator
. . .
```

6-2.4 The class will now show errors, which will be eliminated, one by one.
Find the TODO referring the constructor, and beneath it add a constructor which takes a `ValueExtractor` and passes it to super. The modified code should resemble:

```
public StringAggregator(ValueExtractor extractor) {
     super(extractor);
   }
```

6-2.5 Find the "TODO: Implement the init method," and beneath it implement the **`init`** method such that it creates a new **`StringBuffer`** instance and assigns it to the **`results`** variable. The modified code should resemble:

```
protected void init(boolean isFinal) {
     results = new StringBuffer();
   }
```

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

6-2.6 Find the “TODO: Implement the process method,” and beneath it create an implementation. The implementation should check the length of the string buffer and append a comma then the provided value, as a string. Note that there are any number of possibly ways to implement such a function. Consider that partial roll ups might return the **`StringBuffer`** itself rather than a **`String`**. How would that change the implementation?

The implementation should resemble:

```
protected void process(Object object, boolean isFinal) {
    if (object == null ) return;
    if (!isFinal) {
        if (results.length() == 0) {
           results.append((String)object);
        } else {
            results.append(",");
            results.append((String)object);
        }
    } else { // isFinal
      // basically the same thing
      // but with an entire set of values
      if (results.length()>0) {
         results.append(",");
       }
       results.append((String)object);
    }
}
```

Note that in directory **`Resources\Practice.06.02`** is a code snippet file which can be cut and pasted to minimize typing.

6-2.7 Find the “TODO: Complete the finalizeResults” method and add an implementation which results in null if string buffer is empty, or a String version otherwise.

The implementation should resemble:

```
protected Object finalizeResult(boolean arg0) {
    return results.length() > 0? results.toString():null;
}
```

6-2.8 Save your changes.

Modify the `TestAggregator` class to use `QueryHelper` and the new aggregator

6-2.9 In the **`retail.test`** package, open the class **`TestAggregator.java`**.

6-2.10 Find the TODO resembling: “`// TODO: Replace the state filter with...`”

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

6-2.11 Replace the Filter:

```
Filter stateFilter = EqualsFilter("getAddress.getState","CA");
```

with a `QueryHelper` statement similar to:

```
stateFilter = QueryHelper.createFilter("address.state=?1",
                                   new Object[]{ new String("CA")} );
```

6-2.12 Find the TODO resembling: "`// TODO: aggregate the customer cache using…`"

6-2.13 Aggregators are registered against the aggregate method of a cache with or without a filter.
Creating an instance of **`StringAggrgator`** using a **`ChainedExtractor`** as constructor and specifying the the **`getAddress`** and **`getZip`** methods. The code should resemble:

```
StringAggregator ag =
  new StringAggregator(new ChainedExtractor("getAddress.getZip"));
Object result = customerCache.aggregate(stateFilter,ag);
```

6-2.14 Save the changes.

Test using the Practice 06.02 run configuration

6-2.15 Make sure project **Practice.06.02** is selected, or its solution if testing the solution.

- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the Eclipse.
- If **Practice06.02[.solution]** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice06.02 [.solution]** in the left-hand window.

Analyze execution results

6-2.16 Examine the console window, scrolling to the bottom if required. Look for a statement similar to : "`Aggregator returned '94121,90501'.`"

6-2.17 Why are only two zip codes returned? Change the state filter to specify MA and rerun the test, what happens?

6-2.18 Rerun the test. Did you see the expected result?

Solution Instructions

To run the solution, start Eclipse and connect to the **`D:\Oracle\student\practices`** workspace and open the **Practice06.02.solution**. Follow the directions for testing the application from steps 6-2.15, ignoring any steps to add code, all of which is in the solution.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 7

Chapter 8

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 7

Practices Overview

Developers must know how to implement and execute the various types of grid processing paradigms offered by Coherence.

The objective of this practice is to assist the developer with processing data in parallel across the Coherence grid.

In this practice, you will perform the following tasks:

- Implement, run, and review an EntryProcessor.
- Implement, run, and review an InvocationService.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 7-1: Implement, run, and review an EntryProcessor

Skills Learned

At the end of this practice, you should be able to:

- Write code for an `EntryProcessor` agent that updates orders in the cache with a 10% discount.
- Write code for the query filter to specify the entries for orders that total over $999.99.
- Using the filter and the invocable API, write code to invoke the `EntryProcessor` against filtered items to apply a 10% discount.
- Start a Coherence two-node cluster that uses a partitioned cache to store the data.
- Run code that loads the cache with the sample retail domain model data, and executes the `EntryProcessor` across the cluster.
- Review and explain where data is stored, and explain how the `EntryProcessor` behaves.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Problem Statement

From:	bill.iards@retail-is-us.com
Sent:	Friday, July 15th, 12:02pm
To:	you@acmedev.com
Subject:	Coherence Grid Processing

Hello!

The Coherence clustered cache for retail-is-us.com contains many orders for our customers. We would like to run a promotion where we automatically offer a 10% discount to our customers for each order over $999.99.

The problem we are facing is that our cache contains many customers, and each customer may have placed several orders. The number of orders in the cache that exceed $999.99 is numerous, and the associated data would be very large. Repeatedly passing this large amount of data over the network would tax our corporate bandwidth, not to mention the processing cost of constantly deserializing numerous cache entries. We would also like the application to take advantage of the CPU processing power available across the Coherence grid.

I understand that Coherence EntryProcessors:

1. Work with queries to locate the cache entries in the cluster
2. Serialize and move the processing class to the cache entries where they are located
3. Are processed in parallel across the cluster
4. Provide lock-free processing for high performance

With this in mind, we need you to create a Coherence EntryProcessor that can perform this task. Our engineering staff has provided a baseline Eclipse project which hopefully can be adapted to serve this purpose.

We look forward to seeing the example you create.

Regards,

Bill Iards
Retail Is Us
bill.iards@retail-is-us.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Design

In this practice, you write code that overrides the `AbstractProcessor`'s `process()` method in a class called `DiscountEntryProcessor`. In the main application class, `CacheTester`, you then write code that creates a Coherence filter using the Coherence Query Language. The query will return all cache entries that have an `orderTotal` greater than 999.99. Finally, you write code that invokes the `DiscountEntryProcessor` across the cluster using the `orders.invokeAll()` method. The `filter` and `DiscountEntryProcessor` are passed in as arguments to this call. The filter is used to perform the query to determine which cache entries in the cluster to send the `DiscountEntryProcessor` code to for processing.

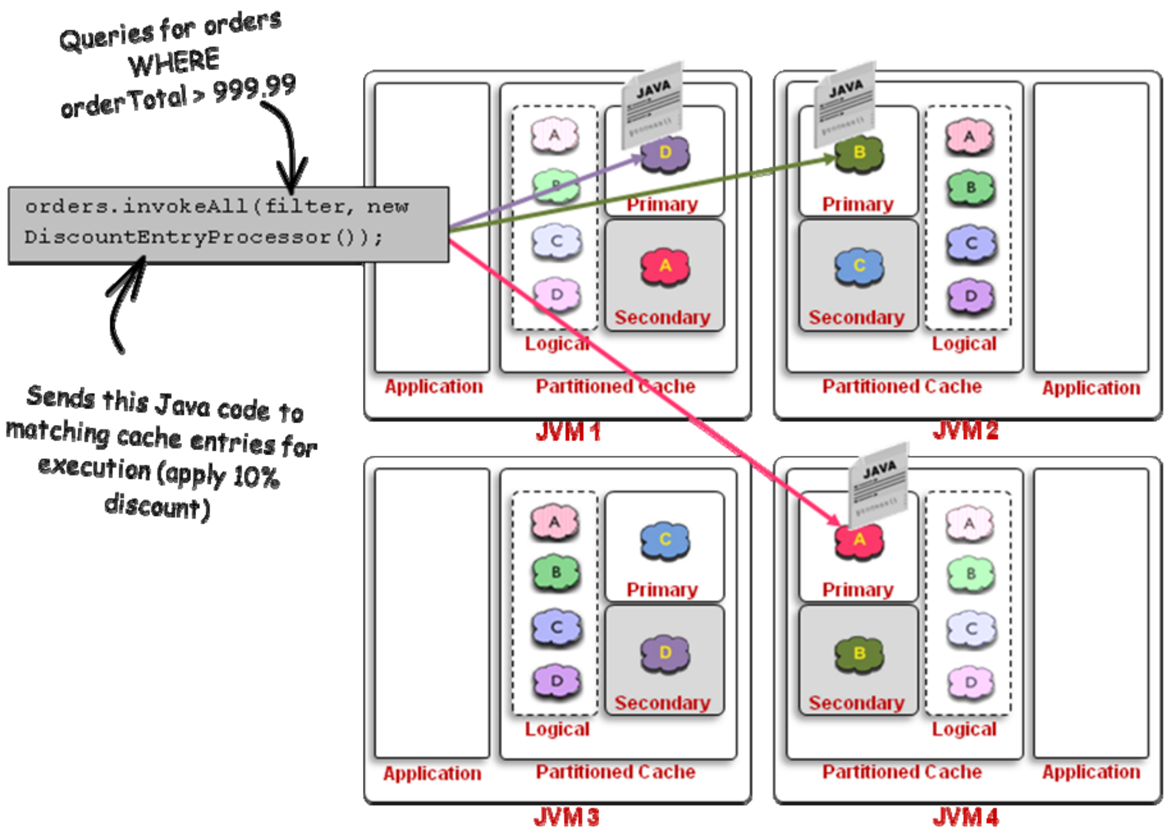

Queries for orders WHERE orderTotal > 999.99
orders.invokeAll(filter, new DiscountEntryProcessor());
Sends this Java code to matching cache entries for execution (apply 10% discount)
JAVA
A
B
C
D
Primary
Secondary
Logical
Application
Partitioned Cache
JVM 1
JVM 2
JVM 3
JVM 4

Figure 1 EntryProcessor Invocation

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\`**`resources`**`\`	Contains a set of practice specific resource directories
Support	`D:\oracle\student\`**`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\`**`software`**	Contains installable Coherence and other files

Table 7.1

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Write and execute code that implements an EntryProcessor.
3	Analyze execution results.

Hints:

- DiscountEntryProcessor extends com.tangosol.util.processor.AbstractProcessor.
- EntryProcessors can be invoked against a NamedCache instance using a filter to select which entries to execute code against when using the invokeAll() method.
- The Coherence Query Language can be used to create the filter.
- The com.tangosol.util.InvocableMap.Entry class represents the cache entry the EntryProcessor is executing against.
- The entry.getValue() method returns the cached application object (an order for example).
- The entry.setValue()method causes the cache entry to update.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to make. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Detailed Instructions

Write and execute code that implements an EntryProcessor

7-1.1	Within Eclipse, expand project Practice.07.01 and expand the src folder to display its contents.
7-1.2	Open the `DiscountEntryProcess.java` file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

7-1.3 Locate the text: "TODO: Place EntryProcessor code here," and replace with the code for a Coherence `EntryProcessor` that extends the `AbstractProcessor` class.

Note: For convenience, the code snippets for this exercise can be found in the resources subdirectory for **Practice.07.01** in the **`code.snippet.txt`** file. Your code should resemble:

```
public class DiscountEntryProcessor extends AbstractProcessor {
  private static final long serialVersionUID = 1L;

  @Override
  public Object process(Entry entry) {
    Order order = (Order)entry.getValue();

    StringBuffer sb = new StringBuffer();
    sb.append("Order [").append(order.getId()).append("] Changed:");
    sb.append("\torderTotal before=").append(order.getOrderTotal());
    order.setOrderTotal((float)(order.getOrderTotal() * .9));
    entry.setValue(order);
    sb.append("\torderTotal after=").append(order.getOrderTotal());
    System.out.println(sb.toString());

    return null;
  }
}
```

Save your changes.

7-1.4 Open the **`CacheTester.java`** file.

7-1.5 Locate the text: "TODO: Place CohQL query code here," and replace with the code that forms a query for entries with an `orderTotal > $999.99`. Your code should resemble:

```
Filter filter = QueryHelper.createFilter("orderTotal > ?1",
        new Object[] { new Float(999.99) });
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

7-1.6 Locate the text: “TODO: Place code to invoke EntryProcessor here,” and replace with the code that invokes the `DiscountEntryProcessor` code across the cluster only on the objects where the `orderTotal` is greater than $999.99. Your code should resemble:

```
orders.invokeAll(filter, new DiscountEntryProcessor());
```

Save your changes. Now that the code is done, you are ready to run your cluster and test the `EntryProcessor`.

7-1.7 Within the **src** folder, open the **`CacheTester.java`** file and examine its code more closely:

1. The program begins by running the `Loader.load()` method, which creates sample data for the retail domain object model, and stores the resulting objects in the Coherence cache. For more information on how the Loader works, and what processing is performed, see the domain model documentation.
2. Once the sample domain model is loaded, the program gets a handle to the `Orders` cache, and creates a CohQL query to retrieve and print out all the order totals in the cache that are greater than $999.99. While iterating through the list of entries, the corresponding keys for these objects are stored in the `keys Set`. This is because once the discount is applied, the order may no longer be greater than $999.99 and the query will not properly retrieve the object again for comparison after the `EntryProcessor` is invoked.
3. Next, the `EntryProcessor` is invoked on the orders cache using the `invokeAll()` method, passing in the query filter and `DiscountEntryProcessor` agent to execute on the entries in parallel within the cluster.
4. While the `EntryProcessor` is executed, the `DiscountEntryProcessor` code prints out some information about the orders and the changed values, which can be seen on both console windows.
5. After the `EntryProcessor` is finished executing, the code iterates through the list of keys that were captured previously, and performs a Coherence `get()` on each to retrieve the current value to see if the `EntryProcessor` has successfullly updated each entry with the proper discount within the cache.

This lab uses two (or more) cluster nodes to demonstrate how an `EntryProcessor` works across multiple nodes.

The next few steps will instruct you how to run this scenario.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

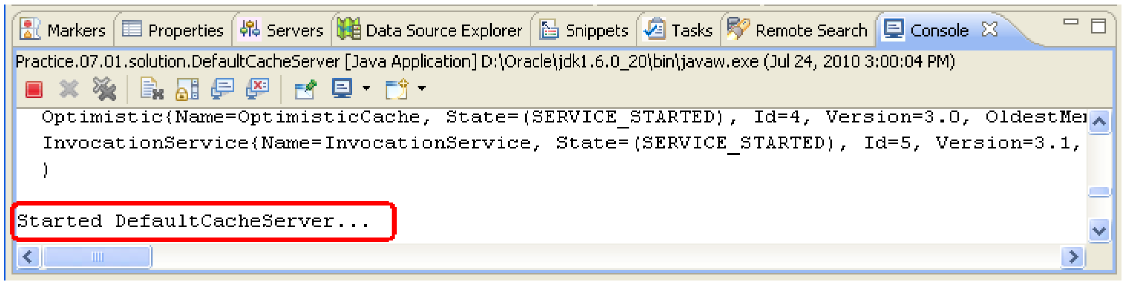
THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

7-1.8 From within Eclipse:

- Make sure the **Practice.07.01** project is selected.
- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the IDE.
- If **Practice.07.01.DefaultCacheServer** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice.07.01.DefaultCacheServer** in the left-hand window.
- To run a cache server instance, select **Run** on the lower right-hand side of the dialog window.

This starts a cache server and displays its output within the Eclipse IDE in the Console tab:



Practice.07.01.solution.DefaultCacheServer [Java Application] D:\Oracle\jdk1.6.0_20\bin\javaw.exe (Jul 24, 2010 3:00:04 PM)

```
  Optimistic(Name=OptimisticCache, State=(SERVICE_STARTED), Id=4, Version=3.0, OldestMe
  InvocationService(Name=InvocationService, State=(SERVICE_STARTED), Id=5, Version=3.1,
  )

Started DefaultCacheServer...
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

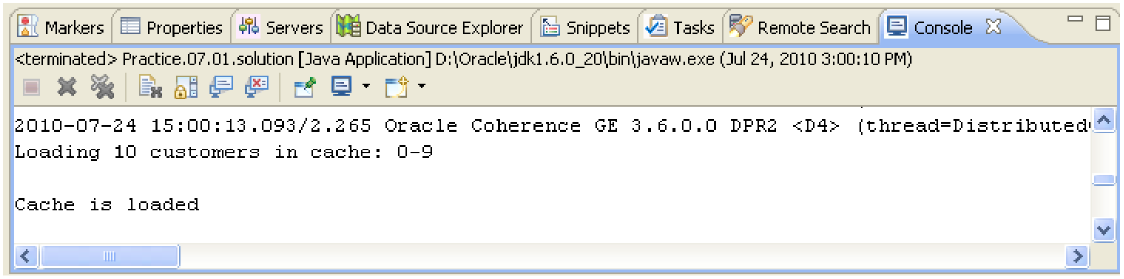
Oracle University and ORACLE CORPORATION use only

7-1.9 To run the `EntryProcessor` across the cluster:

Within Eclipse:

- Make sure the **Practice.07.01** project is selected.
- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the IDE.
- If **Practice.07.01** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice.07.01** in the left-hand window.
- To run a cache server instance, select **Run** on the lower right-hand side of the dialog window.

This starts the `CacheTester` application and displays its output within the Eclipse IDE in the Console tab:



```
<terminated> Practice.07.01.solution [Java Application] D:\Oracle\jdk1.6.0_20\bin\javaw.exe (Jul 24, 2010 3:00:10 PM)
2010-07-24 15:00:13.093/2.265 Oracle Coherence GE 3.6.0.0 DPR2 <D4> (thread=Distributed
Loading 10 customers in cache: 0-9

Cache is loaded
```

HINT: Use the Console tab's Display Selected Console icon to quickly switch between command windows.

After the `CacheTester` application runs, you should see output similar to the following in each console window:

CacheTester window:

```
. . .
Loading 10 customers in cache: 0-9

Cache is loaded

Orders over $999.99 before EntryProcessor
Order: 1 orderTotal=3827.97
Order: 13 orderTotal=7036.45
Order: 14 orderTotal=3238.96
Order: 2 orderTotal=1050.0
Order: 8 orderTotal=2551.98
Order: 10 orderTotal=5231.94
Order: 5 orderTotal=9047.93

Invoking EntryProcessor
Order [1] Changed:     orderTotal before=3827.97     orderTotal
after=3445.1729
Order [14] Changed:    orderTotal before=3238.96     orderTotal
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

```
after=2915.064
Order [2] Changed:      orderTotal before=1050.0      orderTotal
after=945.0
Order [5] Changed:      orderTotal before=9047.93     orderTotal
after=8143.1367

EntryProcessor invoked

The same orders after EntryProcessor
Order: 1 orderTotal=3445.1729
Order: 2 orderTotal=945.0
Order: 5 orderTotal=8143.1367
Order: 8 orderTotal=2296.782
Order: 10 orderTotal=4708.746
Order: 13 orderTotal=6332.805
Order: 14 orderTotal=2915.064
. . .
```

DefaultCacheServer window:

```
. . .
Started DefaultCacheServer...
. . .
Order [13] Changed:     orderTotal before=7036.45     orderTotal
after=6332.805
Order [8] Changed:      orderTotal before=2551.98     orderTotal
after=2296.782
Order [10] Changed:     orderTotal before=5231.94     orderTotal
after=4708.746
. . .
```

You have successfully written and executed an `EntryProcessor` on a Coherence cluster. Are you sure? How do you know?
The next section explores these questions with you.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze execution results

7-1.10 Take a closer look at the output in the `CacheTester` window. The first thing you should notice is that prior to executing the `EntryProcessor`, the code prints out all of the orders that total more than $999.99. The printout includes the ID of the order entry, and the total amount of the order.

```
Orders over $999.99 before EntryProcessor
Order: 1 orderTotal=3827.97
Order: 13 orderTotal=7036.45
Order: 14 orderTotal=3238.96
Order: 2 orderTotal=1050.0
Order: 8 orderTotal=2551.98
Order: 10 orderTotal=5231.94
Order: 5 orderTotal=9047.93
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

7-1.11 The next thing the code prints out is that it is executing the `EntryProcessor`, then the printout displays that orders are getting changed. You should see output similar to the following:

CacheTester window:

```
Invoking EntryProcessor
Order [1] Changed:      orderTotal before=3827.97     orderTotal
after=3445.1729
Order [14] Changed:     orderTotal before=3238.96     orderTotal
after=2915.064
Order [2] Changed:      orderTotal before=1050.0      orderTotal
after=945.0
Order [5] Changed:      orderTotal before=9047.93     orderTotal
after=8143.1367

EntryProcessor invoked
```

DefaultCacheServer window:

```
Order [13] Changed:     orderTotal before=7036.45     orderTotal
after=6332.805
Order [8] Changed:      orderTotal before=2551.98     orderTotal
after=2296.782
Order [10] Changed:     orderTotal before=5231.94     orderTotal
after=4708.746
```

The display shows the ID of the order that is getting changed, the total of the order before the discount is applied, and the total of the order after the discount is applied. Because there are two or more Coherence cluster nodes running, the order cache entries are located across each node. As a result, the printout showing the order entries that changed can be displayed in each console window where a Coherence node is running. This shows that the `EntryProcessor` code ran on each node, as the printout is displayed on each node's standard out.

Match the changed order ID numbers with the IDs in the previous list before the `EntryProcessor` was executed to verify that the order entries are the same.

The output also indicates that the `EntryProcessor` is finished executing.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

7-1.12 After the `EntryProcessor` is done executing, the code checks to see if the `EntryProcessor` successfullly updated the entries within the cache itself. The code saved the ID keys of the original orders that exceeded $999.99, performs a `get()` on each key, and displays the current order total for each entry. You should see output similar to the following:

