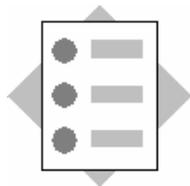


TEP12: Exercise

Exercise 06-04: Identify a Deadlock Situation

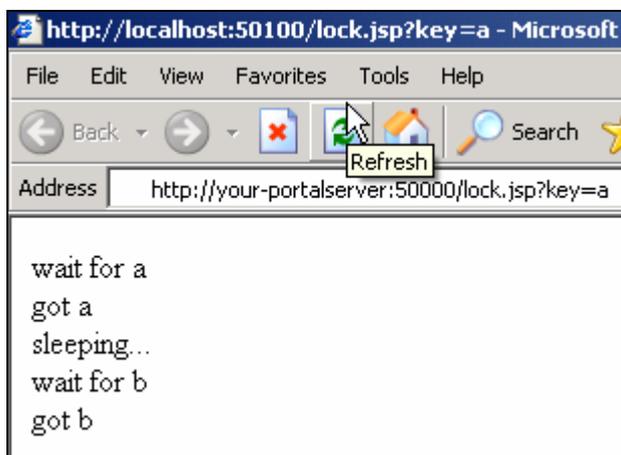


At the conclusion of this exercise, you will be able to:

- Identify the cause of deadlocks that block threads

- 1-1 Copy the file “go.console_output_redirected.bat” from the training share to the directory “\<J2EE_directory>\cluster\server\” on your portal server. Shutdown the server node of your training portal. Start the server node by executing “go.console_output_redirected.bat”.
- 1-2 Copy the file “lock.jsp” from the training share to the directory “\<J2EE_directory>\docs\” on your portal server. Open a browser window and request <http://<your-portalserver>:50000/lock.jsp?key=a>. Open a new browser instance with “CTRL-N” and there change the parameter “key=a” to “key=b” so that you have 2 browser windows open – one with “key=a” and one with “key=b”.

Your browser windows should look like this:



- 1-3 (If you have already installed jEdit in a previous exercise, then you can skip this paragraph.) Execute the “jedit42pre13install.jar”. Just double click the JAR-file to install it. (In case there are problems with the installation of jEdit, you can also skip this step and proceed using Notepad or Wordpad instead.)
- 1-4 Now you have to reload both browser windows where “lock.jsp?key=[a|b]” is opened, as quickly as possible. To do this just click the reload button of the browser quickly in both windows. (It should be done in no longer then 2 seconds for both windows.)

TEP12: Exercise

You successfully created a deadlock between threads when the browser windows are not loaded again any more and the windows are just empty. If the text (as in the above screenshot) is displayed again, then you have to repeat the reload faster.

- 1-5 Do a thread dump by pressing “CTRL-BREAK” in the console window of the server node.
- 1-6 Open the file manager on your portal server and navigate to “\<J2EE_directory>\”. Click right on the file “console_output_redirected.log” and select “Open with jEdit”.
- 1-7 Search in jEdit for the string “FOUND A JAVA LEVEL DEADLOCK”. You should find a message like this one:

```
FOUND A JAVA LEVEL DEADLOCK:
```

```
-----
```

```
"Client_Thread_31":  
  waiting to lock monitor 0x912b2c (object 0x110bc0a0, a java.lang.Integer),  
  which is locked by "Client_Thread_30"  
"Client_Thread_30":  
  waiting to lock monitor 0x91288c (object 0x110bc0b0, a java.lang.Integer),  
  which is locked by "Client_Thread_31"
```

This is the J2EE engine’s internal deadlock detection.

Below of this message you should find the stack trace of the 2 threads where you can see that the execution of the “lock.jsp” caused the deadlock.