```
The same orders after EntryProcessor
Order: 1 orderTotal=3445.1729
Order: 2 orderTotal=945.0
Order: 5 orderTotal=8143.1367
Order: 8 orderTotal=2296.782
Order: 10 orderTotal=4708.746
Order: 13 orderTotal=6332.805
Order: 14 orderTotal=2915.064
```

Match the order IDs to the output that shows the entries as they were changed, and verify that the current order total matches the discounted order total. If they match, then your `EntryProcessor` successfully updated the orders in the cache.

Now that you have run a Coherence `EntryProcessor`, and verified its behavior, take a look at working with Invocation Services.

Note: Close all command windows before proceeding. You can do this using the Terminate icon 🟥 in the Eclipse Console tab.

Solution Instructions

To use the solution, start Eclipse and connect to the `D:\Oracle\student\practices` workspace and open **Practice.07.01.solution**. Follow the directions for testing the application from steps **7-1.7**, ignoring any steps to add code, all of which is in the solution.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 7-2: Implement, run, and review an InvocationService

Skills Learned

At the end of this practice, you should be able to:

- Write code for an `InvocationService` agent that checks the amount of available free memory on all Coherence cluster nodes.
- Write code that invokes the `InvocationService`, both synchronously and asynchronously, using the `InvocationService` API.
- Start a Coherence two-node cluster.
- Run code that executes the `InvocationService` across the cluster.
- Review and explain how `InvocationService` behaves.

Problem Statement

From:	billi.onair@retail-is-us.com
Sent:	Wednesday, July 15th 7:02pm
To:	you@acmedev.com
Subject:	Coherence Invocation Service

We would like to develop some administrative tools that make use of the Coherence clustered Invocation Service for our retail-is-us.com application.

We have not developed an Invocation Service before, so we would like you to develop a simple example that checks the available free memory on each JVM across the cluster. We can take this example and expand its functionality on our own.

I understand that Coherence Invocation Services:

1. Do not deal with cache entries in the cluster
2. Serialize and move the processing class to the cluster nodes for remote processing
3. Are processed in parallel across the cluster

Our engineering staff has provided a baseline Eclipse project which hopefully can be adapted to serve this purpose.

We look forward to seeing the example you create.

Regards,

Billi Onair
Retail Is Us
billi.onair@retail-is-us.com

Design

In this practice you write code that overrides the `AbstractInvocable`'s `run()` method in a class called `FreeMemoryAgent`, to capture the free memory on Coherence cluster nodes. This practice involves invoking the `InvocationService` with the `FreeMemoryAgent` synchronously and asynchronously. For the asynchronous invocation, you write the `MyInvocationObserver` class that implements the

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

`com.tangosol.net.InvocationObserver` interface to receive notifications when each cluster node has finished executing the service. In the main application class, `CacheTester`, you write code that gets a reference to the `InvocationService`, and invoke it synchronously using the `InvocationService`'s `query()` method. Finally, you write code that invokes the `InvocationService` asynchronously using the `InvocationService`'s `execute()` method.

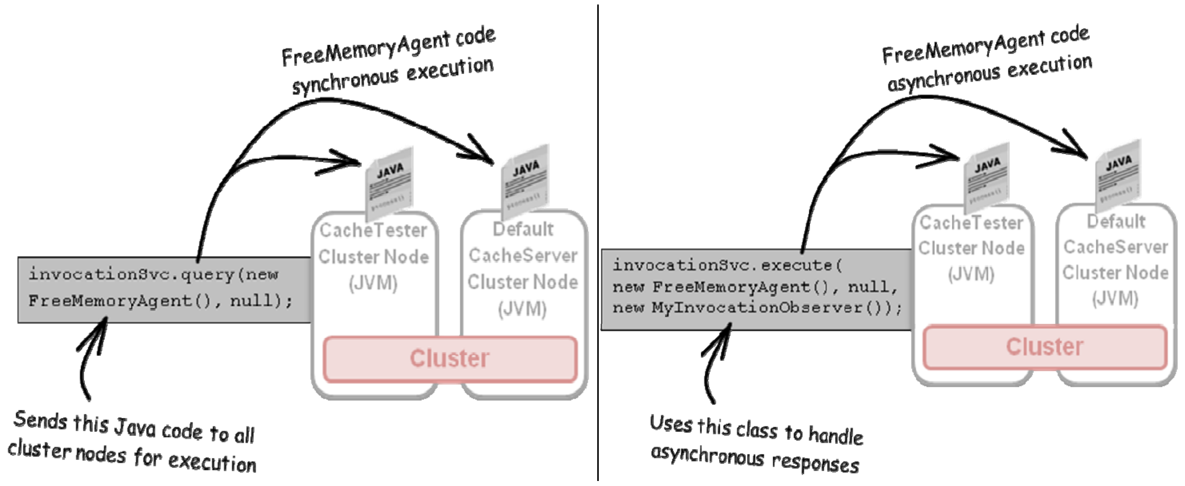

FreeMemoryAgent code synchronous execution
JAVA
JAVA
CacheTester Cluster Node (JVM)
Default CacheServer Cluster Node (JVM)
Cluster
invocationSvc.query(new FreeMemoryAgent(), null);
Sends this Java code to all cluster nodes for execution
FreeMemoryAgent code asynchronous execution
JAVA
JAVA
CacheTester Cluster Node (JVM)
Default CacheServer Cluster Node (JVM)
Cluster
invocationSvc.execute(new FreeMemoryAgent(), null, new MyInvocationObserver());
Uses this class to handle asynchronous responses

Figure 2 InvocationService Invocation

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources\`**	Contains a set of practice specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Table 7.2

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Overview Instructions

1	Write and execute code that implements an InvocationService.
2	Analyze execution results.

Hints:

- FreeMemoryAgent extends com.tangosol.net.AbstractInvocable.
- MyInvocationObserver implements com.tangosol.net.InvocationObserver.
- CacheTester invokes the InvocationService synchronously using is.query() and asynchronously using is.execute().
- MyInvocationObserver is passed to the InvocationService's execute() method for handling asynchronous responses.
- The memberCompleted() method of MyInvocationObserver handles the responses for each completed execution.

Detailed Instructions

Write and execute code that implements an InvocationService	
7-2.1	Within Eclipse, expand project Practice.07.02 and expand the src folder to display its contents.
7-2.2	Open the **`FreeMemoryAgent.java`** file.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

7-2.3	Locate the text: "TODO: Place Invocable agent code here," and replace with the code for a Coherence `InvocationService` agent that extends the `AbstractInvocable` class. This is the code that will get executed across the cluster. **Note:** For convenience the code snippets for this exercise can be found in the resources subdirectory for **Practice.07.02** in the **`code.snippet.txt`** file. Your code should resemble:

```
import com.tangosol.net.AbstractInvocable;

public class FreeMemoryAgent extends AbstractInvocable {
  private static final long serialVersionUID = 1L;

  public void run() {
    Runtime rt = Runtime.getRuntime();
    setResult(rt.freeMemory());
  }
}
```

Save your changes.

7-2.4	Open the **`MyInvocationObserver.java`** file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

7-2.5 Locate the text: "TODO: Place InvocationObserver code here," and replace with the code for a Coherence `InvocationObserver` that implements the `InvocationObserver` interface. This class handles the callback results when executing the `FreeMemoryAgent` asynchronously. Your code should resemble:

```
import com.tangosol.net.InvocationObserver;
import com.tangosol.net.Member;

public class MyInvocationObserver implements InvocationObserver {
  @Override
  public void invocationCompleted() {
    System.out.println(
        "\nMyInvocationObserver processing completed\n");
  }

  @Override
  public void memberCompleted(Member member, Object result) {
    System.out.println(
        "In Async Observer: Member: " + member + " Free: " + result);
  }

  @Override
  public void memberFailed(Member member, Throwable throwable) {}

  @Override
  public void memberLeft(Member member) {}
}
```

Save your changes.

7-2.6 Open the **`CacheTester.java`** file.

7-2.7 Locate the text: "TODO: Place code to get access to the Invocation Service," and replace with the code that returns a reference to the default InvocationService that is configured with Coherence. Your code should resemble:

```
InvocationService is = (InvocationService)
    CacheFactory.getService("InvocationService");
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

7-2.8 Locate the text: "TODO: Place the code that synchronously executes the FreeMemoryAgent class and prints out the results," and replace with the code that synchronously invokes the `FreeMemoryAgent` code across the cluster. Your code should resemble:

```
Map<Member, Integer> freeMemMap =
    is.query(new FreeMemoryAgent(), null);

for (Map.Entry<Member, Integer> freeMem :
    freeMemMap.entrySet()) {
  System.out.println(
      "Member: " + freeMem.getKey() +
      " Free: " + freeMem.getValue());
}
```

7-2.9 Locate the text: "TODO: Place the code here that asynchronously executes the FreeMemoryAgent class uses the MyInvocationObserver class to print out results," and replace with the code that asynchronously invokes the `FreeMemoryAgent` code across the cluster. Your code should resemble:

```
is.execute(new FreeMemoryAgent(), null,
    new MyInvocationObserver());
```

Save your changes. Now that the code is done, you are ready to run your cluster and test the `InvocationService`.

7-2.10 Within the **src** folder, open the **`CacheTester.java`** file and examine its code more closely:

1. The program gets a handle to the `InvocationService`.
2. It executes the `FreeMemoryAgent` class across the cluster synchronously and prints out the results.
3. The program then executes the agent asynchronously using the `MyInvocationObserver` class which handles printing out the results and execution complete message.

This lab uses two (or more) cluster nodes to demonstrate how an `InvocationService` works across multiple nodes.

The next few steps will instruct you how to run this scenario.

7-2.11 From within Eclipse:

- Run the **Practice.07.02.DefaultCacheServer** run configuration to start a `DefaultCacheServer` within the Eclipse IDE in the Console tab.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

7-2.12 To run the `Invocable` agent across the cluster:

Within Eclipse:

- Run the **Practice.07.02** run configuration to start the `CacheTester` application within the Eclipse IDE in the Console tab:

HINT: Use the Console tab's Display Selected Console icon 🖥 to quickly switch between command windows.

After the `CacheTester` application runs, you should see output similar to the following in each console window:

CacheTester window:

```
. . .
Invoking the service synchronously
Member: Member(Id=5, Timestamp=2010-08-05 05:26:01.14,
Address=139.185.35.126:8089, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:2060,
Role=CacheTesterCacheTester) Free: 98591504
Member: Member(Id=1, Timestamp=2010-08-05 05:20:33.781,
Address=139.185.35.126:8088, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:4060,
Role=CoherenceServer) Free: 117415472

Invoking the service asynchronously
Asynchronous execution finished

In Async Observer: Member: Member(Id=5, Timestamp=2010-08-05
05:26:01.14, Address=139.185.35.126:8089, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:2060,
Role=CacheTesterCacheTester) Free: 97875744
In Async Observer: Member: Member(Id=1, Timestamp=2010-08-05
05:20:33.781, Address=139.185.35.126:8088, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:4060,
Role=CoherenceServer) Free: 117415472

MyInvocationObserver processing completed
. . .
```

DefaultCacheServer window:

```
. . .
Started DefaultCacheServer...
. . .
```

Nothing of interest is printed out

You have successfully written and executed an `Invocable` agent on a Coherence cluster. Are you sure? How do you know? The next section explores these questions with you.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze execution results

7-2.13 Take a closer look at the output in the CacheTester window. The first thing you should notice is that it says it is synchronously invoking the `InvocationService`, and the code prints out the results of all the cluster nodes. You should see output similar to the following:

```
Invoking the service synchronously
Member: Member(Id=5, Timestamp=2010-08-05 05:26:01.14,
Address=139.185.35.126:8089, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:2060,
Role=CacheTesterCacheTester) Free: 98591504
Member: Member(Id=1, Timestamp=2010-08-05 05:20:33.781,
Address=139.185.35.126:8088, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:4060,
Role=CoherenceServer) Free: 117415472
```

7-2.14 The next thing the code prints out is that it is invoking the `InvocationService` again, but this time ascynhronously. The next line shows that the invocation was sent off and program execution was returned to the `CacheTester` code. Then the `MyInvocationObserver` code prints out the results when each remote execution has completed. When all invocations are completed, `MyInvocationObserver` is called again and prints out that processing is complete. You should see output similar to the following:

CacheTester window:

```
Invoking the service asynchronously
Asynchronous execution finished

In Async Observer: Member: Member(Id=5, Timestamp=2010-08-05
05:26:01.14, Address=139.185.35.126:8089, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:2060,
Role=CacheTesterCacheTester) Free: 97875744
In Async Observer: Member: Member(Id=1, Timestamp=2010-08-05
05:20:33.781, Address=139.185.35.126:8088, MachineId=4478,
Location=site:us.oracle.com,machine:edRSr26p1,process:4060,
Role=CoherenceServer) Free: 117415472

MyInvocationObserver processing completed
```

Now that you have run a Coherence `InvocationService`, and verified its behavior, take a look at working with CommonJ work managers.

Note: Close all command windows before proceeding. You can do this using the Terminate icon 🟥 in the Eclipse Console tab.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Solution Instructions

To use the solution, start Eclipse and connect to the `D:\Oracle\student\practices` workspace and open **Practice.07.02.solution**. Follow the directions for testing the application from steps **7-2.10**, ignoring any steps to add code, all of which is in the solution.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 8

Chapter 9

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 8

Practices Overview

Developers must be able to configure and use Coherence transactional schemes to demarcate units of work that involve multiple cache operations as a single atomic operation.

The objective of this practice is to assist the developer with using Coherence transaction features.

In this practice, you will perform the following tasks:

- Configure a Coherence transactional scheme.
- Write a transactional Coherence client using the `NamedCache` API.
- Write a transactional Coherence client using the `Connection` API.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 8-1: Configure, Run, and Review Transactional Clients

Skills Learned

At the end of this practice, you should be able to:

- Configure a transactional scheme and map it to a cache.
- Write transactional clients using the `NamedCache` API.
- Write transactional clients using the `Connection` API.
- Start a Coherence node cluster that uses the transactional scheme.
- Identify that Coherence is using the specific configuration.
- Review and explain how transaction connections work, and how to distinguish that a transactional scheme is in use.

Problem Statement

From:	Billi.Onair@retail-is-us.com
Sent:	Thursday, July 16th, 1:27pm
To:	you@acmedev.com
Subject:	Using Coherence Transactions

Hello!

Our application has a requirement to perform multiple cache operations with Coherence as a single unit of work that either wholly succeeds or fails. We would like to use Coherence transaction features to demarcate and control units of work.

As we understand it, the transaction features of Coherence include a transactional scheme, a Transaction Framework, and a Connection API that can provide this capability for our application.

Our engineers have set up a base development project for you to develop some simple examples that demonstrate Coherence transaction features. We look forward to seeing your examples!

Regards,

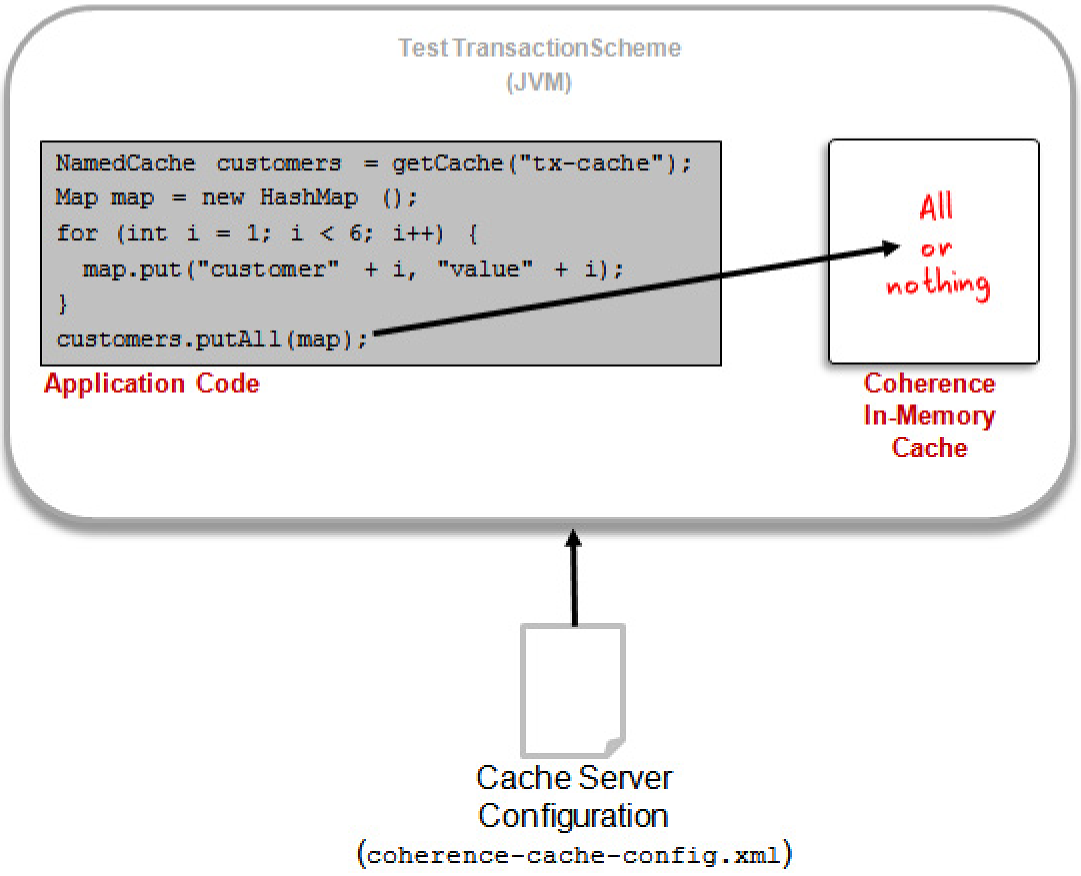
Billi Onair
Retail Is Us
Billi.Onair@retail-is-us.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Design

In this practice, you will add a transactional scheme configuration to a `coherence-cache-config.xml` file. You will use the `<transactional-scheme>` element to configure the transactional scheme, and the `<cache-mapping>` element to map all caches that start with "tx-" to your transactional scheme. Next, you will examine the code of a simple Coherence client that gets a cache called "tx-cache" and performs a `putAll()` to atomically store multiple entries into the cache. Then, you will write code that uses the Connection API to create two connections to the same tx-cache. You will see how the two connections manage and see data within various transactional states. Finally, you will run each of the programs to see how each approach works. The two diagrams below show the architecture of the `NamedCache` API approach and the `Connection` API approach.


TestTransactionScheme
(JVM)
Application Code
All or nothing
Coherence In-Memory Cache
Cache Server Configuration
(coherence-cache-config.xml)

```
NamedCache  customers  = getCache("tx-cache");
Map map  = new HashMap  ();
for  (int  i  = 1;  i  < 6;  i++)  {
  map.put("customer"  + i,  "value"  + i);
}
customers.putAll(map);
```

Figure 1 NamedCache API

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

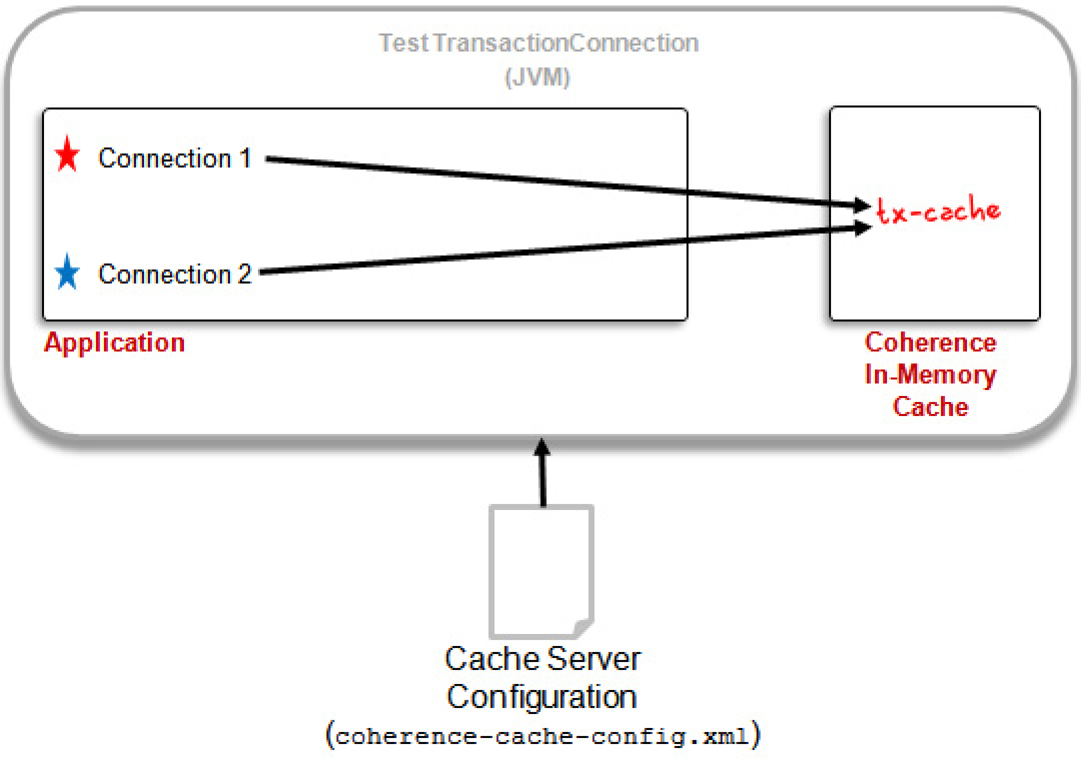

Test TransactionConnection
(JVM)
Connection 1
Connection 2
tx-cache
Application
Coherence In-Memory Cache
Cache Server Configuration
(coherence-cache-config.xml)

Figure 2 Connection API

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources\`**	Contains a set of practice specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Table 8.1

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Configure a transactional scheme.
2	Write a transactional Coherence client using the `NamedCache` API.
3	Write a transactional Coherence client using the `Connection` API.

Hints:

- A transactional scheme is configured using the `<transactional-scheme>` element within the `<caching-schemes>` element.
- A cache is mapped to a particular scheme using the `<cache-name>` and `<scheme-name>` elements within the `<cache-mapping>` element.
- You specify to Coherence which cache configuration file to use by setting the system property `-Dtangosol.coherence.cacheconfig=`*`your_file_name.xml`*.
- The `NamedCache.putAll()` method is atomic when using a transactional cache. Either all entries or none are written to the cache.
- A `Connection` is created using the `DefaultConnectionFactory().createConnection()` method.
- Transactions are started implicitly when there is no current transaction for a `Connection`, and an operation is performed against the cache.
- When using a transactional cache, `cache.insert()` and `cache.update()` are used instead of `cache.put()`.
- Multiple `Connection`s can work with the same cache. For example “tx-cache'. But one `Connection` cannot see data values that are changed by another `Connection` until the `Connection` has committed its changes.
- A `Connection` is closed using the `Connection`'s `close()` method: `conn.close()`.

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to make. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Detailed Instructions

Configure a transactional scheme	
8-1.1	Within Eclipse, expand project **Practice.08.01** and expand the **config** and **src** folders to display their contents.
8-1.2	Open the `coherence-cache-config.xml` file.
8-1.3	Locate the text: “TODO: Place transactional scheme here” within the `<caching-`

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

	`schemes>` element, and replace with the code for a Coherence transactional caching scheme.
	Note: For convenience the code snippets for this exercise can be found in the resources sub-directory for **Practice.08.01** in the **`code.snippet.txt`** file. Your code should resemble:
	
	`<transactional-scheme>`
	  `<scheme-name>transactional</scheme-name>`
	  `<service-name>TestTxnService</service-name>`
	  `<autostart>true</autostart>`
	`</transactional-scheme>`

8-1.4	Locate the text: “TODO: Place cache-mapping here” within the `<caching-scheme-mapping>` element, and replace with the code to map a cache name that starts with tx- to the transactional cache scheme defined in the last step. Your code should resemble:
	
	`<cache-mapping>`
	  `<cache-name>tx-*</cache-name>`
	  `<scheme-name>transactional</scheme-name>`
	`</cache-mapping>`
	
	Save your changes.

Write a transactional Coherence client using the `NamedCache` API	
8-1.5	Within the **src** folder, open the **`TestTransactionScheme.java`** file and examine its code. The code looks exactly the same as other code that you have used before. However, because the cache is configured as a transactional cache, the `putAll()` method is atomic, so either all entries or no entries are written to the cache. Now that the cache is configured, test it by running **`TestTransactionScheme.java`**.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

8-1.6 From within Eclipse:

- Make sure the **Practice.08.01** project is selected.
- Examine the **Practice.08.01.TestTransactionScheme** run configuration.
- Run the **Practice.08.01.TestTransactionScheme** run configuration to see the example work.

You should see output similar to:

```
Oracle Coherence Version 3.6.0.0 Build 17229
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

2010-09-03 12:49:57.296/0.453 Oracle Coherence GE 3.6.0.0 <Info>
(thread=main, member=n/a): Loaded cache configuration from
"file:/D:/Oracle/student/practices/Practice.08.01/config/coherence-
cache-config.xml"

. . .

2010-09-03 12:50:01.171/4.328 Oracle Coherence GE 3.6.0.0 <Info>
(thread=main, member=1): Loaded cache configuration from
"jar:file:/D:/Oracle/coherence/lib/coherence.jar!/internal-txn-cache-
config.xml"
2010-09-03 12:50:01.312/4.469 Oracle Coherence GE 3.6.0.0 <Info>
(thread=main, member=1): Loaded cache configuration from
"jar:file:/D:/Oracle/coherence/lib/coherence.jar!/internal-txn-cache-
config.xml"
2010-09-03 12:50:01.328/4.485 Oracle Coherence GE 3.6.0.0 <D5>
(thread=DistributedCache:TestTxnService, member=1): Service
TestTxnService joined the cluster with senior service member 1
2010-09-03 12:50:01.390/4.547 Oracle Coherence GE 3.6.0.0 <Info>
(thread=TransactionFrameworkThread, member=1): Started version
manager
2010-09-03 12:50:01.531/4.688 Oracle Coherence GE 3.6.0.0 <D5>
(thread=DistributedCache:TestTxnService, member=1): Transactional
Storage High Units: 10M
customer5 = value5
customer4 = value4
customer1 = value1
customer3 = value3
customer2 = value2
```

Examine the output:

- Determine if `coherence-cache-config.xml` from this project was loaded.
- Note the internal transaction cache configuration that was loaded from the `coherence.jar` file.
- Note the `TestTxnService` service that you configured in your file has started.
- There is no output to demonstrate that `putAll()` is atomic.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Write a transactional Coherence client using the `Connection` API

8-1.7 **TestTransactionConnection Design:**

There are three scenarios run by this code:

1. A single connection (**Connection #1**) that inserts two entries, then updates one of the entries from **value** to **value2**, and then **commits** the changes. The diagram below shows the scenario.

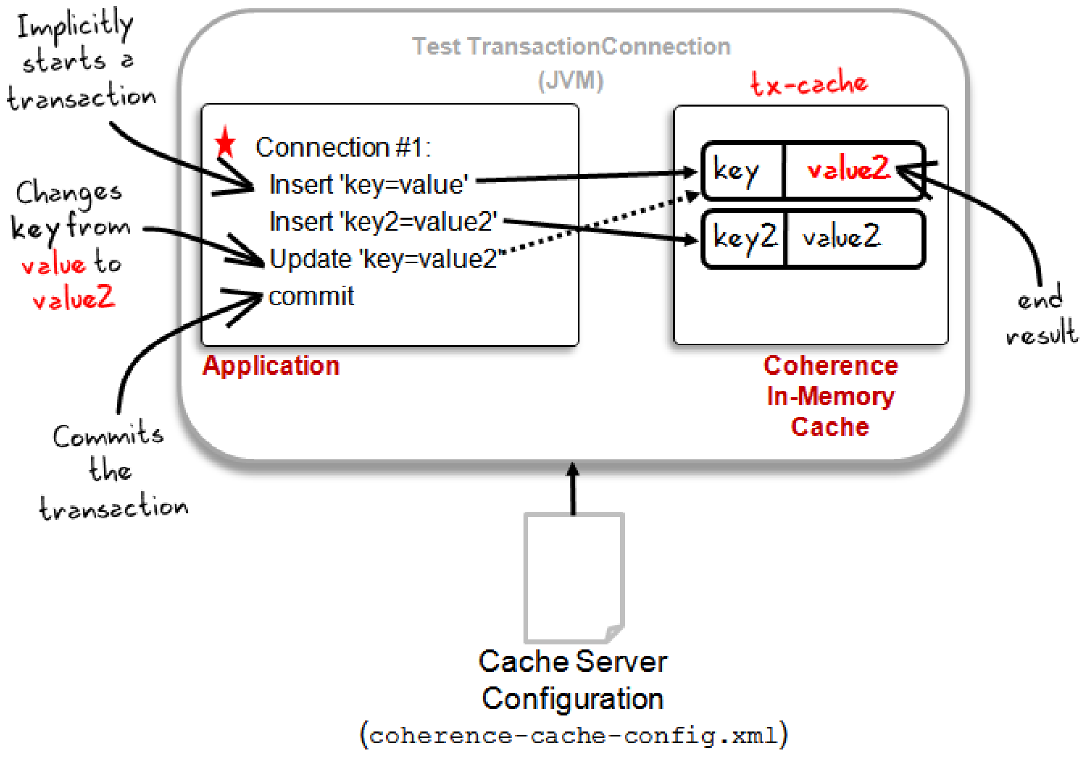

Implicitly starts a transaction
Changes key from value to value2
Commits the transaction
Test TransactionConnection (JVM)
tx-cache
Connection #1:
Insert 'key=value'
Insert 'key2=value2'
Update 'key=value2'
commit
key
value2
key2
value2
end result
Application
Coherence In-Memory Cache
Cache Server Configuration
(coherence-cache-config.xml)

Figure 3 Connection API: Scenario 1

2. A second connection (**Connection #2**) is established that **uses the same tx-cache as Connection #1**. **Connection #1** updates the key entry's value from **value2** to **value3**. Before the change is committed by **Connection #1**, **Connection #2** cannot see the new value. Then **Connection #1 commits** its transaction, and the change becomes visible to **Connection #2**. The diagram below shows the scenario.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

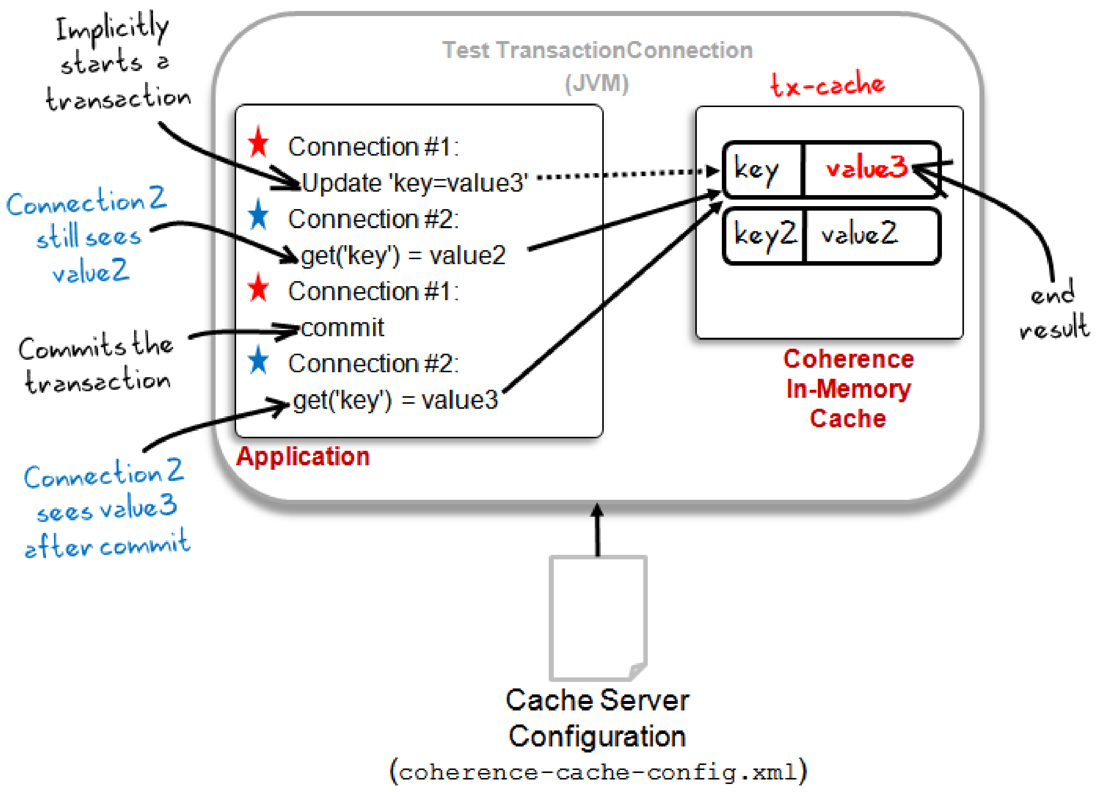

Implicitly starts a transaction
Connection 2 still sees value2
Commits the transaction
Connection 2 sees value3 after commit
Test TransactionConnection (JVM)
tx-cache
Connection #1: Update 'key=value3'
Connection #2: get('key') = value2
Connection #1: commit
Connection #2: get('key') = value3
Application
key
value3
key2
value2
end result
Coherence In-Memory Cache
Cache Server Configuration (coherence-cache-config.xml)

Figure 4 Connection API: Scenario 2

3. **Connection #1** updates the key entry's value again from **value3** to **value4**. Before the change is committed by **Connection #1**, **Connection #2** cannot see the new value. Then **Connection #1** performs a **rollback** on its transaction, leaving the value set to **value3**. **Connection #2** sees the value as **value3** after the transaction is finished. The diagram below shows the scenario.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

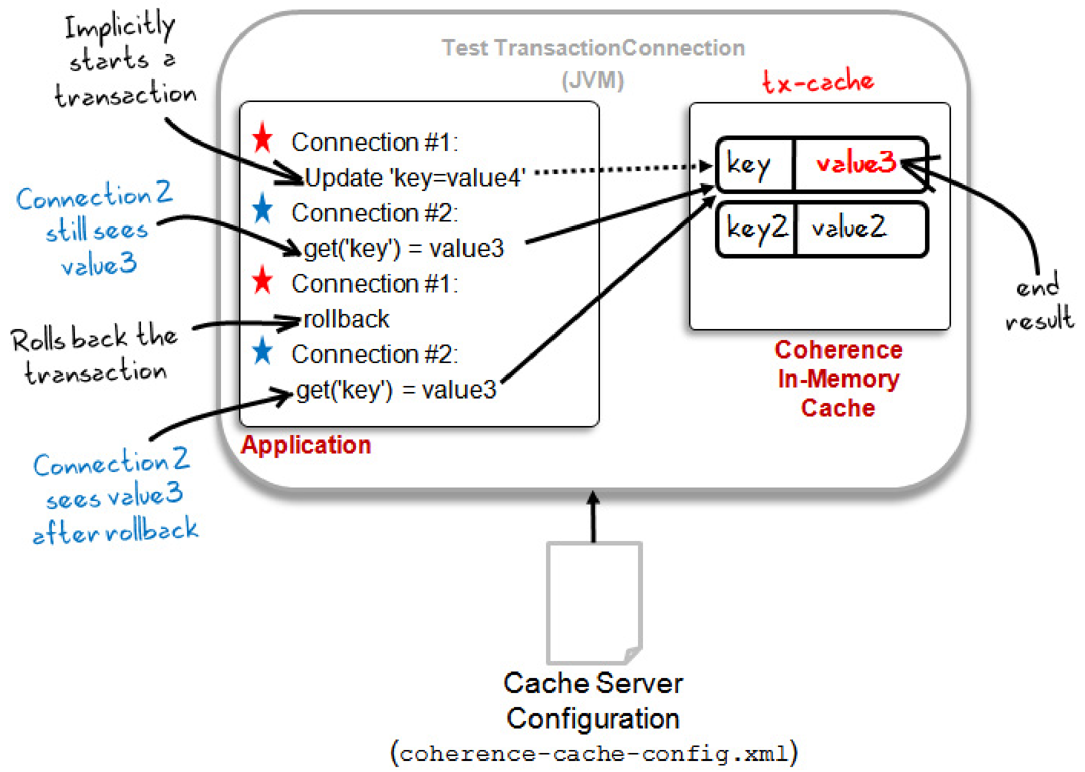

Implicitly starts a transaction
Connection 2 still sees value3
Rolls back the transaction
Connection 2 sees value3 after rollback
Test TransactionConnection (JVM)
tx-cache
Connection #1:
Update 'key=value4'
Connection #2:
get('key') = value3
Connection #1:
rollback
Connection #2:
get('key') = value3
Application
key
value3
key2
value2
end result
Coherence In-Memory Cache
Cache Server Configuration
(coherence-cache-config.xml)

Figure 5 Connection API: Scenario 3

The next few steps will instruct you how to write and run this program.

8-1.8 Open the `TestTransactionConnection.java` file.

8-1.9 Locate the text: "TODO: Place code for connection 1 here," and replace with the code for a Coherence `Connection`.
Note: For convenience the code snippets for this exercise can be found in the resources sub-directory for **Practice.08.01** in the **`code.snippet.txt`** file. Your code should resemble:

```
Connection conn = new
DefaultConnectionFactory().createConnection("TestTxnService");
conn.setAutoCommit(false);
conn.setEager(true);
```

8-1.10 Locate the text: "TODO: Place code to get transactional cache via connection 1 here," and replace with the code that gets a `NamedCache` using the `Connection` opened in the last step. Your code should resemble:

```
OptimisticNamedCache cache = conn.getNamedCache("tx-cache");
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

8-1.11 Locate the text: “TODO: Place code that inserts two entries here,” and replace with the code that inserts two values into the cache using the `NamedCache` reference obtained in the last step. Set the keys and values as follows:

Key	Value
key	value
key2	value2

Note that you use the `insert()` method rather than the `put()` method. Your code should resemble:

```
cache.insert("key", "value");
cache.insert("key2", "value2");
```

8-1.12 Locate the text: “TODO: Update key to value2 on connection 1 here,” and replace with the code that updates the **key** value from **value** to **value2** in the cache using the `NamedCache` reference. Logically set the values as follows:

Key	Value
key	**value2**
key2	value2

Note that you use the `update()` method rather than the `put()` method. Your code should resemble:

```
cache.update("key", "value2", null);
```

8-1.13 Locate the text: “TODO: Commit connection #1 here,” and replace with the code that commits all inserts and updates in the cache using the `Connection` #1 reference obtained earlier. Your code should resemble:

```
conn.commit();
```

8-1.14 Locate the text: “TODO: Place code for connection 2 here,” and replace with the code for a Coherence `Connection`. Your code should resemble:

```
Connection conn2 = new
DefaultConnectionFactory().createConnection("TestTxnService");
conn2.setAutoCommit(false);
```

8-1.15 Locate the text: “TODO: Place code to get transactional cache via connection 2 here,” and replace with the code that gets a `NamedCache` using the `Connection` opened in the last step. Your code should resemble:

```
OptimisticNamedCache cache2 = conn2.getNamedCache("tx-cache");
```

8-1.16 Locate the text: “TODO: Rollback connection #1 here,” and replace with the code that rolls back all changes in the cache using the `Connection` #1 reference obtained earlier. Your code should resemble:

```
conn.rollback();
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

8-1.17 Locate the text: "TODO: Close both connections here," and replace with the code that closes `Connection` #1 and `Connection` #2. Your code should resemble:

```
conn.close();
conn2.close();
```

8-1.18 Now `TestTransactionConnection.java` should be complete. You did not code every single update and commit in the file, as it would have been mundane and repetitive. Please take a few moments and examine the code to see how it matches up with the design for each of the scenarios described above.
Once you are ready, run the program and see how it works.

8-1.19 From within Eclipse:

- Make sure project **Practice.08.01** is selected.
- Examine the **Practice.08.01.TestTransactionConnection** run configuration.
- Run the **Practice.08.01.TestTransactionConnection** run configuration to see the example work.

You should see output similar to:

```
2010-09-03 13:56:04.890/4.515 Oracle Coherence GE 3.6.0.0 <Info>
(thread=main, member=1): Loaded cache configuration from
"jar:file:/D:/Oracle/coherence/lib/coherence.jar!/internal-txn-cache-
config.xml"
2010-09-03 13:56:04.890/4.515 Oracle Coherence GE 3.6.0.0 <D5>
(thread=DistributedCache:TestTxnService, member=1): Service
TestTxnService joined the cluster with senior service member 1
2010-09-03 13:56:04.953/4.578 Oracle Coherence GE 3.6.0.0 <Info>
(thread=TransactionFrameworkThread, member=1): Started version
manager

OPENING CONNECTION #1
Getting tx-cache
2010-09-03 13:56:05.093/4.718 Oracle Coherence GE 3.6.0.0 <D5>
(thread=DistributedCache:TestTxnService, member=1): Transactional
Storage High Units: 10M

INSERTING key AND key2 INTO CACHE (BEGIN TRANSACTION)
UPDATING key value TO value2
Committing transaction (END TRANSACTION)

OPENING CONNECTION #2
Getting same tx-cache that connection #1 is using

VERIFYING THAT CONNNECTION #2 CAN SEE THE SAME DATA INSERTED BY
CONNECTION #1
UPDATING key value2 TO value3 USING CONNECTION #1 (BEGIN TRANSACTION)
Value for key that connection #2 sees=value2
Committing the update on connection #1 (END TRANSACTION)
Value for key that connection #2 sees after commit on connection
#1=value3

TESTING ROLLBACK SCENARIO
UPDATING key value3 TO value4 USING CONNECTION #1 (BEGIN TRANSACTION)
Value for key that connection #2 sees=value3
Rolling back the update on connection #1 (END TRANSACTION)
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

```
Value for key that connection #2 sees after rollback on connection
#1=value3
```

Now that you have learned how to use Coherence transactions, take a look at integrating a data source with Coherence.

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace. and open **Practice.08.01.solution**. Follow the directions for testing the application from steps **8-1.5**, ignoring any steps to add code, all of which is in the solution.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 9

Chapter 10

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 9

Practices Overview

Coherence is, by nature, the natural manager of data for applications. However, situations arise which require some other system to provide data. Such systems might provide data to Coherence once, regularly, or only on demand. Coherence supports a variety of read and read/write scenarios via the `CacheLoader` and `CacheStore` interfaces.

In this practice, you will perform the following tasks:

- Update an existing class to support the full range of `CacheStore` functionality.
- Add support for custom constructors and initialization via `init-params` in configurations.
- Define a custom caching scheme to support a remote data solution.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 9-1: Developer and Registering CacheStores

Skills Learned

At the end of this practice, you should be able to:

- Extend an existing class to support all the methods required by the `CacheStore` Interface.
- Register a cache store declaratively within a configuration file.
- Extend a cache-store configuration to use custom constructor arguments.
- Test and verify that the cache store functions properly.

Problem Statement

From:	Ted.Ious@retail-is-us.com
Sent:	Friday, July 17th, 6:12pm
To:	you@acmedev.com
Subject:	Integrating with a backing store

Good evening,

Veronica mentioned how well you did on some of our other assignments, and we were wondering if you could help with a problem we are having.

We need to be able to use a database as a backing store for several of the caches. We have a baseline Java class which provides core functionality for reading, writing and deleting data from the database, all written using JDBC. We understand that Coherence has a feature that supports read/write access to any backing store via an implementation of some interface. In fact, one of our engineers started a partial implementation but is unavailable and the implementation is broken. Would it be possible for you to complete the implementation, as well as provide an example of how to register the class with Coherence? We would greatly appreciate it if you could!

Regards,

Ted Ious
Retail Is Us
Ted.Ious@retail-is-us.com

PS. We also have heard that Coherence supports custom initialization. Can you also show how to implement some custom initialization? That way we can get the hard coded test values out of the implementation, and into configuration where they belong.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Design

This practice requires two core updates or changes. The first change is completion of a cache store implementation, including a constructor for custom initialization. The second change requires configuration of the cache store such that loading of the Customer cache is managed using the newly created cache store class.

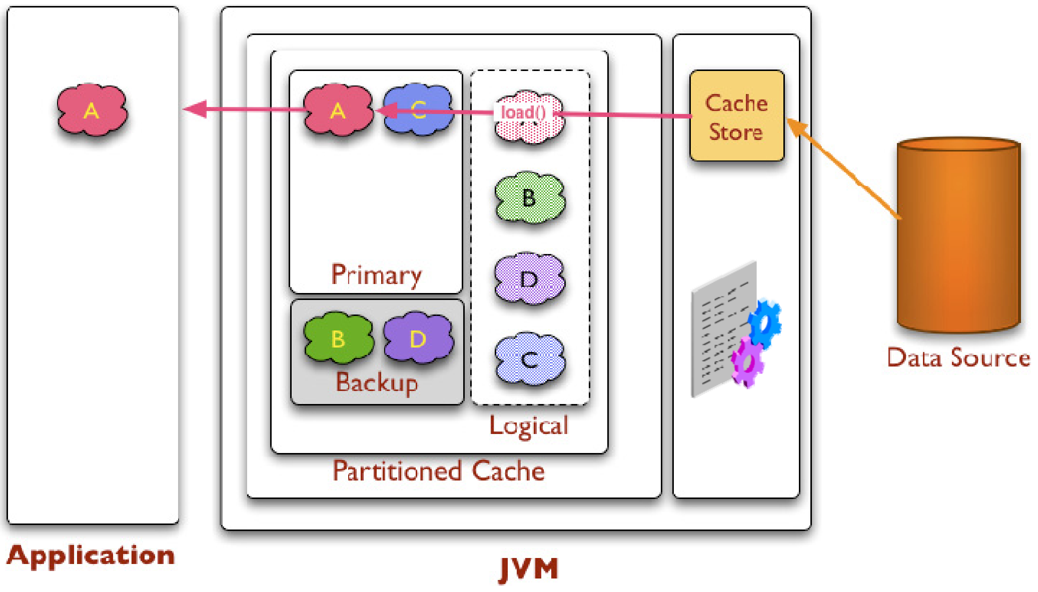


Figure 1 Cache Store Load Diagram

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\resources\`	Contains a set of practice specific resource directories, including code snippets for this practice

Overview Instructions

1	Open Practice.09.01.
2	Complete the **`CustomerCachestore`** class implementation.
3	Modify **`coherence-cache-config.xml`** to use the custom cache store.
4	Test using the Practice.09.01 run configuration.
5	Analyze execution results.

Hints:

- The `CacheStore` interface adds four additional methods on top of the `CacheLoader` interface. These methods are:

```
void store(java.lang.Object oKey, java.lang.Object oValue);
void storeAll(java.util.Map mapEntries);
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

```
void erase(java.lang.Object oKey);
void eraseAll(java.util.Collection colKeys);
```

- It is common to implement `eraseAll` and `storeAll` in a fashion similar to:

```
public void eraseAll(Collection colKeys) {
    for (Object key : colKeys) {
        erase(key);
    }
}
```

  A more efficient implementation might provide an underlying erase by collection mechanism.
- The `CustomerJdbcSupport` class exposes the following methods:
  `public Customer (Long key)`: Return a customer object for the given key
  `saveCustomer(Collection<Customer> customers)`: Insert or update a collection of customers
  `deleteCustomers(Collection<Customer>)`: Delete the customer, if they exist
- A cache name is bound to a cache scheme using a `<cache-mapping>` element within the `<caching-scheme-mapping>` element of `coherence-cache.config.xml`. For example:

```
    <cache-mapping>
         <cache-name>Special</cache-name>
         <scheme-name>SpecialScheme</scheme-name>
    </cache-mapping>
```

  maps a cache named `Special` against a scheme named `SpecialScheme.`
- A caching scheme is defined within the `<caching-schemes>` element of the Coherence cache configuration XML file.
- A custom `CacheStore`, or `CacheLoader`, is defined as part of the `backing-map-scheme` element of a `<*-scheme>` by defining a `<cachestore-scheme>` element.
- A class is defined within a class-scheme using the a class-scheme element similar to:

```
<class-scheme>
      <class-name>
            com. . .ClassImplementingCachestoreOrLoader
      </class-name>
```

- An instantiated class can take custom arguments for its constructor by using a series of `<init-param>` elements within an `<init-params>` element. Each is of the form:

```
<init-param>
      <param-type>java.lang.[Float, . . . ,String]</param-type>
      <param-value>appropriate value</param-value>
</init-param>
```

Remember! Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

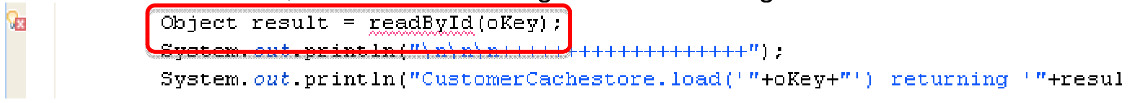
Detailed Instructions

Open Practice.09.01	
9-1.1	Within Eclipse, expand project **Practice.09.01** and expand the **src** folder to display its contents.
Complete the `CustomerCachestore` class implementation	
9-1.2	Navigate to the **`retail.cachestore`** package and open the class **`CustomerCachestore.java`**. **Note:**This class has been partially developed, and supports only `CacheLoader` methods at this time.

9-1.3 C	Find the **load** method, it contains a coding error resembling: (see screenshot below)
	Replace the
	`Object result = readById(oKey);`
	with;
	`Object result = (oKey instanceof Long)?loadCustomer((Long)oKey):null;`
	
	Note that the solution code has a correct implementation.

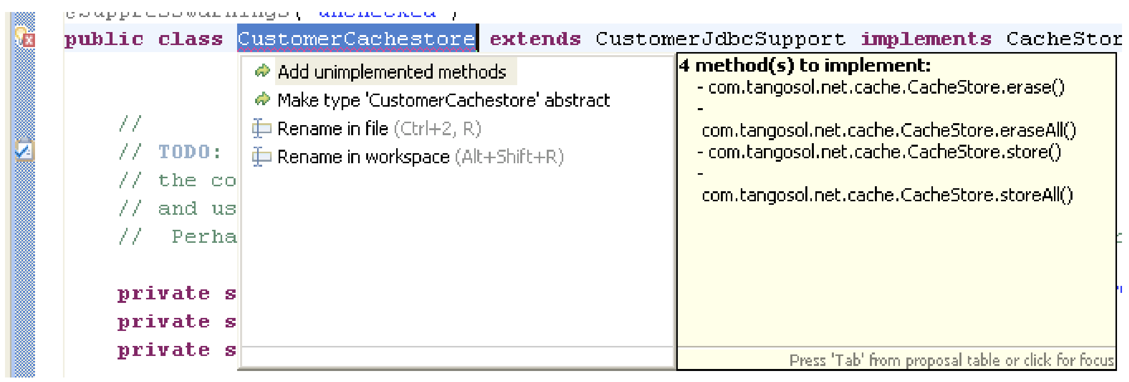
9-1.4	Likewise update the **loadAll** method to use the corrected **load** method, replacing
	`Object value = readById(key);`
	With
	`Object value = load(key);`

9-1.5	Locate: “TODO: Once completed, this class …,” and modify the class to implement the `CacheStore` interface. The modified code should resemble: (see code below)
9-1.6	The class will show errors. Either by hand or using quick fix, add the missing methods.



```
Object result = readById(oKey);
System.out.println("\n\n\n++++++++++++++++++");
System.out.println("CustomerCachestore.load('"+oKey+"') returning '"+resul
```

```
public class CustomerCachestore
     extends CustomerJdbcSupport
          implements CacheStore {
. . .
```



```
public class CustomerCachestore extends CustomerJdbcSupport implements CacheStor
    //
    // TODO:
    // the co
    // and us
    //  Perha
    private s
    private s
    private s
```

Add unimplemented methods
Make type 'CustomerCachestore' abstract
Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)

4 method(s) to implement:
- com.tangosol.net.cache.CacheStore.erase()
- com.tangosol.net.cache.CacheStore.eraseAll()
- com.tangosol.net.cache.CacheStore.store()
- com.tangosol.net.cache.CacheStore.storeAll()

Press 'Tab' from proposal table or click for focus

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

9-1.7	Assuming you used quick fix, the new methods will resemble:

```
public void erase(Object oKey) {
   // TODO Auto-generated method stub
}
public void eraseAll(Collection colKeys) {
   // TODO Auto-generated method stub
}
public void store(Object oKey, Object oValue) {
   // TODO Auto-generated method stub
}
public void storeAll(Map mapEntries) {
   // TODO Auto-generated method stub
}
```

9-1.8 Complete the implementation of the `erase` method. The underlying class has a **`deleteCustomers`** method, which can be used for ease of implementation. The method should resemble:

```
public void erase(Object oKey) {
   System.out.println("\n\n\n+++++++++++++++++++");
   System.out.println("CustomerCachestore.erase('"+oKey+"')
called ");
   System.out.println("+++++++++++++++++++\n\n\n");
   Collection<Long> keys = new LinkedList<Long>()
   if ( oKey instanceof Long)
      keys.add((Long)oKey);
   deleteCustomers(keys);
}
```

Note that the **`student\resources\Practice.09.01`** directory contains a code snippet file, **`code.snippets.methods.txt`**, which contains completed implementations of all required methods.

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

9-1.9 Complete the implementation of the `Store` method. The underlying class has an `insertOrUpdate` method which can be used for ease of implementation.

```
public void store(Object oKey, Object oValue) {
   System.out.println("\n\n\n++++++++++++++++++");
   System.out.println("CustomerCachestore.store('"+oKey+"','"
     +oValue+"') called ");
   System.out.println("++++++++++++++++++\n\n\n");
   Collection<Customer> customers = new LinkedList<Customer>();
   if ( oValue instanceof Customer) {
     customers.add((Customer) oValue);
   }
     saveCustomers(customers);
}
```

9-1.10 Complete the implementation of the `storeAll` method. Your implementation should use the existing `store` method and should resemble:

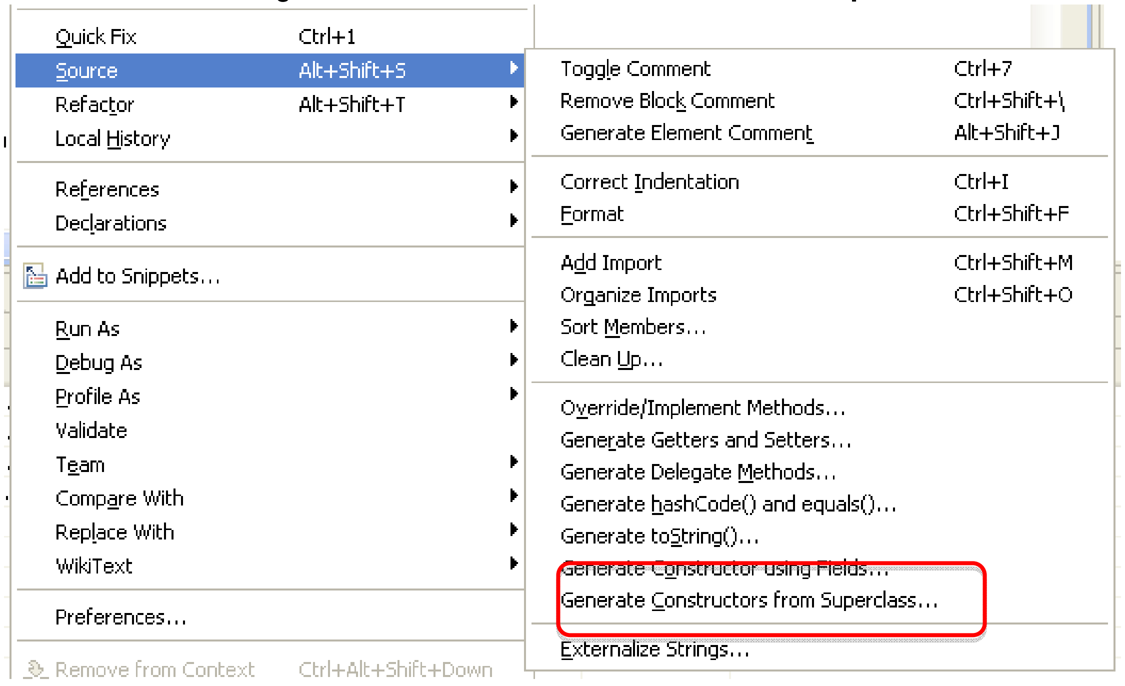
```
public void storeAll(Map mapEntries) {
      Iterator it = mapEntries.entrySet().iterator();
         while (it.hasNext()) {
            Map.Entry pair = (Map.Entry)it.next();
            store(pair.getKey(),pair.getValue());
        }
   }
```

9-1.11 Complete the implementation of the `eraseAll` method. Your implementation should use the existing `erase` method and should resemble:

```
public void eraseAll(Collection colKeys) {
   deleteCustomers(colKeys);
}
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

9-1.12 Add a constructor which takes three arguments, and then calls the underlying constructor with the values for the connection information, username, and password.
You can use the Eclipse source support to insert the constructor by right-clicking inside the class and choosing **Source > Generate Constructors from Superclass**.


Quick Fix Ctrl+1
Source Alt+Shift+S
Refactor Alt+Shift+T
Local History
References
Declarations
Add to Snippets...
Run As
Debug As
Profile As
Validate
Team
Compare With
Replace With
WikiText
Preferences...
Remove from Context Ctrl+Alt+Shift+Down
Toggle Comment Ctrl+7
Remove Block Comment Ctrl+Shift+\
Generate Element Comment Alt+Shift+J
Correct Indentation Ctrl+I
Format Ctrl+Shift+F
Add Import Ctrl+Shift+M
Organize Imports Ctrl+Shift+O
Sort Members...
Clean Up...
Override/Implement Methods...
Generate Getters and Setters...
Generate Delegate Methods...
Generate hashCode() and equals()...
Generate toString()...
Generate Constructor using Fields...
Generate Constructors from Superclass...
Externalize Strings...

9-1.13 When complete, the new constructor should resemble:

```
public CustomerCachestore(String connectionInfo,
                          String uName, String pwd) {
    super(connectionInfo, uName, pwd);
    System.out.println("\n\n\n++++++++++++++++++");
    System.out.println("CustomerCachestore called with '"
        +connectionInfo+"', '"+uName+"', '"+pwd+"'");
    System.out.println("++++++++++++++++++\n\n\n");
}
```

Note that the `student\resources\Practice.09.01` directory contains a code snippet file, `code.snippets.constructor.txt`, which has a completed copy of the required constructor.

9-1.14 Save your work.

Oracle University and ORACLE CORPORATION use only

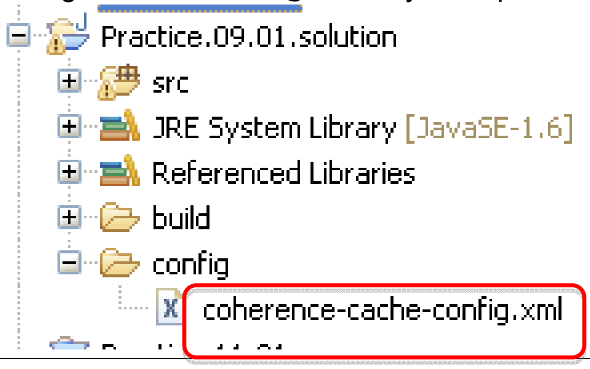
Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Modify `coherence-cache-config.xml` to use the custom cache store.

9-1.15 Navigate to the **`config`** directory and open the `coherence-cache-config.xml` file.



9-1.16 Find the TODO resembling: "TODO: Add cache-mapping element here," and add a new **`cache-mapping`** for the Customer cache.
When complete, the new mapping should resemble:

```
    . . .
    <cache-mapping>
      <cache-name>Customers</cache-name>
      <scheme-name>CustomCacheStoreScheme</scheme-name>
    </cache-mapping>

</caching-scheme-mapping>
. . .
```

Note that the mapping maps to a scheme that has not been added yet.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

9-1.17 Find the TODO resembling: "TODO: Add new cache store scheme element here," and add a new distributed-scheme for the cache store class. The completed scheme should resemble:

```
<distributed-scheme>
      <scheme-name>CustomCacheStoreScheme</scheme-name>
      <service-name>CustomCacheStoreSchemeCache</service-name>
      <backing-map-scheme>
        <read-write-backing-map-scheme>
          <internal-cache-scheme>
            <local-scheme/>
          </internal-cache-scheme>
          <!-- Define the cache store scheme -->
          <cachestore-scheme>
            <class-scheme>
              <class-name>
                retail.cachestore.CustomerCachestore
              </class-name>
              <init-params>
               <init-param>
                  <param-type>java.lang.String</param-type>
                  <param-value>jdbc:oracle:thin:@localhost:1521:XE
                              </param-value>
                </init-param>
                <init-param>
                  <param-type>java.lang.String</param-type>
                  <param-value>COHERENCE</param-value>
                </init-param>
                 <init-param>
                  <param-type>java.lang.String</param-type>
                  <param-value>COHERENCE</param-value>
                </init-param>
              </init-params>
            </class-scheme>
          </cachestore-scheme>
        </read-write-backing-map-scheme>
      </backing-map-scheme>
    </distributed-scheme>
```

Note that the entire element can be found in the `resources` directory as **`code.snippet.cache-scheme.txt`**. Since this element is rather long, it is suggested that you cut and paste the XML.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

9-1.18 Save your changes.

Test using the Practice.09.01 run configuration

9-1.19 Navigate to the `retail.test` package and open the **`TestCacheStore`** class. Examine the class, noticing that no specialized code exists for loading the cache store.

9-1.20 Make sure project **Practice.09.01** is selected, or its solution if testing the solution.

- Click the **arrow** next to the icon for run configurations in the upper left-hand side of the Eclipse.
- If **Practice.06901[.solution]** is not showing, then select **Run Configurations…**
- On the **Run Configurations** screen, select **Java Application** > **Practice.09.01 [.solution]** in the left-hand window.

This starts the test and displays its results in the console window.

Analyze execution results

9-1.21 Examine the console results closely. Notice that the console has many statements similar to:

```
+++++++++++++++++++
CustomerCachestore.store('10','Customer:
    ID:10
    Name:Stephanie Kramer
    Address:      Address:
      Street:100 Remsen Ave
      City:Columbia
      State:MD
      Zip:21044

    Credit Card IDs:[]
    Order IDs:[]
') called
+++++++++++++++++++

MERGE INTO customer c USING (SELECT ? id, ? name, ? street, ? city, ?
state, ? zip FROM dual) c_new ON (c.id = c_new.id)WHEN MATCHED THEN
UPDATE SET c.name = c_new.name,   c.street = c_new.street, c.city =
c_new.city,   c.state = c_new.state, c.zip = c_new.zip WHEN NOT
MATCHED THEN INSERT (c.id, c.name, c.street, c.city, c.state, c.zip)
VALUES (c_new.id, c_new.name, c_new.street, c_new.city, c_new.state,
c_new.zip)
1
```

Each set of statements represents an `INSERT` to the underlying **`customer`** table in the database, and shows that the cache store was being called for each store operation performed by the **`Loader.load()`** method.

9-1.22 Close any open projects and editor windows, and proceed to the next exercise.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Solution Instructions

To exercise the solution, start Eclipse and connect to the **D:\Oracle\student\practices** workspace. Navigate **File** > **Import** > **General > Existing Projects into Workspace**. Click **Next** and browse to the `D:\Oracle\student\practices` directory and import **Practice09.01.solution**.

Follow the directions for testing the application from steps 9-1.17.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 10

Chapter 11

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 10

Practices Overview

Coherence supports a number of cache types, event models, various backing data source strategies, and combinations of each. Choosing the optimum combination can be difficult, and not always obvious. In addition, Coherence's cache-aside, write-aside, write-behind, and write-through caching strategies make architecture and application design even more challenging.

In this practice, you will perform the following tasks:

- Examine a set of predefined application domain scenarios.
- Review each in terms of a proposed Coherence architecture.
- Critically review, compare, and contrast proposed architectures in terms of performance, reliability, data lifecycle, and other criteria.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 10-1: Examining Sample Topologies

Skills Learned

At the end of this practice, you should be able to:

- Review a proposed application scenario and describe its component elements.
- Examine a given architecture, and compare/contrast design decisions in terms of performance, reliability, and data size.
- Enumerate a list of advantages and disadvantages with respect to a given architecture.

Problem Statement

From:	Ted.lous@retail-is-us.com
Sent:	Friday, July 17th, 9:00pm
To:	you@acmedev.com
Subject:	Examining Potential Architectures

Good evening,

We have a number of applications, and we have been sketching proposed architectures. As the resident expert, we were hoping you could examine each, and comment on the pros and cons. Of course, there is no right answer as everything is a tradeoff, but we had hoped you could provide some insight that can help us make the best architectural decisions using Coherence. Each example is presented with a description of each component, as well as an overall architecture diagram.

Regards,

Ted Ious
Retail Is Us
Ted.lous@retail-is-us.com

Oracle University and ORACLE CORPORATION use only

Design

Three Coherence-based application designs are provided, along with a set of requirements. Examine each design, and provide an analysis of the advantages and disadvantages of each implementation component.

Overview Instructions

1	Review the Financial Application Design.
2	Review the Online Music Store Design.
3	Review the Airline Booking Services Design.

Detailed Instructions

Review the Financial Application Design	
10-1.1	Review the design individually or in small groups.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

10-1.2	Explain the reasoning, which may or may not be sound, behind each component, and list advantages and disadvantages of each selected Coherence topology.

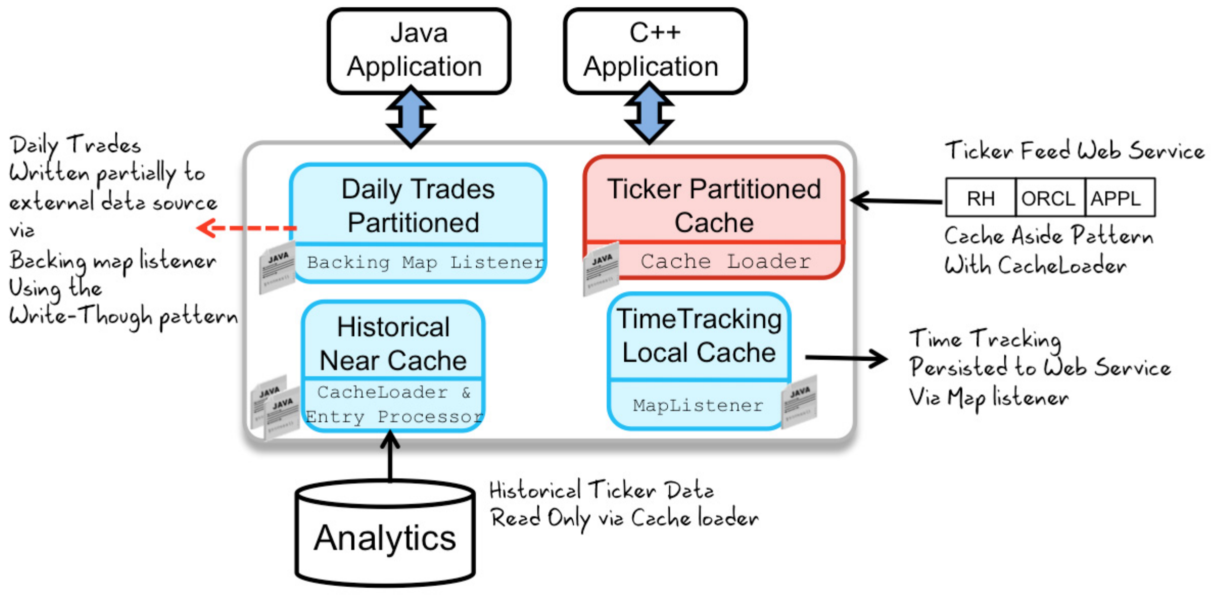

Java Application
C++ Application
Daily Trades Written partially to external data source via Backing map listener Using the Write-Though pattern
Daily Trades Partitioned
Backing Map Listener
Ticker Partitioned Cache
Cache Loader
Ticker Feed Web Service
RH
ORCL
APPL
Cache Aside Pattern With CacheLoader
Historical Near Cache
CacheLoader & Entry Processor
TimeTracking Local Cache
MapListener
Time Tracking Persisted to Web Service Via Map listener
Analytics
Historical Ticker Data Read Only via Cache loader

Figure 1 Financial Application Design

Component Description	
Component	**Description**
Ticker Partitioned Cache	The ticker cache is a partitioned read-only cache. The cache uses a `CacheLoader` implementing the cache-aside pattern to access a backing Web service that is known to be slow, but tends to be reliable. Individual traders tend to look at a set of ticks, but often look outside their set at any possible stock.
Time Tracking Local Cache	Each of the servers in the Coherence cluster is located in a local office, connected by a high speed connection to other offices. Timing records are maintained locally, but persisted locally on add via a Map Listener, again to a Web service running in the local office.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Historical Local Cache	Historical trading data is maintained in a backing database shared by all offices. Each office, depending on local trading, performs analytics on the data, often running many computations on the same data. The proposed scenario is a near-cache scheme, whereby the data that is maintained is probably fairly large, and needs to be shared and available across all cluster nodes, and requires fast reading by a front-tier local cache. An `EntryProcessor` is involved as setting up a query-based filter that selects the keys that are required for certain computations, and the computation code is sent via `EntryProcessor`s to each of the data entries that match the query. The `EntryProcessor` returns results based on calculations, but does not have to perform any updates on the data because it is historical in nature.
Daily Trades Partitioned Cache	Daily trades are tracked in a transactional Web application, but written via a backing map listener on each cluster member to a slower external data source. The backing map listener writes each trade synchronously to this source, per trading requirements. If the trade cannot be written, the backing map throws an exception, and the broker aborts the trade. The cache is transactional, and managed via the write-through pattern.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Detailed Instructions

Review the Online Music Application Design	
10-2.1	Review the design individually or in small groups.
10-2.2	Explain the reasoning, which may or may not be sound, behind each component, and list advantages and disadvantages of each selected Coherence topology.

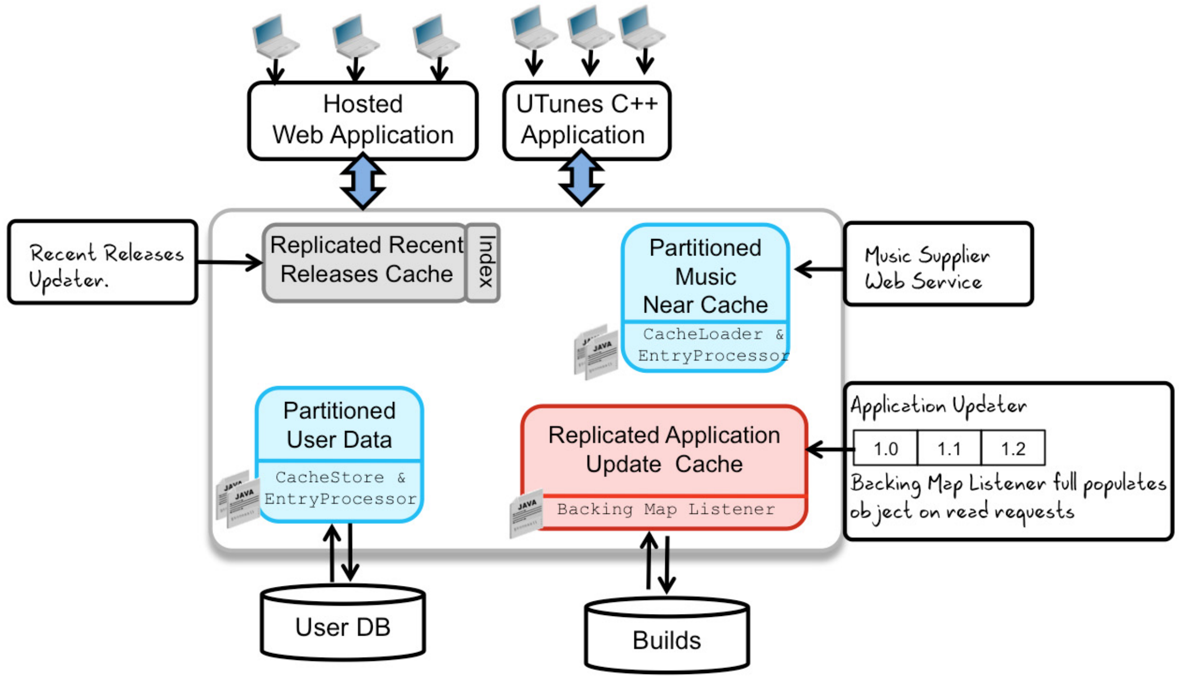

Hosted Web Application
UTunes C++ Application
Recent Releases Updater.
Replicated Recent Releases Cache
Index
Partitioned Music Near Cache
CacheLoader & EntryProcessor
Music Supplier Web Service
Partitioned User Data
CacheStore & EntryProcessor
Replicated Application Update Cache
Backing Map Listener
Application Updater
1.0
1.1
1.2
Backing Map Listener full populates object on read requests
User DB
Builds

Figure 2 Online Music Application Design

Component Description	
Component	**Description**
Hosted Web Application	A hosted Web application exists which provides a Web services interface to remote devices that can access a music library and buy music. Many instances of this application exist to load balance across the entire country.
UTunes Application	Users can load music onto their computers using the C++ UTunes application, which provides access to search, buy, and access music.
Recent Releases Cache	The recent releases replicated cache provides instance access to recently-released music to all members of the cluster. A replicated cache was selected because all recently-released music is often downloaded by many users. The list of recent releases is relatively small at any given time, and the list is not updated frequently. The cache is also indexed on several fields, including artist and title.
Partitioned Music Cache	Users access music locally, but often from many different genres and artists. These users access the same music, often repeatedly, but only for a brief period, and not across the cluster. The Partitioned Near Music Cache stores this music for a brief period before

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Component	Description
	discarding it. The cache loads music lazily via a `CacheLoader`. Titles and artwork are loaded first, and song content is only loaded on demand. An `EntryProcessor` is part of an administrative tool that applies discounts to music based on specials the business is having, for marking certain entries in other ways as well, such as clearance, and for other reasons.
Replicated Application Update Cache	Applications, both mobile and UTunes itself, are upgraded over time. The replicated Application Update Cache uses a backing map listener to complete the population of software updates when actual downloads exist. Data is loaded lazily with only platform, version, availability, and other data initially available.
Partitioned User Data Cache	User data is local and stored partitioned. A `CacheStore` is used to load a user and update details, credits, and so on. It is being considered if a write-behind or write-through strategy is appropriate here. This cache also has the potential to use an `EntryProcessor`. Situations exist where customers could have their status updated (none, sliver, gold, platinum,) as well as to target certain promotions to users that have spent a certain amount.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Instructions

Review the Airline Booking Design	
10-3.1	Review the design individually or in small groups.
10-3.2	Explain the reasoning, which may or may not be sound, behind each component, and list advantages and disadvantages of each selected Coherence topology.

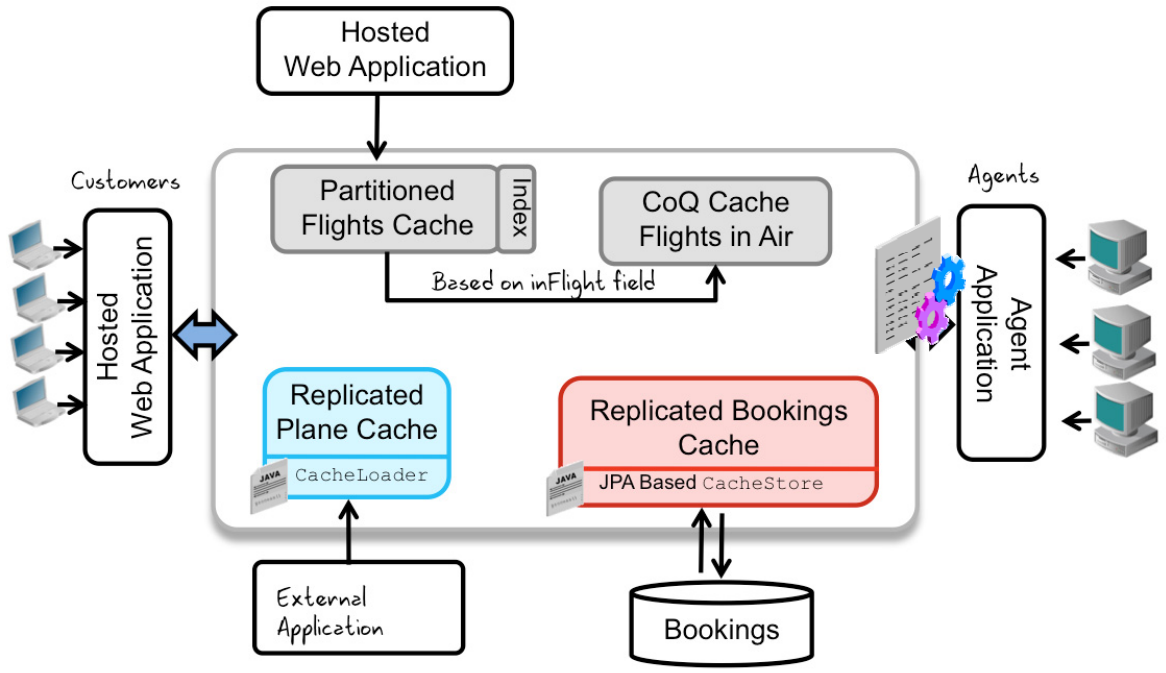


Figure 3 Airline Booking System Application Design

Component Description	
Component	**Description**
Hosted Web Application	Customer access is provided via a hosted Web application.
Agent Application	Agents access the system via a custom SWING-based application.
Partitioned Flights Cache	Flights are partitioned and indexed based on the `inFlight` status which comes from an external air traffic Web service.

CoQ Cache, Flights in Air	The continuous query cache, flights in air, is used by agents and based on the Partitioned Flights cache. A query, using the `inFlight` field with a status of `flying`, is used to keep this cache constantly up to date.
Replicated Plane Info	The Plane Information case is used by booking agents to determine flight capacity and is updated by an external C++ application. There is a discussion currently as to whether this should be a near cache or not.
Replicated Bookings Cache	The replicated bookings cache is used by both customers and agents to track bookings. The cache is a JPA-based cache store enabled as write-through. Historically, this data was accessed via a Web application based on JPA that was leveraged for the new infrastructure. Another discussion is going on which is pushing for a near cache rather than a replicated cache.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 11

Chapter 12

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 11

Practices Overview

Developers must be able to configure the Coherence Proxy Service scheme for use with Coherence*Extend TCP/IP clients.

The objective of this practice is to assist the developer with configuring Coherence*Extend and writing Coherence*Extend Java and C++ clients.

In this practice, you will perform the following tasks:

- Configure, run, and review a Coherence*Extend-enabled cache server.
- Configure, run, and review a Coherence*Extend Java client.
- Configure, run, and review a Coherence*Extend C++ client.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 11-1: Configure, Run, and Review a Coherence*Extend-enabled Cache Server

Skills Learned

At the end of this practice, you should be able to:

- Specify command line arguments to the Coherence server start script to override the default `coherence-cache-config.xml` file.
- Configure a Coherence*Extend Proxy Service scheme.
- Start a Coherence proxy server.
- Identify that Coherence is using the specific configuration.
- Run subsequent labs for this lesson using this environment.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Problem Statement

From:	Billi.Onair@retail-is-us.com
Sent:	Tuesday,July 21st, 5:05amm
To:	you@acmedev.com
Subject:	Configuring a Coherence*Extend Proxy Service

Hello!

Our application continues to grow exponentially! We know that Coherence supports running Java applications as full cluster members, either with or without local storage enabled. This works fine in an environment where the nodes are all running on the same high-speed network and written in Java. However, we now have a need to expand internationally. This involves running clients over WAN connections, and languages other than Java.

What do you do when a client application runs on somebody's desktop over a WAN? If the application is written in Java, it can run with local storage disabled, but then it is still running as a member of the cluster, which probably is not a good idea, because there can potentially be thousands of clients. Disabling local storage helps, but that still has some overhead associated with it, and is not the best choice for client applications. It works better for long-running applications that work closely with the cluster. What if the application is not written in Java, but is written in C++ or a language that supports .Net?

We have heard that we should consider Coherence*Extend. We understand that Coherence*Extend is the only option for applications not written in Java to connect to a cluster, and is a great way to connect transient Java clients without making them actual members of the cluster. This is also desirable because it avoids the high latency associated with WAN requests, and the requirement for cluster nodes to be "good citizens" that provide reasonable GC, CPU, and memory performance.

Our engineers say that you have the skills to configure and run a Coherence Proxy Server. We would like you to set up an example cluster that includes a Proxy Server that we can use to test with different Coherence*Extend clients. We look forward to seeing your example!

Regards,

Billi Onair
Retail Is Us
Billi.Onair@retail-is-us.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Design

In this practice, you will add a Proxy Service configuration to a `coherence-cache-config.xml` file. In addition to configuring cache schemes, this file is used by Coherence to configure Proxy Services. You will use the `<proxy-scheme>` element to configure the Proxy Service. The configuration includes a `<tcp-acceptor>` element that specifies which networking host and port to use for TCP/IP connections. Next, you configure the system property that instructs Coherence to use your configuration file that contains the Proxy Service configuration. Finally, you run the cluster with the Proxy Service running as part of the cluster. This practice sets up the cluster and Proxy Service that is used by the rest of the practices in this lesson. The diagram below shows the architecture of the practices. The red dotted line circles the part of the architecture that this practice involves.

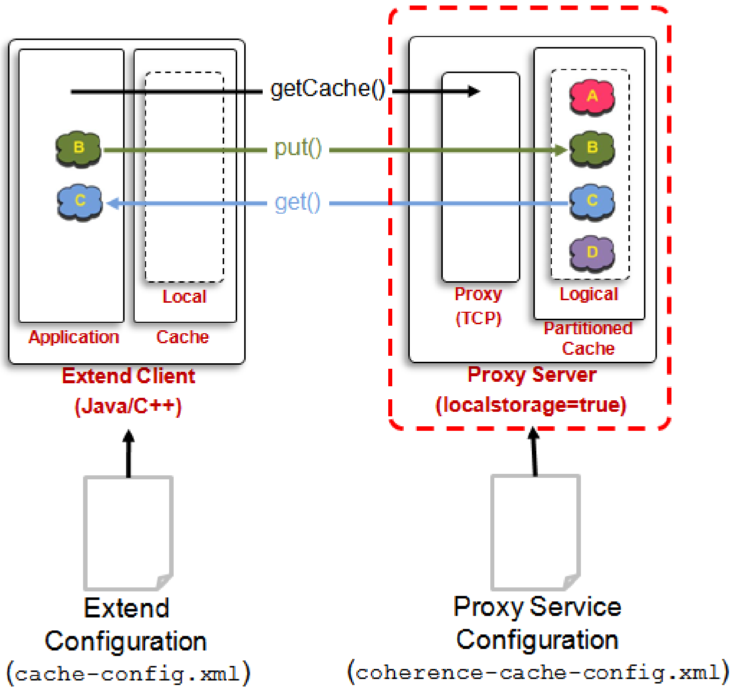


Figure 1 Extend Proxy Architecture: Proxy Service

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources\`**	Contains a set of practice specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Coherence Cache Override (`-Dtangosol.coherence.cacheconfig`**)** Specifies which configuration file to use for Coherence caching	
Not Set	By default, Coherence will use the `coherence-cache-config.xml` file that is bundled in `%COHERENCE_HOME%\lib\coherence.jar`.
`filename`	Specifies the filename of the configuration file to use.

Table 11.1

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Configure a proxy server and run in a Coherence cluster.
2	Analyze cluster execution results.

Hints:

- The `<proxy-scheme>` element configures a Proxy Service.
- A Proxy Service is configured in the `coherence-cache-config.xml` file.
- The `<acceptor-config>` element contains the `<tcp-acceptor>` element, which specifies the host and port to use for TCP/IP Extend client connections.
- **Remember!** Many of the practices in this course (including this practice) have predefined Eclipse `TODO`: markers in their project resources to help you quickly locate the changes you need to make. To see a list of these predefined tasks for the currently open projects, select **Window > Show View > Tasks.** Note that ALL open projects will display their tasks in the Tasks tab. You may want to close all projects, except the current one, to minimize the number of tasks displayed.

Detailed Instructions

Configure a proxy server and run in a Coherence cluster	
11-1.1	Within Eclipse, expand project **Practice.11.01** and expand the **config** and **src** folders to display their contents.
11-1.2	Open the `coherence-cache-config.xml` file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

11-1.3 Locate the text: "TODO: Place proxy scheme here," within the `<caching-schemes>` element, and replace it with the configuration for a Coherence proxy scheme.
Note: For convenience, the configuration snippets for this exercise can be found in the `resources` subdirectory for **Practice.11.01** in the **`code.snippet.txt`** file. Your configuration should resemble:

```
<proxy-scheme>
  <scheme-name>MyExtendProxyScheme</scheme-name>
  <service-name>ExtendTcpProxyService</service-name>
  <thread-count>50</thread-count>

  <acceptor-config>
    <tcp-acceptor>
      <local-address>
        <address>localhost</address>
        <port>9099</port>
      </local-address>
    </tcp-acceptor>
  </acceptor-config>

  <autostart>true</autostart>
</proxy-scheme>
```

11-1.4 From within Eclipse:

1. Make sure the **Practice.11.01** project is selected.
2. Click the **arrow** next to the icon for run configurations in the upper left side of the IDE.
3. Select **Run Configurations…**
4. On the **Run Configurations** screen, select **Java Application** > **Practice.11.01.ProxyServer** in the left window.
5. Click the **Arguments** tab on the right side of the dialog, and enter the system property that points to your **`coherence-cache-config.xml`** file in the **VM arguments** input box.
6. Click the **Apply** button.

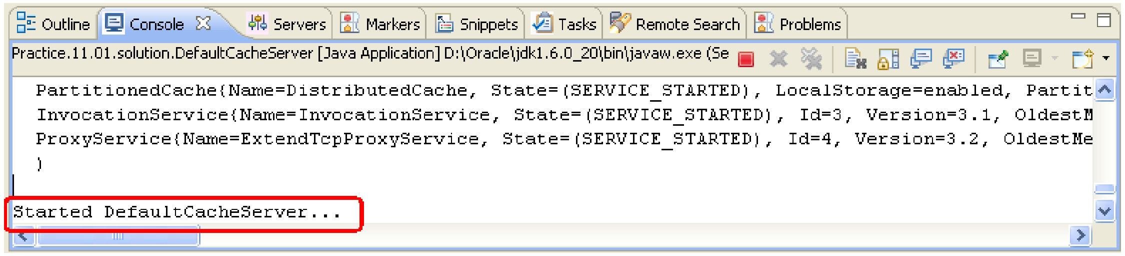
Your VM Arguments input box should resemble:

```
-Xms128m -Xmx128m -Dtangosol.coherence.ttl=0 -
Dtangosol.coherence.cacheconfig=D:\Oracle\student\practi
ces\Practice.11.01\config\coherence-cache-config.xml
```

Now that the configuration is done, you are ready to start your cluster.

11-1.5 To run the **ProxyServer** instance, select **Run** on the lower right side of the **Run Configurations** dialog window.

This starts a DefaultCacheServer that uses your Proxy Service configuration and displays its output within the Eclipse IDE in the Console tab:



```
Practice.11.01.solution.DefaultCacheServer [Java Application] D:\Oracle\jdk1.6.0_20\bin\javaw.exe (Se
  PartitionedCache(Name=DistributedCache, State=(SERVICE_STARTED), LocalStorage=enabled, Partit
  InvocationService(Name=InvocationService, State=(SERVICE_STARTED), Id=3, Version=3.1, OldestM
  ProxyService(Name=ExtendTcpProxyService, State=(SERVICE_STARTED), Id=4, Version=3.2, OldestMe
  )

Started DefaultCacheServer...
```

You should see output similar to:

```
Oracle Coherence Version 3.6.0.0 Build 17229
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All
rights reserved.


2010-06-28 15:11:24.703/0.547 Oracle Coherence GE 3.6.0.0 DPR2
<Info> (thread=main, member=n/a): Loaded cache configuration
from
"file:/D:/Oracle/student/practices/Practice.11.01/config/cohere
nce-cache-config.xml"
2010-06-28 15:11:25.171/1.015 Oracle Coherence GE 3.6.0.0 DPR2
<D5> (thread=Cluster, member=n/a): Service Cluster joined the
cluster with senior service member n/a
. . .
2010-06-28 15:11:26.640/2.484 Oracle Coherence GE 3.6.0.0 DPR2
<Info> (thread=Proxy:ExtendTcpProxyService:TcpAcceptor,
member=4): TcpAcceptor now listening for connections on
139.185.35.126:9099
2010-06-28 15:11:26.656/2.500 Oracle Coherence GE 3.6.0.0 DPR2
<D5> (thread=Proxy:ExtendTcpProxyService:TcpAcceptor,
member=4): Started:
TcpAcceptor{Name=Proxy:ExtendTcpProxyService:TcpAcceptor,
State=(SERVICE_STARTED), ThreadCount=50, HungThreshold=0,
TaskTimeout=0, Codec=Codec(Format=POF), PingInterval=0,
PingTimeout=0, RequestTimeout=0,
LocalAddress=[edrsr26p1/139.185.35.126:9099],
LocalAddressReusable=true, KeepAliveEnabled=true,
TcpDelayEnabled=false, ReceiveBufferSize=0, SendBufferSize=0,
ListenBacklog=0, LingerTimeout=-1,
BufferPoolIn=BufferPool(BufferSize=2KB, BufferType=DIRECT,
Capacity=Unlimited), BufferPoolOut=BufferPool(BufferSize=2KB,
BufferType=DIRECT, Capacity=Unlimited)}
2010-06-28 15:11:26.656/2.500 Oracle Coherence GE 3.6.0.0 DPR2
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

```
<D5> (thread=Proxy:ExtendTcpProxyService, member=4): Service
ExtendTcpProxyService joined the cluster with senior service
member 4
2010-06-28 15:11:26.656/2.500 Oracle Coherence GE 3.6.0.0 DPR2
<Info> (thread=main, member=4):
Services
  (
  ClusterService{Name=Cluster, State=(SERVICE_STARTED,
STATE_JOINED), Id=0, Version=3.6, OldestMemberId=2}
  InvocationService{Name=Management, State=(SERVICE_STARTED),
Id=1, Version=3.1, OldestMemberId=2}
  PartitionedCache{Name=DistributedCache,
State=(SERVICE_STARTED), LocalStorage=enabled}
  InvocationService{Name=InvocationService,
State=(SERVICE_STARTED), Id=3, Version=3.1, OldestMemberId=2}
  ProxyService{Name=ExtendTcpProxyService,
State=(SERVICE_STARTED), Id=4, Version=3.2, OldestMemberId=4}
  )

Started DefaultCacheServer...
```

You have successfully configured a proxy server and ran it in a Coherence cluster. Are you sure? How do you know?
The next section explores these questions with you.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze cluster execution results	
11-1.6	Take a closer look at the output in the command window. The first thing you should find is the line showing that your cache configuration file was used.
11-1.7	The next messages to look for have “ExtendTcpProxyService” and “Proxy” in them. This shows that your proxy configuration is running. You should also see a message indicating that the node is listening on your TCP host and port: `2010-06-28 15:11:26.640/2.484 Oracle Coherence GE 3.6.0.0 DPR2 <Info> (thread=Proxy:ExtendTcpProxyService:TcpAcceptor, member=4): ` **`TcpAcceptor now listening for connections on 139.185.35.126:9099`** Now that you have run a Coherence proxy server, and verified that it started and is listening for TCP connections, you will write some clients that use it. **Note:** Leave the proxy server running for the next lab.

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace and open **Practice.11.01.solution**. Follow the directions for testing the application from steps **11-1.6**, ignoring any steps to add configuration, all of which is in the solution.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 11-2: Configure, Run, and Review a Coherence*Extend Java client

Skills Learned

At the end of this practice, you should be able to:

- Write a Coherence*Extend Java client.
- Verify that the application ran successfully.

Problem Statement

From:	Tom.Foolery@retail-is-us.com
Sent:	Wednesday, July 22, 9:02am
To:	you@acmedev.com
Subject:	Writing a Java Extend Client

Hello!

Thank you for setting up a Coherence Proxy Server for us. Now that a Proxy Server is configured and running, clients can connect to it via TCP/IP and use the cluster. We would like to test connectivity for our Java clients against the new Proxy Server.

As we understand it, a client-side configuration file is needed to tell the Coherence*Extend Java client how to connect to a running proxy server, as well as the cache configuration that the Extend client will use. A Coherence*Extend client can map a cache directly to a remote caching scheme serviced by a proxy server. The Proxy Service will delegate cache calls to the corresponding cache within the cluster. Additionally, the client can configure a near cache that specifies the remote caching scheme as the back of the near cache. This way a Coherence*Extend client can keep frequently-used data close; thereby reducing read time, cluster-side CPU cycles, and network bandwidth.

Our engineers have set up a base project that you can use to write a Java Extend client that connects to our Proxy Server. We look forward to seeing your example!

Regards,

Tom Foolery
Retail Is Us
Tom.Foolery@retail-is-us.com

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Design

In this practice, you will add a remote cache scheme to an Extend `coherence-cache-config.xml` file. You use the `<remote-cache-scheme>` element to define the connection properties that connect the Extend client to the Proxy Service configured in the previous practice. The configuration includes an `<initiator-config>` element that contains a `<tcp-initiator>` element that specifies which networking host and port to use for TCP/IP connections. The remote cache scheme name is set as `MyExtendTCPScheme`, which is used by cache mappings to specify that cache requests are handled by the remote cache scheme, which works over TCP/IP.

In addition to configuring the remote cache scheme, this file is used by Coherence*Extend to configure a near cache. You will use the `<near-scheme>` element to configure the near cache. The front scheme of the near cache is a local cache, and the back scheme is a reference to `MyExtendTCPScheme`, which means the back scheme is handled by the remote cache scheme. Next, the `<cache-mapping>` element is used to map requested caches to the appropriate schemes. In this practice, you will define two cache mappings:

1. A near cache mapping, whereby all requests for caches that start with "near-" as part of the cache name are mapped to the near cache scheme, `MyNearScheme`.
2. A direct mapping to the remote cache scheme for all other cache names. This results in the cache type being whatever maps to the cache name as it is configured on the cache configuration where the Proxy Service is configured.

Then, you write the code that gets `NamedCache` reads and writes data to the cache. Finally, when the code is completed, you will run the Java Extend client two times to see how it behaves when the near cache is used, and when the partitioned cache configured on the Proxy side is used directly.

This practice uses the cluster and Proxy Service that should still be running after the previous practice. The diagram below shows the architecture of the practices. The red dotted line circles the part of the architecture that this practice involves.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

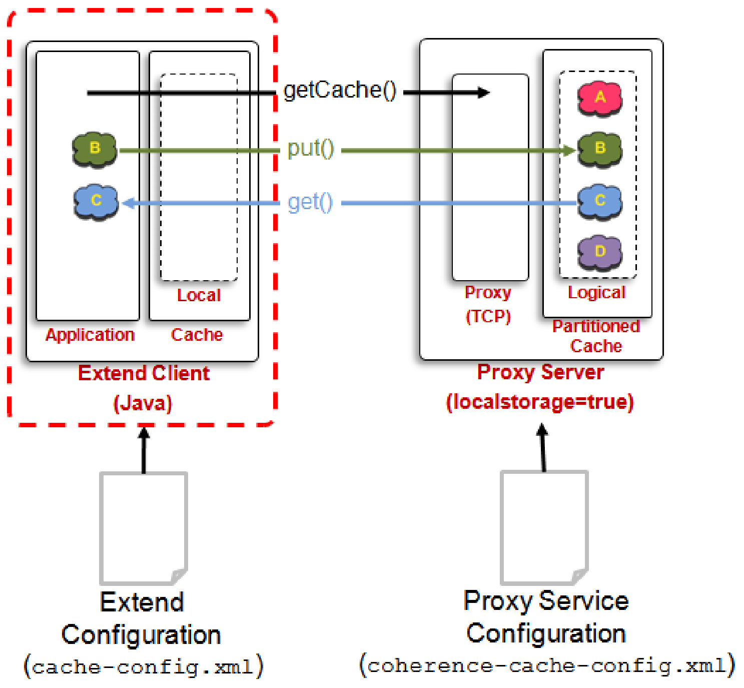


Figure 2 Extend Proxy Architecture: Java Client

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\` **`resources\`**	Contains a set of practice specific resource directories
Support	`D:\oracle\student\` **`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\` **`software`**	Contains installable Coherence and other files

Coherence Cache Override (`-Dtangosol.coherence.cacheconfig`**)** Specifies which configuration file to use for Coherence caching	
Not Set	By default, Coherence will use the `coherence-cache-config.xml` file that is bundled in `%COHERENCE_HOME%\lib\coherence.jar`.
`filename`	Specifies the filename of the configuration file to use

Table 11.2

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Hints:

- The `<near-scheme>` element configures a near cache that runs within the memory space of the Java Extend client. The front-tier local cache runs in memory on the client. The back-tier cache is mapped to the `MyExtendTCPScheme`, which is configured as a remote cache scheme that will connect to the Proxy Service from Practice.11.01.
- The `<remote-cache-scheme>` configures the client side of the Extend Proxy Service connection. It contains the `<initiator-config>` element which uses the `<tcp-initiator>` element to define remote addresses where the host and port of the server side Proxy Service are configured.
- There are two cache mappings to define: One that uses the near-scheme when a cache name starts with "near-", and one that uses the partitioned scheme defined on the server side cluster for all other cache names.
- The code for writing a Coherence*Extend Java client is the same as other Java caching programs you have written during the rest of this course.
- The Java Extend client is run twice: once with the near cache, and once using the remote cache directly.

Overview Instructions

1	Configure, code, and run a Coherence*Extend Java client.
2	Analyze execution results.

Detailed Instructions

Configure, code, and run a Coherence*Extend Java client	
11-2.1	Within Eclipse, expand project **Practice.11.02** and expand the **config** and **src** folders to display their contents.
11-2.2	Open the `coherence-cache-config.xml` file.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

11-2.3 Locate the text: “TODO: Place near cache scheme here,” within the `<caching-schemes>` element, and replace it with the configuration for a Coherence near-caching scheme.
Note: For convenience, the configuration snippets for this exercise can be found in the resources subdirectory for **Practice.11.02** in the **`code.snippet.txt`** file. Your configuration should resemble:

```
<near-scheme>
  <scheme-name>MyNearScheme</scheme-name>

  <front-scheme>
    <local-scheme/>
  </front-scheme>

  <back-scheme>
    <remote-cache-scheme>
      <scheme-ref>MyExtendTCPScheme</scheme-ref>
    </remote-cache-scheme>
  </back-scheme>

</near-scheme>
```

11-2.4 Locate the text: “TODO: Place remote cache scheme here,” within the `<caching-schemes>` element, and replace it with the configuration for a Coherence remote caching scheme that connects to the proxy server from the previous lab using TCP/IP. Your configuration should resemble:

```
<remote-cache-scheme>
  <scheme-name>MyExtendTCPScheme</scheme-name>
  <service-name>ExtendTcpCacheService</service-name>
  <initiator-config>
    <tcp-initiator>
      <remote-addresses>
        <socket-address>
          <address>localhost</address>
          <port>9099</port>
        </socket-address>
      </remote-addresses>
    </tcp-initiator>
  </initiator-config>
</remote-cache-scheme>
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

11-2.5	Locate the text: “TODO: Place remote cache scheme here,” within the `<caching-scheme-mapping>` element, and replace it with the configuration to map a cache of any name that starts with near- to the near cache scheme defined previously. Your configuration should resemble:
	`<cache-mapping>`
	  `<cache-name>near-*</cache-name>`
	  `<scheme-name>MyNearScheme</scheme-name>`
	`</cache-mapping>`

11-2.6	Locate the text: “TODO: Place direct TCP cache-mapping here,” within the `<caching-scheme-mapping>` element, and replace it with the configuration to map a cache of any name to the remote cache scheme defined previously, other than caches with names starting with "near-" (as they are mapped to the near scheme). Your configuration should resemble:
	`<cache-mapping>`
	  `<cache-name>*</cache-name>`
	  `<scheme-name>MyExtendTCPScheme</scheme-name>`
	`</cache-mapping>`
	
	Save your changes.

11-2.7	Within the **src** folder, open the **`JavaExtendClient.java`** file for editing. Locate the text: “TODO: Put call to getCache() here,” and replace with the code that uses a `CacheFactory` to obtain a reference to a `NamedCache` using the variable `cacheName`. Your code should resemble: `NamedCache myCache = CacheFactory.getCache(cacheName);`
11-2.8	Within the same **`JavaExtendClient.j`**`ava` file, locate the text: “TODO: Put Customer object into cache here,” and replace with the code that uses the reference to the `NamedCache` returned in the last step to put the `Customer` object into the cache. **Note:** There are two places to replace in this file with this code. Your code should resemble: `myCache.put(cust.getId(), cust);`

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

11-2.9 Within the same **`JavaExtendClient.java`** file, locate the text: “TODO: Get Customer object from cache here,” and replace with the code that uses the reference to the `NamedCache` returned in the previous step to get the `Customer` object from the cache. Your code should resemble:

```
        cust = (Customer)myCache.get(cust.getId());
```

11-2.10 Within the same **`JavaExtendClient.java`** file, locate the text: “TODO: Put call to disconnect the Extend client here,” and replace with the code that disconnects from a running Coherence proxy server. Your code should resemble:

```
        CacheFactory.shutdown();
```

Save your changes. Now that the coding is done, keep the file open and take a closer look at what the code does.

11-2.11 Examine the **`JavaExtendClient.java`** code and follow along with the following:

1. The program takes one optional argument, `-cache` followed by the name of a cache. The default cache name is `myCache`. Because the client-side configuration has two cache mapping defined, one for `near-*` that maps to a near cache, and one for `*` that maps directly to the remote cache, the `-cache` option causes the program to either use a near cache or directly use a remote cache.
2. The program creates a `NamedCache` called whatever your specified cache name is, and creates a single `Customer` object that is placed into the cache. The program then times how long it takes to read the `Customer` object from the cache five times, then times how long it takes to write the `Customer` object to the cache five times.
3. The program is run twice, using a near cache and remote cache type, and the performance numbers are compared. This reveals how a near cache on Coherence*Extend performs compared to direct cache usage over TCP/IP.

The next few steps will instruct you how to run this scenario.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

11-2.12 From within Eclipse:

1. Open the Run Configuration for **Practice.11.02**.
2. Click the **Arguments** tab on the right side of the dialog, and enter the system property that points to your `coherence-cache-config.xml` file in the **VM arguments** input box.
3. Note that **Program Arguments** has `-cache myCache` as the arguments to send to the running program.
4. Click the **Apply** button.

Your VM Arguments input box should resemble:

```
-Xms128m -Xmx128m -Dtangosol.coherence.ttl=0 -
Dtangosol.coherence.cacheconfig=D:\Oracle\student\practices\Pra
ctice.11.02\config\coherence-cache-config.xml
```

Now that the configuration is done, you are ready to start your cluster.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

11-2.13 Select **Run** on the lower right side of the **Run Configurations** dialog window to run the program using the remote cache directly. Take note of the rate of operations per second. You should see output similar to:

```
Oracle Coherence Version 3.6.0.0 Build 17229
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

2010-06-30 14:49:49.328/0.532 Oracle Coherence GE 3.6.0.0 DPR2 <Info>
(thread=main, member=n/a): Loaded cache configuration from
"file:/D:/Oracle/student/practices/Practice.11.02/config/coherence-
cache-config.xml"
2010-06-30 14:49:49.515/0.719 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=ExtendTcpCacheService:TcpInitiator, member=n/a): Started:
TcpInitiator{Name=ExtendTcpCacheService:TcpInitiator,
State=(SERVICE_STARTED), ThreadCount=0, Codec=Codec(Format=POF),
PingInterval=0, PingTimeout=0, RequestTimeout=0, ConnectTimeout=0,
RemoteAddresses=[edrsr26p1/139.185.35.126:9099],
KeepAliveEnabled=true, TcpDelayEnabled=false, ReceiveBufferSize=0,
SendBufferSize=0, LingerTimeout=-1}
2010-06-30 14:49:49.531/0.735 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=main, member=n/a): Opening Socket connection to
139.185.35.126:9099
2010-06-30 14:49:49.531/0.735 Oracle Coherence GE 3.6.0.0 DPR2 <Info>
(thread=main, member=n/a): Connected to 139.185.35.126:9099
==================================================
Time for myCache get():StopWatch{startTime=4406704865035128,
finishTime=4406704875944393, elapsedTime=10909265 ns (0.010909 sec),
rate=458.326019 per sec}
==================================================
Time for myCache put():StopWatch{startTime=4406704880807485,
finishTime=4406704892325491, elapsedTime=11518006 ns (0.011518 sec),
rate=434.102917 per sec}
=====================================================
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

11-2.14 Open the **Run Configuration** again and:

1. Change the **Program Arguments** to: `-cache near-myCache`.
2. Click the **Apply** button to save your changes.
3. Select **Run** on the lower right side of the **Run Configurations** dialog window to run the program using the near cache which is backed by the remote cache. Take note of the rate of operations per second.

You should see output similar to:

```
Oracle Coherence Version 3.6.0.0 Build 17229
 Grid Edition: Development mode
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

2010-06-30 14:56:11.718/0.547 Oracle Coherence GE 3.6.0.0 DPR2 <Info>
(thread=main, member=n/a): Loaded cache configuration from
"file:/D:/Oracle/student/practices/Practice.11.02/config/coherence-
cache-config.xml"
2010-06-30 14:56:11.921/0.750 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=ExtendTcpCacheService:TcpInitiator, member=n/a): Started:
TcpInitiator{Name=ExtendTcpCacheService:TcpInitiator,
State=(SERVICE_STARTED), ThreadCount=0, Codec=Codec(Format=POF),
PingInterval=0, PingTimeout=0, RequestTimeout=0, ConnectTimeout=0,
RemoteAddresses=[edrsr26p1/139.185.35.126:9099],
KeepAliveEnabled=true, TcpDelayEnabled=false, ReceiveBufferSize=0,
SendBufferSize=0, LingerTimeout=-1}
2010-06-30 14:56:11.921/0.750 Oracle Coherence GE 3.6.0.0 DPR2 <D5>
(thread=main, member=n/a): Opening Socket connection to
139.185.35.126:9099
2010-06-30 14:56:11.937/0.766 Oracle Coherence GE 3.6.0.0 DPR2 <Info>
(thread=main, member=n/a): Connected to 139.185.35.126:9099
==================================================
Time for near-myCache get():StopWatch{startTime=4407087230734333,
finishTime=4407087236990962, elapsedTime=6256629 ns (0.006257 sec),
rate=799.152387 per sec}
==================================================
Time for near-myCache put():StopWatch{startTime=4407087241571436,
finishTime=4407087255930913, elapsedTime=14359477 ns (0.014359 sec),
rate=348.202097 per sec}
=====================================================
```

You have successfully created a client-side Coherence*Extend configuration and ran a Java Extend client that connected to a Coherence cluster using a proxy server. Are you sure? How do you know?

The next section explores these questions with you.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Analyze local cache execution results	
11-2.15	Take a closer look at the output in the command window. The first thing you should find in the window is the line showing that your cache configuration file was loaded (it shows the `remote-cache-scheme` configured.)
11-2.16	Next, you want to look for messages that have the word "ExtendTcpCacheService" in them. This shows that the remote TCP cache service was started, and that it connected to the remote proxy server. This is a positive indication that a Coherence*Extend configuration is in use.
11-2.17	The next thing you notice while looking at both runs of the program is that it produced different performance numbers. The first run that used the remote cache directly had a slower read number, and a slightly higher write number per second. This is because the near cache that is running locally on the Coherence*Extend Java client side is accessed for the last four reads instead of going all the way to the proxy server to get the data again. The write number decreased slightly because writes always have to go back to the cluster, and the near cache also has to get invalidated. Now that you have configured and run a Coherence*Extend Java client, and verified its behavior, take a look at running a C++ client against your proxy server. **Note:** Close the JavaExtendClient command window before proceeding. You will also need to make some configuration changes to the proxy server, so shut all servers down.

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace and open **Practice.11.02.solution**. Follow the directions for testing the application from steps **11-2.11**, ignoring any steps to add code, all of which is in the solution.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practice 11-3: Configure, Run, and Review a Coherence*Extend C++ client

Skills Learned

At the end of this practice, you should be able to:

- Write a Coherence*Extend C++ client that uses POF.
- Verify that the application ran successfully.

Problem Statement

From:	Ted.Ious@retail-is-us.com
Sent:	Wednesday, July 22nd, 11:12am
To:	you@acmedev.com
Subject:	Writing a C++ Extend Client

Hello!

We have developed a C++ application that we would like to integrate with Coherence. We would like your help in testing connectivity for our C++ client against the new Proxy Server.

As we understand it, C++ applications require using POF serialization and the Coherence*Extend C++ binaries to use a Coherence cache.

Our engineers have set up a base project that you can use to create your test environment. You can also modify the Practice.11.01 environment to enable it to use POF, install the Coherence*Extend C++ product, examine the C++ source code to identify the Coherence C++ API and POF coding techniques in use, run the Coherence*Extend C++ client to test that it works, and review the results to ensure that our settings took effect. We look forward to seeing your example!

Regards,

Ted Ious
Retail Is Us
Ted.Ious@retail-is-us.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Design

In this practice you will use the same configuration that was used in the previous practice for the C++ Extend client. The C++ client is already developed, so there is no coding required for this practice. The client is already built as well, so there is no need to install a compiler or go through a build process.

1. First, you will modify the Proxy Server configured in Practice.11.01 to enable POF serialization, which requires restarting the Proxy Server.
2. Next, you need to install the Coherence*Extend C++ product so the C++ Coherence libraries are available on the machine.
3. Then you will examine the C++ client code to understand how it works and to see how it uses POF serialization.
4. Finally, you will run the C++ client and examine the results.

This practice uses the Proxy Service that should still be running after the previous practice. The diagram below shows the architecture of the practices. The red dotted line circles the part of the architecture that this practice involves.

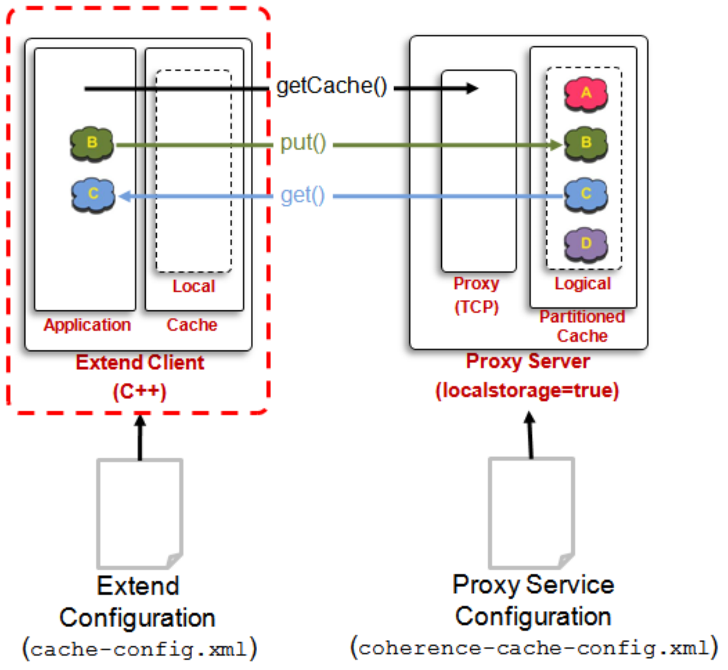


Figure 3 Extend Proxy Architecture: C++ Client

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Resources and References

POF Serialization Approaches

Approach	Coherence headers in data object	Requires deriving from Object	Supports `const` data members	External serialization routine	Requires zero-arg constructor
`Managed<T>`	**No**	**No**	**Yes**	**Yes**	**Yes**
`PortableObject`	Yes	Yes	No	No	Yes
`PofSerializer`	Yes	Yes	Yes	Yes	No

Resource Directories		
Practice Resources	`D:\oracle\student\`**`resources`**`\`	Contains a set of practice specific resource directories
Support	`D:\oracle\student\`**`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\`**`software`**	Contains installable Coherence and other files

Coherence Cache Override (`-Dtangosol.coherence.cacheconfig`) Specifies which configuration file to use for Coherence caching	
Not Set	By default, Coherence will use the `coherence-cache-config.xml` file that is bundled in `%COHERENCE_HOME%\lib\coherence.jar`.
`filename`	Specifies the filename of the configuration file to use

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Sample Application Files Describes the purpose of each file in the C++ POF Trade example.	
`Trade.h`	Defines the interface of the basic `Trade` class and a few operator functions. Something to keep in mind with this example is that the `Trade` object model is completely unaware of Coherence, containing no Coherence API calls.
`Trade.cpp`	The implementation of the `Trade` class
`TradeSerializer.cpp`	The POF serialization class that implements the actual serialization routines
`cppexample.cpp`	The main part of the program that uses the `Trade` class and stores `Trade` objects into the Coherence cache using POF
`runexample.cmd`	Sets the environment and runs the `Trade` program

Table 11.3

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Overview Instructions

1	Configure a proxy server to use POF.
2	Install Coherence*Extend C++ product.
3	Examine Trade C++ source code.
4	Run a Coherence*Extend C++ client.
5	Analyze execution results.

Hints:

- The Extend cache configuration is identical to the configuration used for the Java Extend client.
- The Proxy Server Run Configuration needs to set the `-Dtangosol.pof.enabled` system property that enables POF serialization for the cluster.
- This practice uses the Proxy Server configured in Practice.11.01.
- You need to install the Coherence C++ product to run the C++ client.
- The C++ `TradeSerializer.cpp` file contains the `serialize()` and `deserialize()` methods that perform POF serialization operations.

Detailed Instructions

Configure a proxy server to use POF	
11-3.1	Within Eclipse, expand project **Practice.11.03** and expand the **config** and **src** folders to display their contents.
11-3.2	Open the `extend-cache-config.xml` file. Note that it is identical to the configuration file used in Practice.11.02.
11-3.3	Open the Run Configuration for **Practice.11.01** again and add the system property to the **Program Arguments** that instructs Coherence to enable POF serialization. **Note:** Although enabling POF is not required for running the proxy server, it is required for using POF, which is needed for this lab. Your **Program Arguments** should resemble: `-Xms128m -Xmx128m -Dtangosol.coherence.ttl=0 -`**`Dtangosol.pof.enabled=true`**` -Dtangosol.coherence.cacheconfig=D:\Oracle\student\practices\Practice.11.01\config\coherence-cache-config.xml` Click the **Apply** button to save your changes, then restart the proxy server.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Install Coherence*Extend C++ product	
11-3.4	Extract the contents of `D:\stage\coherence-cpp-3.6.0.0b16628-windows-x86-vs2005.zip` file to `D:\Oracle\coherence`. This should create a `D:\Oracle\coherence\coherence-cpp` folder that contains the Coherence*Extend C++ product.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Examine Trade C++ source code

11-3.5 Within the **Practice.11.03** project within Eclipse, open the **`Trade.h`** file and examine its code. It defines a few macros that represent POF index postions for when the serialization process takes place. The **`TradeSerializer.cpp`** file uses the following macros defined in `Trade.h`:

Macro	Value
`TRADE_SYMBOL`	0
`TRADE_PRICE`	1
`TRADE_ID`	2
`TRADE_QUANTITY`	3

Next, **`Trade.h`** defines a few member variables for the **`Trade`** class:

Variable	Type	Description
`symbol`	`std::string`	The stock symbols for this trade
`price`	`double`	The price of the trade
`id`	`int`	The trade ID
`quantity`	`int`	The number of shares for this trade

Then, it defines a basic constructor to set the values, a copy constructor, the getter and setter methods for the variables, and a few operational functions. The two constructors are required for using the `Managed<T>` POF serialization approach.

11-3.6 Open the **`Trade.cpp`** file and examine its code. You will see that it is a very basic implementation of the methods defined in `Trade.h`. Although the `Trade.cpp` class does not contain any Coherence code, it does implement a few functions that are required for using the `Managed<T>` POF approach:

Description	Example
Zero-arg constructor	`Trade::Trade()`
Copy constructor	`Trade::Trade(const Trade& that)`
Equality comparison operator	`bool operator== (const Trade& tradeA, const Trade& tradeB)`
Standard output stream function	`std::ostream& operator<< (std::ostream& out, const Trade& trade)`
A hash function	`size_t hash_value(const Trade& trade)`

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

11-3.7 Open the **`TradeSerializer.cpp`** file and examine its code. This is where the POF serialization methods are implemented for the **`Trade`** object.

The `COH_REGISTER_MANAGED_CLASS(1001, Trade)` statement registers the **`Trade`** class as a `Managed<T>` class with Coherence with the numeric identifier, **1001**, which must be unique across the cluster per class.

The methods **`TradeSerializer.cpp`** implements are:

Returns	Method	Parameters	Description

`void`	`serialize`	`PofWriter::Handle hOut`
		`const Trade& trade`
			Serializes the Trade object's variables using the POF Writer API

`Trade`	`deserialize`	`PofReader::Handle hIn`	Deserializes the Trade object's variables using the POF Reader API

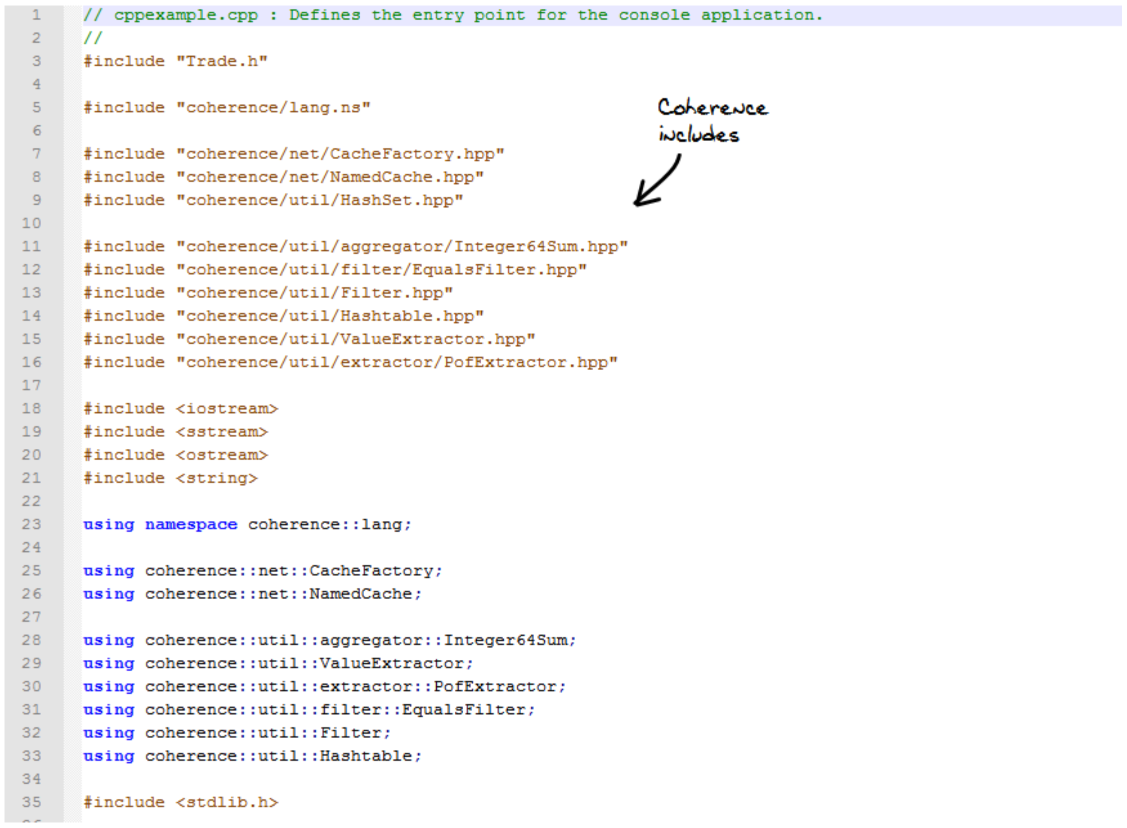
The serialize method serializes the variables in order: `symbol`, `price`, `id`, and `quantity`. The deserialize method deserializes the variables in the same order.

The serialize method performs the serialization of the passed in constant `Trade` object reference directly to the `hOut` stream. The deserialization method processes the passed `hIn` stream to get the original variable values, then creates a new `Trade` object with those values, and returns the `Trade` object.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

11-3.8 Open the `cppexample.cpp` file and examine its code. This is the code that serves as the main entry point into the `Trade.exe` executable, which uses the Coherence*Extend C++ API to store and retrieve `Trade` objects into/from the cache using POF serialization. The screen shots below illustrate what the C++ API is doing:



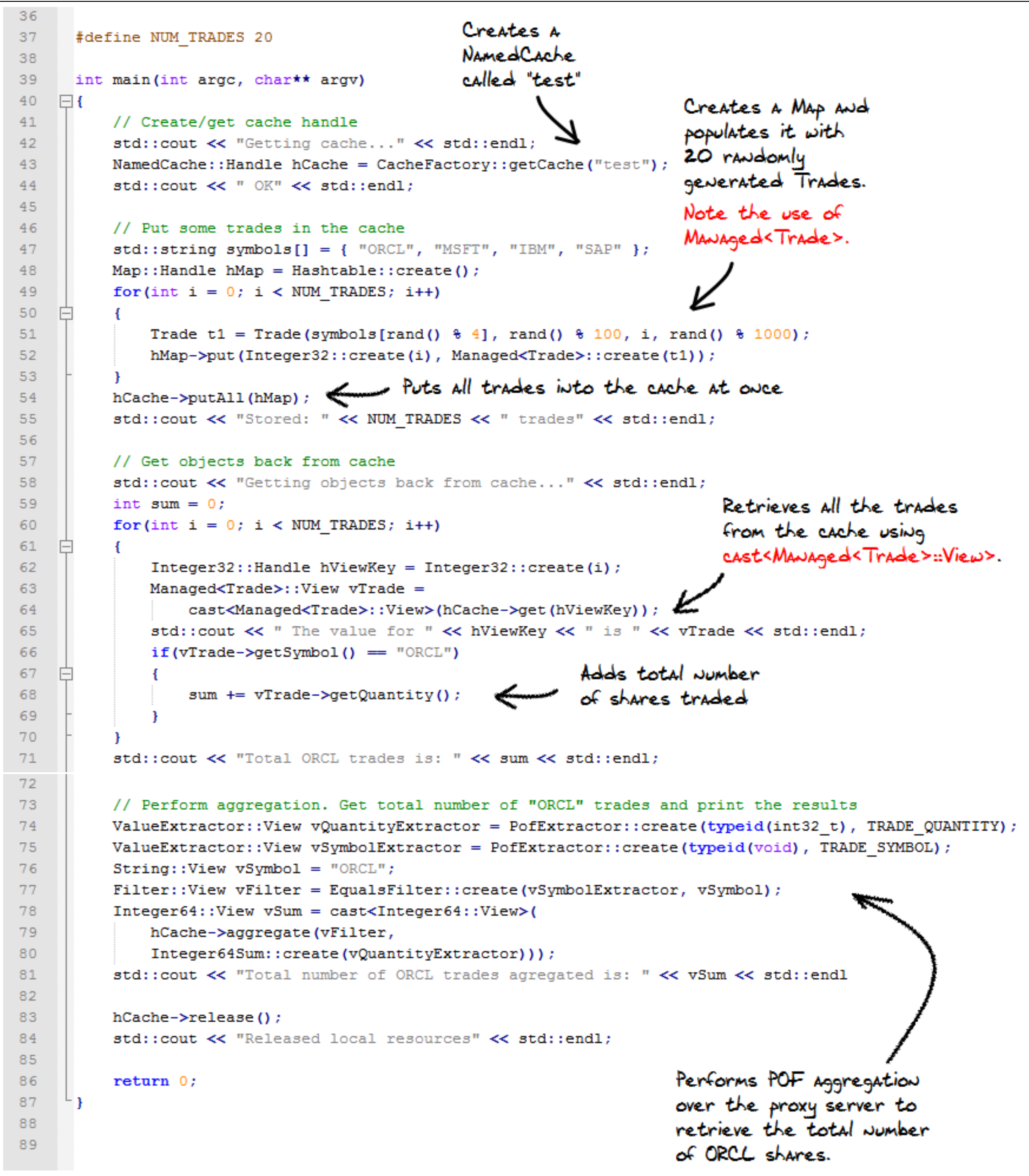
```
// cppexample.cpp : Defines the entry point for the console application.
//
#include "Trade.h"

#include "coherence/lang.ns"

#include "coherence/net/CacheFactory.hpp"
#include "coherence/net/NamedCache.hpp"
#include "coherence/util/HashSet.hpp"

#include "coherence/util/aggregator/Integer64Sum.hpp"
#include "coherence/util/filter/EqualsFilter.hpp"
#include "coherence/util/Filter.hpp"
#include "coherence/util/Hashtable.hpp"
#include "coherence/util/ValueExtractor.hpp"
#include "coherence/util/extractor/PofExtractor.hpp"

#include <iostream>
#include <sstream>
#include <ostream>
#include <string>

using namespace coherence::lang;

using coherence::net::CacheFactory;
using coherence::net::NamedCache;

using coherence::util::aggregator::Integer64Sum;
using coherence::util::ValueExtractor;
using coherence::util::extractor::PofExtractor;
using coherence::util::filter::EqualsFilter;
using coherence::util::Filter;
using coherence::util::Hashtable;

#include <stdlib.h>
```

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only



```
#define NUM_TRADES 20

int main(int argc, char** argv)
{
    // Create/get cache handle
    std::cout << "Getting cache..." << std::endl;
    NamedCache::Handle hCache = CacheFactory::getCache("test");
    std::cout << " OK" << std::endl;

    // Put some trades in the cache
    std::string symbols[] = { "ORCL", "MSFT", "IBM", "SAP" };
    Map::Handle hMap = Hashtable::create();
    for(int i = 0; i < NUM_TRADES; i++)
    {
        Trade t1 = Trade(symbols[rand() % 4], rand() % 100, i, rand() % 1000);
        hMap->put(Integer32::create(i), Managed<Trade>::create(t1));
    }
    hCache->putAll(hMap);
    std::cout << "Stored: " << NUM_TRADES << " trades" << std::endl;

    // Get objects back from cache
    std::cout << "Getting objects back from cache..." << std::endl;
    int sum = 0;
    for(int i = 0; i < NUM_TRADES; i++)
    {
        Integer32::Handle hViewKey = Integer32::create(i);
        Managed<Trade>::View vTrade =
            cast<Managed<Trade>::View>(hCache->get(hViewKey));
        std::cout << " The value for " << hViewKey << " is " << vTrade << std::endl;
        if(vTrade->getSymbol() == "ORCL")
        {
            sum += vTrade->getQuantity();
        }
    }
    std::cout << "Total ORCL trades is: " << sum << std::endl;

    // Perform aggregation. Get total number of "ORCL" trades and print the results
    ValueExtractor::View vQuantityExtractor = PofExtractor::create(typeid(int32_t), TRADE_QUANTITY);
    ValueExtractor::View vSymbolExtractor = PofExtractor::create(typeid(void), TRADE_SYMBOL);
    String::View vSymbol = "ORCL";
    Filter::View vFilter = EqualsFilter::create(vSymbolExtractor, vSymbol);
    Integer64::View vSum = cast<Integer64::View>(
        hCache->aggregate(vFilter,
        Integer64Sum::create(vQuantityExtractor)));
    std::cout << "Total number of ORCL trades agregated is: " << vSum << std::endl

    hCache->release();
    std::cout << "Released local resources" << std::endl;

    return 0;
}
```

Creates a NamedCache called "test"

Creates a Map and populates it with 20 randomly generated Trades. Note the use of Managed<Trade>.

Puts all trades into the cache at once

Retrieves all the trades from the cache using cast<Managed<Trade>::View>.

Adds total number of shares traded

Performs POF aggregation over the proxy server to retrieve the total number of ORCL shares.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Run a Coherence*Extend C++ POF client

11-3.9 Within Eclipse, double-click the `runexample.cmd` file to run the program.

HINT: If you get an error that says a file cannot be executed, you must manually install the Microsoft Visual Studio redistribution package to ensure that the proper runtime libraries are installed on the machine. This is done by double-clicking the **`D:\Oracle\student\software\vcredist_x86_2005_SP1_wSec_Upt.exe`** file, accepting the license agreement, and allowing the setup program to finish running. Then try to run the lab again.

You should see output similar to:

```
Config File: config\extend-cache-config.xml
Getting cache...
Oracle Coherence for C++ 3.6.0.0 DPR4 <D5>: Optional operational
override "tangosol-coherence-override.xml" is not specified, using
default

Oracle Coherence for C++ Version 3.6.0.0 Build 17229
 RTC Microsoft Windows x86 Release msvc 2005 build
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

2010-07-06 16:26:20.739/0.625 Oracle Coherence for C++ RTC 3.6.0.0
DPR4 <D5> (thread=main): Loaded cache configuration from
"..\config\extend-cache-config.xml"
2010-07-06 16:26:20.833/0.703 Oracle Coherence for C++ RTC 3.6.0.0
DPR4 <D5>
(thread=ExtendTcpCacheService:coherence::component::util::TcpInitiato
r):Started:coherence::component::util::TcpInitiator@006C95C8{Name=Ext
endTcpCacheService:coherence::component::util::TcpInitiator,
State=(SERVICE_STARTED), ThreadCount=0,
Codec=coherence::component::net::extend::PofCodec@006CFDF8{Format=POF
}, PingInterval=0, PingTimeout=0, RequestTimeout=0, ConnectTimeout=0,
RemoteAddresses=[ARCYNE/141.144.96.231:9099], KeepAliveEnabled=0,
TcpDelayEnabled=0, ReceiveBufferSize=0, SendBufferSize=0,
LingerTimeout=0}
2010-07-06 16:26:20.864/0.734 Oracle Coherence for C++ RTC 3.6.0.0
DPR4 <D5> (thread=main): Opening Socket connection to
ARCYNE/141.144.96.231:9099
2010-07-06 16:26:20.864/0.734 Oracle Coherence for C++ RTC 3.6.0.0
DPR4 <Info> (thread=main): Connected to ARCYNE/141.144.96.231:9099
 OK
Stored: 20 trades
Getting objects back from cache...
 The value for 0 is Trade(Symbol=IBM, Price=67, Id=0, Quantity=41)
 The value for 1 is Trade(Symbol=ORCL, Price=69, Id=1, Quantity=500)
 The value for 2 is Trade(Symbol=IBM, Price=58, Id=2, Quantity=478)
 The value for 3 is Trade(Symbol=MSFT, Price=5, Id=3, Quantity=464)
 The value for 4 is Trade(Symbol=MSFT, Price=27, Id=4, Quantity=281)
 The value for 5 is Trade(Symbol=IBM, Price=95, Id=5, Quantity=491)
 The value for 6 is Trade(Symbol=SAP, Price=36, Id=6, Quantity=827)
 The value for 7 is Trade(Symbol=MSFT, Price=2, Id=7, Quantity=604)
 The value for 8 is Trade(Symbol=MSFT, Price=82, Id=8, Quantity=292)
```

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

```
 The value for 9 is Trade(Symbol=SAP, Price=18, Id=9, Quantity=716)
 The value for 10 is Trade(Symbol=SAP, Price=26, Id=10, Quantity=447)
 The value for 11 is Trade(Symbol=ORCL, Price=69, Id=11,
Quantity=538)
 The value for 12 is Trade(Symbol=SAP, Price=99, Id=12, Quantity=667)
 The value for 13 is Trade(Symbol=SAP, Price=3, Id=13, Quantity=894)
 The value for 14 is Trade(Symbol=MSFT, Price=33, Id=14,
Quantity=322)
 The value for 15 is Trade(Symbol=SAP, Price=41, Id=15, Quantity=664)
 The value for 16 is Trade(Symbol=SAP, Price=68, Id=16, Quantity=253)
 The value for 17 is Trade(Symbol=MSFT, Price=62, Id=17,
Quantity=644)
 The value for 18 is Trade(Symbol=SAP, Price=59, Id=18, Quantity=37)
 The value for 19 is Trade(Symbol=IBM, Price=29, Id=19, Quantity=741)
Total ORCL trades is: 1038
Total number of ORCL trades agregated is: 1038
Released local resources
Press any key to continue . . .
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Analyze local cache execution results	
11-3.10	Take a closer look at the output in the command window. The first thing you should notice is the line that indicates that the C++ product is in use: `Oracle Coherence for C++ 3.6.0.0 Build 17229`
11-3.11	The next thing you should find in the window is the line showing that your cache configuration file was loaded (it lists the `remote-cache-scheme` configured.)
11-3.12	Next, you want to look for messages that have the word “ExtendTcpCacheService” in them. This shows that the remote TCP cache service was started, and that it connected to the remote proxy server. This is a positive indication that a Coherence*Extend configuration is in use.
11-3.13	Now that you have verified that the C++ client is running and using your proxy configuration, you can analyze the output of the actual Trade application. It should show a series of 20 trades that were performed, followed by two total lines that show the sum of all ORCL trades that were performed. One calculation was performed while looping through the cache entries during retrieval from the cache, and the second was calculated using `PofExtractors` to get the `TRADE_QUANTITY` and `TRADE_SYMBOL` data from the cache, and performing an aggregation on the remote cluster to return the sum. If everything was successful, regardless of how the data was calculated, both values should match. You have configured and run a Coherence*Extend C++ POF client, and verified its behavior. **Note:** Close all command windows before proceeding.

Solution Instructions

Since there are no lab steps that require making changes to source or configuration files in Practice.11.03, there is no separate solution project for this lab.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Practices for Lesson 12

Chapter 13

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practices for Lesson 12

Practices Overview

Developers and administrators need to know how their Coherence clusters are performing. This requires setting up the JMX management infrastructure provided by Coherence to expose the runtime management attributes of a running cluster. Administrators must also know how to configure and use the Reporter feature of Coherence that provides default reports for the various critical components involved in running a cluster.

The objective of this practice is to assist the developer or administrator with using the management tools supplied by Coherence.

In this practice, you will perform the following tasks:

- Configuring the Reporter
- Running the Reporter

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Practice 12-1: Configuring and Running the Reporter

Skills Learned

At the end of this practice, you should be able to:

- Configure a Coherence JMX management node.
- Configure the Reporter to run on the management node.
- Configure a cluster node to run as a managed node.
- Change settings that control how the Reporter runs.
- Identify and write a Reporter report group batch file.
- Start a Coherence cluster that includes a management server.
- Connect a JMX console to a Coherence management server to analyze Coherence MBean attributes and operations.
- Start and stop the Reporter from a JMX console.
- Generate default reports using the Reporter.
- View reports generated by the Reporter.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Problem Statement

From: ted.ious@retail-is-us.com
Sent: Thuesday, July 23rd, 12:02pm
To: you@acmedev.com
Subject: Coherence Management Features

Hello!

Our planned production Coherence cluster for retail-is-us.com requires some operational procedures for monitoring its performance. We would like to use the Coherence JMX management infrastructure, the Reporter, and the JConsole JMX console to get started with Coherence monitoring features.

I understand that using Coherence management features require:

1. Configuring a management node that exposes JMX data
2. Configuring the Reporter on the management node which will collect metrics for the entire cluster
3. Configuring the other nodes in the cluster to enable them to be managed
4. Running the management server as part of the Coherence cluster
5. Connecting JConsole to the management server in order to:
    a. View Coherence MBean data
    b. View Reporter MBean data

With this in mind, we need you to create a basic example that can get us started down our Coherence management path. Our engineering staff has provided a baseline Eclipse project which hopefully can be adapted to serve this purpose.

We look forward to seeing the example you create.

Regards,

Ted Ious

Ted Ious
Retail Is Us
ted.ious@retail-is-us.com

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Design

In this practice, you will configure a JMX management server that is also configured to run the Reporter, and configure a managed node that managed by your management server. The Reporter starts when the managed server starts because the `-Dtangosol.coherence.management.report.autostart` system property is set to `true`. When the Reporter starts, it uses the `coherence.jar!reports/report-all.xml` file to determine which reports to run because the `-Dtangosol.coherence.management.report.configuration` system property is set to `reports/report-all.xml`. Then you configure the Reporter to run the report every 25 seconds by changing the `<frequency>` element within the `report-all.xml` file. Next, you run the two-node cluster, connect JConsole to the management server, and analyze the MBean attributes and operations available in Coherence. Finally, you examine the reports that are generated by the Reporter.

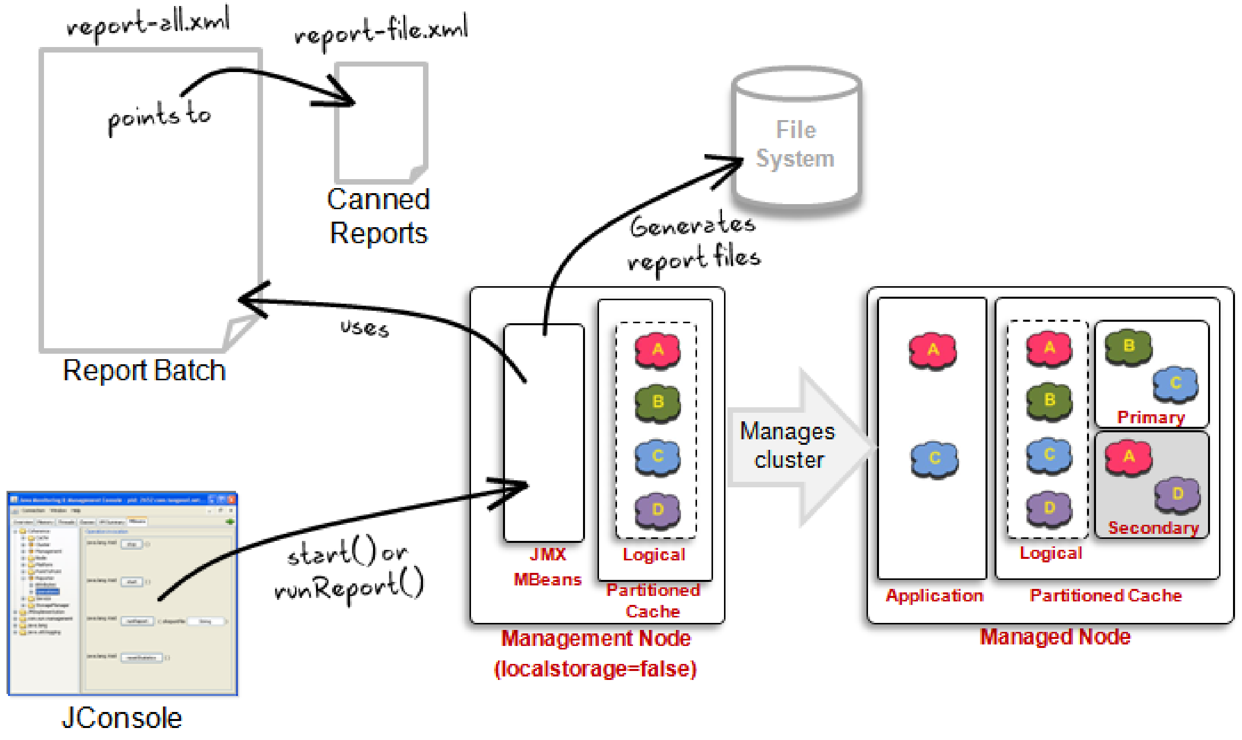


Figure 1 Reporter Lab Architecture

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Resources and References

Resource Directories		
Practice Resources	`D:\oracle\student\`**`resources`**`\`	Contains a set of practice specific resource directories
Support	`D:\oracle\student\`**`support`**	Contains various jar files used by the practices
Software	`D:\oracle\student\`**`software`**	Contains installable Coherence and other files

Table 12.1

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Overview Instructions

1	Configure JMX and Reporter features.
2	Start a management-enabled cluster, and use JConsole to use management features.
3	View Coherence MBean data and reports generated by the Reporter.

Hints:

- You can identify Reporter system properties versus Management system properties by seeing if the word 'report' is in the full property name.
- `-Dtangosol.coherence.management` configures a management server to enable all monitoring.
- `-Dcom.sun.management.jmxremote` configures the Java 6 management infrastructure so JConsole can connect to the JVM.
- `-Dtangosol.coherence.management.remote` configures a cluster node so that it allows remote management of its MBeans.
- `-Dtangosol.coherence.management.report.autostart` instructs Coherence to automatically start the Reporter when the node is started.
- `-Dtangosol.coherence.management.report.configuration` instructs the Reporter to use a particular report batch file.
- Reports run every 25 seconds, or can be run on demand using the Reporter's MBean `runReport` operation.
- Reports are generated into the current running folder (the main project folder in Eclipse).

Detailed Instructions

Configure JMX and Reporter features	
12-1.1	Within Eclipse, expand project **Practice.12.01** and expand the **src** folder to display its contents.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

12-1.2 From within Eclipse:

1. Make sure the **Practice.12.01** project is selected.
2. Click the **arrow** next to the icon for run configurations in the upper left side of the IDE.
3. Select **Run Configurations…**
4. On the **Run Configurations** screen, select **Java Application** > **Practice.12.01.ManagementServer** in the left window.
5. On the right side of the dialog, select the **Main** tab if it is not already open. Take note that this run configuration is going to run the Coherence **DefaultCacheServer**. This exercise is to set certain options that cause it to run as a Coherence management node.
6. Select the **Arguments** tab to display the arguments used for running the node.
7. Within the **VM arguments** input box, enter the settings that instruct the Coherence node to run as a management server that manages MBeans on local and remote cluster nodes, and configures the node to allow JConsole connections.

Note: For convenience the code snippets for this exercise can be found in the `resources` subdirectory for **Practice.12.01** in the **`code.snippet.txt`** file.

See the table below for more information about each system property:

System Property	Description
`-Dtangosol.coherence.management=all`	Configures the cluster node to manage the specified MBeans for the cluster. Possible settings are: • `none` • `local-only` • `remote-only` • `all`
`-Dcom.sun.management.jmxremote`	Configures the node to use the management infrastructure built into Java 1.6, which enables JConsole to connect to the node and view MBean data, as well as other information.

When finished, click the **Apply** button to apply the settings. Your view should resemble:

```
-Dtangosol.coherence.management=all
-Dcom.sun.management.jmxremote
```

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

12-1.3 This exercise is to set certain options that will cause the node to run as a Coherence managed node, that will in turn provide data back to the management node. While still in the **Run Configurations** dialog:

1. Select the **Practice.12.01.ManagedCacheServer** in the left window.
2. On the right side of the dialog, select the **Main** tab if it is not already open. Take note that this run configuration is going to run the CacheTester program that is part of this practice's source code.
3. Select the **Arguments** tab to display the arguments used for running the node.

Within the **VM arguments** input box, enter the settings that instruct the Coherence node to run as a managed node that allows remote management of its MBeans.

See the table below for more information about each system property:

System Property	Description
`-Dtangosol.coherence.management.remote=true`	Enables remote management of a Coherence node's MBeans.

When finished, click the **Apply** button to apply the settings. Your view should resemble:

```
-Dtangosol.coherence.management.remote=true
```

12-1.4 **For Your Information:** The last two steps show the basic settings for how to configure a Coherence JMX management node, and a cluster node that allows remote management of its MBeans. For this exercise, the Reporter still needs to be configured. These steps separate the different configurations to differentiate between JMX and Reporter settings. A Coherence node will run as a JMX node without configuring the Reporter. The Reporter cannot run without JMX configured.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

12-1.5 Now it is time to configure the Reporter.
While still in the **Run Configurations** dialog:

1. Select the **Practice.12.01.ManagementServer** in the left window.
2. Select the **Arguments** tab if it is not already open.
3. Within the **VM arguments** input box, add the settings that instruct the Coherence node to run as a managed node that allows remote management of its MBeans.

See the table below for more information about each system property:

System Property	Description
`-Dtangosol.coherence.management.report.autostart=true`	Instructs Coherence to automatically start the Reporter when the node is started
`-Dtangosol.coherence.management.report.configuration=reports/report-all.xml`	By default, the Reporter uses a report-group configuration file located in the `coherence.jar` file, `reports/report-group.xml`. This file contains the configuration that references a set of default reports that the Reporter can run. This report group file can be overridden using this system property. There is another report-group file in `coherence.jar`: `reports/report-all.xml`. This file references more default reports than the default file provides. Set this property to point to the `report-all.xml` file to run all of these default reports.

When finished, click the **Apply** button to apply the settings.

Note: You can differentiate the Reporter properties from JMX settings by finding the word, **report**, in the fully-qualified property name.

Your view should resemble:

```
-Dtangosol.coherence.management.report.configuration=
reports/report-all.xml
-Dtangosol.coherence.management.report.autostart=true
```

Click the **Close** button to exit the Run Configurations dialog window.

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

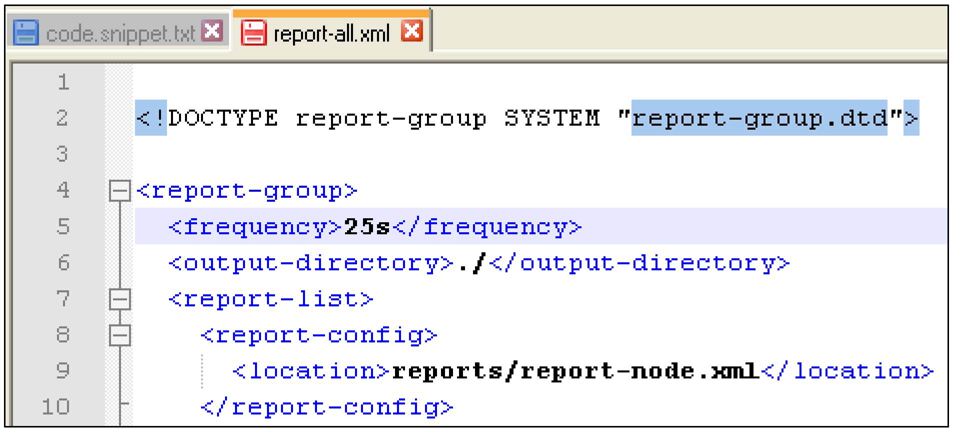
12-1.6 Make sure that there are no Coherence nodes running on the machine.
By default, the Reporter group files are configured to run every minute (`1m`). This can be changed by opening the file within the `coherence.jar` file and changing the frequency setting. There are a couple of ways this can be done:

a. Change the extension of `coherence.jar` to `.zip`, open the file with Winzip, make the changes using a text editor, saving the file, let Winzip automatically update the file in the archive, and change the extension back to `.jar`.

b. Change the Windows file association for `.jar` files to open using Winzip instead. Open `coherence.jar` with Winzip, make the changes using a text editor, saving the file, and let Winzip automatically update the file in the archive.

c. Extract the `reports/report-all.xml` file to the local disk. Modify the file in place, and add it to your classpath before `coherence.jar`. This may be difficult in Eclipse, so you should only resort to this method if you are experienced with how to do this.

Once you are editing the `report-all.xml` file, change the `frequency` setting to **25s** and save your changes.



```

<!DOCTYPE report-group SYSTEM "report-group.dtd">

<report-group>
  <frequency>25s</frequency>
  <output-directory>./</output-directory>
  <report-list>
    <report-config>
      <location>reports/report-node.xml</location>
    </report-config>
```

12-1.7 *Optionally*, take some time to open some of the default report files that the `report-all.xml` file references to see an example of how reports are configured.
Now the JMX and Reporter settings are configured. Next is a look at the code for the managed cluster node, running the cluster, and inspecting Coherence MBeans using JConsole.

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Start a management-enabled cluster, and use JConsole to use management features	
12-1.8	Ensure that project **Practice.12.01** is selected within Eclipse.
12-1.9	Open the `CacheTester.java` file to review how it works. The program: 1. Starts a loop that executes five times. a. Uses the Loader program to load the cache with the retail domain object model. b. Sleeps for 30 seconds. 2. Gets the orders cache and performs `get()` requests on all orders in the cache to generate some statistics. 3. Stops while awaiting input on stdin to keep the process running. The loop is designed to cause the cache to have more data after each iteration. The Reporter runs the reports every 25 seconds, while the `CacheTester` code is sleeping for 30 seconds in between each time it loads the cache. This should produce different data for each run of the Reporter. The next few steps will instruct you how to run this scenario.
12-1.10	Start the management node from within Eclipse: 1. Make sure the **Practice.12.01** project is selected. 2. Open the **Run Configurations** dialog again. 3. Select **Practice.12.01.ManagementServer**, and then click the **Run** button. This starts a cache server and displays its output within the Eclipse IDE in the Console tab. Wait until you see the "Started DefaultCacheServer" message before moving on to the next step. **Note:** In this case, the management node runs as a storage-enabled node due to the simplicity of the example. In a real-world environment, the management node is usually storage-disabled.
12-1.11	Start the managednode from within Eclipse: 1. Make sure the **Practice.12.01** project is selected. 2. Open the **Run Configurations** dialog again. 3. Select **Practice.12.01.ManagedCacheServer**, and then click the **Run** button. This starts a cache server and displays its output within the Eclipse IDE in the Console tab. Wait until you see the "Press enter when done" message before moving on to the next step. **Note:** There is not much to see in the output windows for this lab. The visibility of data for this lab is found by using JConsole to view live runtime data, and generated Reporter files that contained the results of running the default reports.

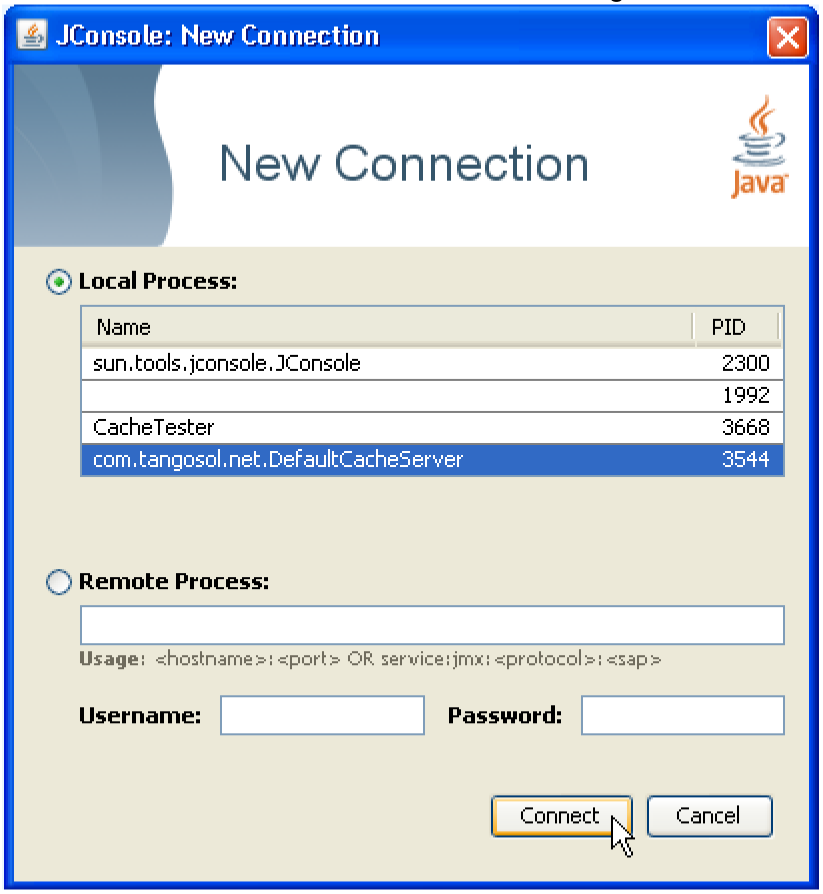
Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

12-1.12 Open a Windows command shell, and run JConsole by typing the following:

```
jconsole
```

You should see a window resembles the following:



[✓ Local Process:]

Name	PID
sun.tools.jconsole.JConsole	2300
	1992
CacheTester	3668
com.tangosol.net.DefaultCacheServer	3544

Select the **DefaultCacheServer** entry as it is configured as the management node, and click the **Connect** button to connect.

Now that you have successfully connected JConsole to the management node, take a closer look at some of the MBeans within Coherence and the Reporter.

Oracle University and ORACLE CORPORATION use only

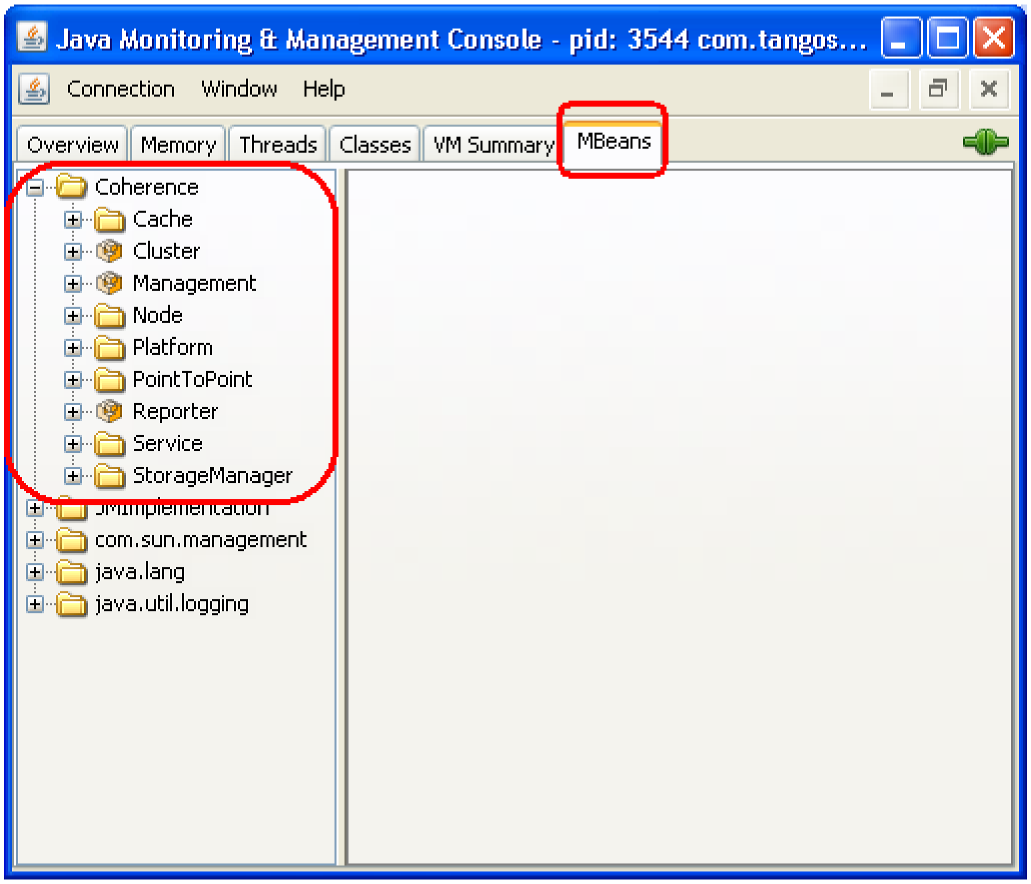
Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

View Coherence MBean data and reports generated by the Reporter

12-1.13 JConsole provides a lot more information than just MBean data that is running within the current JVM. The opening JConsole screen shows various other statistics for the running JVM. Select the **MBeans** tab to open the MBean browser view. You view should resemble:

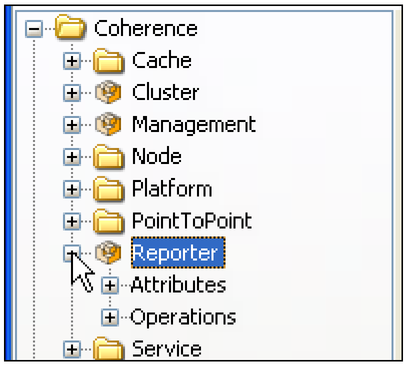


Take a few moments to explore the various MBean that are exposed by Coherence:

- Within the Cache tree Coherence lists the names of each cache in the cluster, and provides some data points for each.
- The Cluster MBean attributes provide important information about the cluster, and operations for shutting down and ensuring that the cluster is running.
- The Node tree provides information for each node running in the cluster. This is where you would check for Publisher and Receiver SuccessRate statistics, as well as some of the other statistics shared in the presentation.
- The Reporter MBean is where the rest of this lab takes place. It contains the attributes and operations for the Reporter feature.
- The Service MBean provides statistics about each of the Coherence services running in the cluster.
- The StorageManager MBean provides statistics related to the storage of cache entries on a per-cache basis.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

12-1.14 Expand the **Reporter** MBean view to reveal the attributes and operations nodes in the left tree.



Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

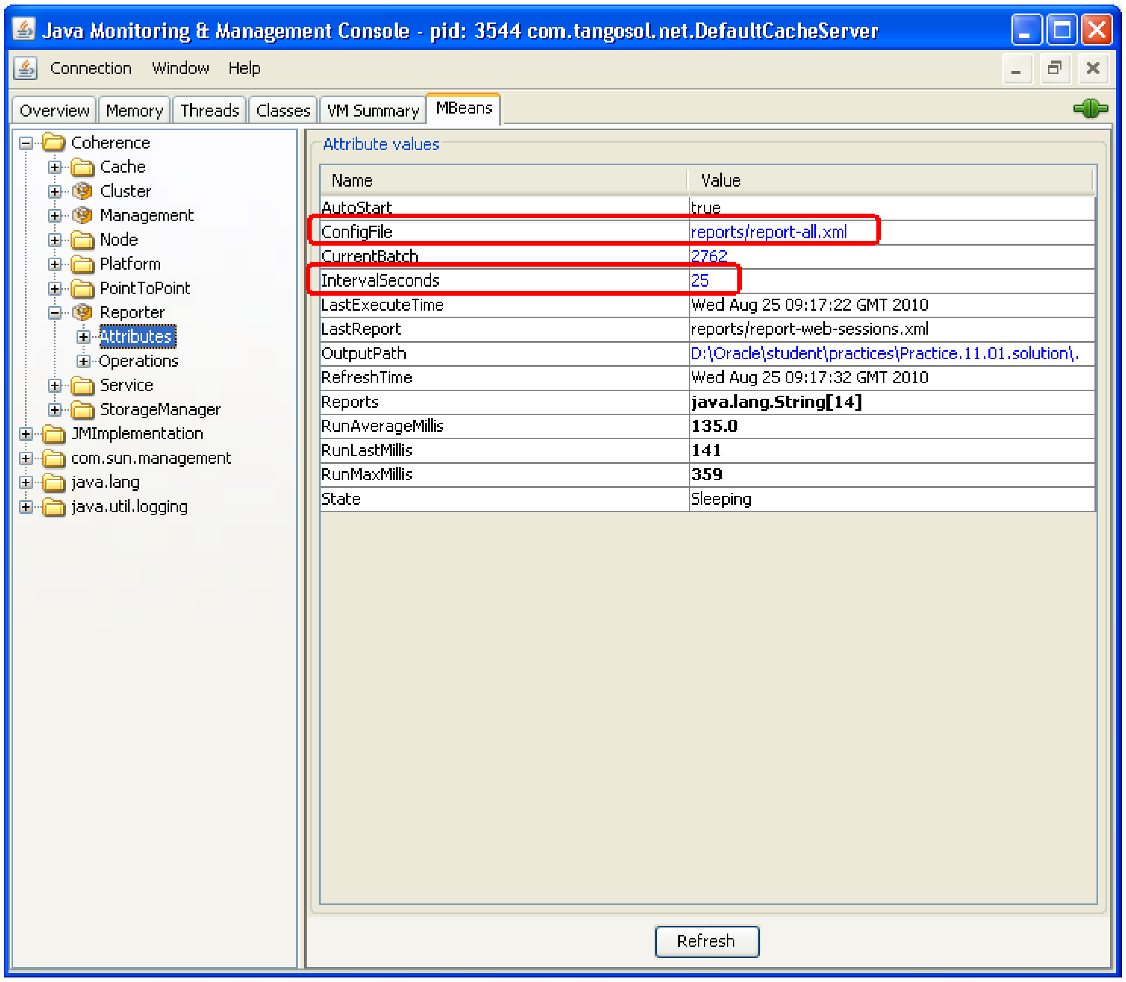
12-1.15 Select the **attributes** node to reveal the Reporter MBean attributes on the right side of the window. You should be able to see the settings that you configured for the Reporter here. The **ConfigFile** attribute should be set to your property setting:

```
-Dtangosol.coherence.management.report.configuration
```

The **IntervalSeconds** attribute should be set to **25s** as configured in the **`reports/report-all.xml`** file. You can also manually change this attribute dynamically. If using Java 1.6, you may have to specify that the type is `long` when entering a new value. For example, enter **30L** to set the attribute to 30 seconds. This is a bug in JConsole.

By default, the Reporter generates files into the current working directory of the management node. This value can be changed here by setting a new **OutputPath**.

Your screen should resemble:



Name	Value
AutoStart	true
ConfigFile	reports/report-all.xml
CurrentBatch	2762
IntervalSeconds	25
LastExecuteTime	Wed Aug 25 09:17:22 GMT 2010
LastReport	reports/report-web-sessions.xml
OutputPath	D:\Oracle\student\practices\Practice.11.01.solution\.
RefreshTime	Wed Aug 25 09:17:32 GMT 2010
Reports	java.lang.String[14]
RunAverageMillis	135.0
RunLastMillis	141
RunMaxMillis	359
State	Sleeping

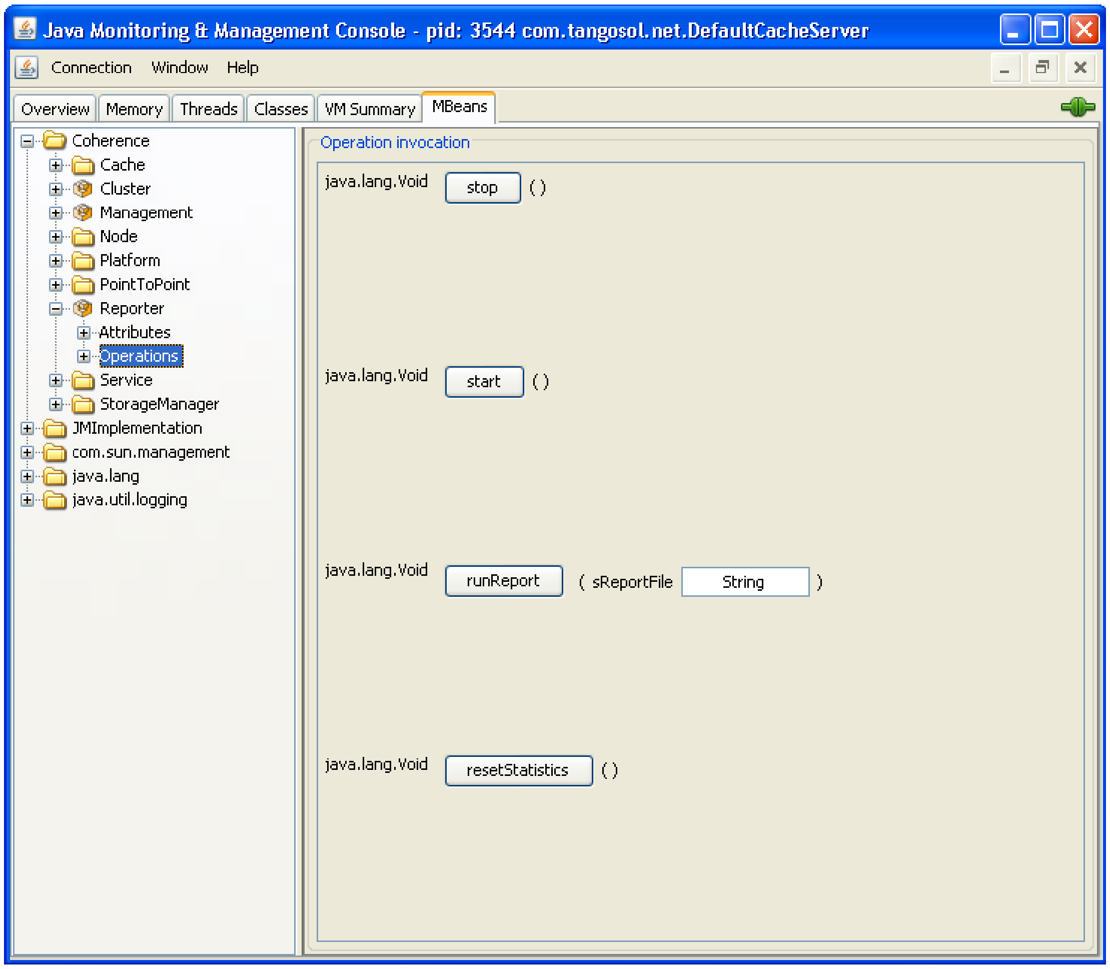
Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

12-1.16 Select the **operations** node to reveal the Reporter MBean operations on the right side of the window. You should be able to see the four operations available for the Reporter MBean:

Operation	Description
`stop()`	Stop the Reporter from generating reports.
`start()`	Start the Reporter to begin generating reports.
`runreport()`	Run a specified report. For example, `reports/report-group.xml`. This can be any report-group file in `coherence.jar` or a file URL for use with the filesystem.
`resetStatistics()`	Reset the runtime performance statistics for the Reporter.

Your screen should resemble:



Now that you have learned how to interact with Reporter MBean data using JConsole, take a look at the report files generated by the Reporter.

Oracle University and ORACLE CORPORATION use only

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

12-1.17	Using Windows Explorer, browse to `D:\Oracle\student\practices\Practice.12.01`, which is where the files should be located. The files are prefixed with a date and time that follows a `YYYYMMDDHH` format. The rest of the file name says more about what type of report is contained in the file. For example, `cache-size` indicates that the report contains metrics related to the size of the caches at runtime.
12-1.18	Double-click the `cache-size.txt` file to open it in a text editor. The file is tab-delimited by default, and in most cases would be opened in Microsoft Excel for further processing, such as generating charts like the ones shown in the lesson slides. The data contained in these files could also be used programmatically for any need that may arise. The point is that the JMX hooks and the Reporter-generated metrics provide insight into the runtime details of what is happening inside a Coherence cluster. The JMX infrastructure provides real-time metrics, and the Reporter can provide historical metrics for analyzing trends. **Note:** Within a report file, there is a column labeled as "Batch". Every time the Reporter runs the report (as specified by the `frequency` setting), it groups that data into a batch, or instance, of running that report. All rows with the same batch number are related. Take a few moments to open and look at the other files created by the Reporter.
12-1.19	**Note:** Close all command windows before proceeding. You can do this using the Terminate icon 🟥 in the Eclipse Console tab.

Solution Instructions

To use the solution, start Eclipse and open the `D:\Oracle\student\practices` workspace, and open **Practice.12.01.solution**. Follow the directions for testing the application from step **12-1.6**, ignoring any steps to add code, all of which is in the solution.

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and ORACLE CORPORATION use only

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.