

AVP-3225

Java Troubleshooting with ISA 5 using Thread and Monitor Dump Analyzer



Java Troubleshooting with ISA 5 using Thread and Monitor Dump Analyzer

•	What this lab is about	2
•	Lab requirements	3
•	What you should be able to do	3
•	Part 1:Lab Set Up	4
•	Part 2:IBM Support Assistant 5 Beta	5
•	Part 3:Advanced Javacore Analysis with IBM Thread and Monitor Dump Analyzer Tool	.20
•	Part 4:(Optional) Mapping a Thread Id in a Javadump to the output of WebSphere logging or trac 57	е.
•	Reference Links	. 61

What this lab is about

This lab is provided AS-IS, with no formal IBM support.

In this lab you will use IBM Support Assistant 5 Beta (ISA 5) to diagnose JVM issues experienced by a WebSphere Application Server 8.5.

A badly implemented web application will be used to simulate a hung thread scenario. Diagnostic data such as Javadumps (javacore.txt files) and system dumps will be used to determine the code causing the issue, and inspect the Java objects associated with the hung thread. The lab will also explain how to correlate thread id between the various logs and dumps.

The lab demonstrates how ISA 5 facilitates team team-based collaboration, and provides server-level diagnostic tools to carry out analysis. The ISA tools "Thread and Monitor Dump Analyzer (TMDA)" and "Memory Analyzer" will be used.



Lab requirements

List of system and software required for the attendee to complete the lab.

- WebSphere Application Server V8.5.0.1
- IBM Support Assistant V5 (Beta 2) with the following tools installed:
 - ► WebSphere Application Server Configuration Visualizer
 - ► Thread Dump and Monitor Analyzer (ISA 5 desktop tool, and report)
 - IBM Monitoring and Diagnostic Tools for Java[™] Memory Analyzer (ISA 5 desktop tool, and report tool)
 - IBM Extensions for Memory Analyzer (Packaged with the Memory Analyzer ISA 5 desktop tool)

What you should be able to do

At the end of this lab you should be able to:-

- Use ISA 5 to manage Java diagnostic data and utilize problem determination tools
- Diagnose a hung WebSphere thread using Thread and Monitor Dump Analyzer for Java
- Correlate thread information between WebSphere log, traces, Javadumps and system dumps



Part 1: Lab Set Up

_ Login to the VMWare image with the username/password below:

Username : Administrator

Password : Impact2013AVP

NOTE:

Due to the physical memory on the VMware image being used for this lab please understand that certain operations may take time to perform – please be patient. The expected duration of this lab is **55 minutes** and each part has an estimated duration. The final lab section is marked as **optional** so you can attempt this if there is sufficient time.



Part 2: IBM Support Assistant 5 Beta

Note:

The IBM® Support Assistant (ISA) is a free application that provides features for problem determination, and a platform for obtaining diagnostic tools. The most recent release of ISA (version 5.0), which is currently in beta, brings these capabilities into a server environment. This enables an administrator to install a single instance of ISA that can be used by a group of users and accessed via a web browser. Therefore resources, files, information, and server-level tools can be shared. This facilitates team based collaboration and avoids the need for each team member to install diagnostic tools on their local workstation.

ISA v5.0 can be installed from an EAR file into an existing WebSphere Application Server, or using a simple "all-in-one" unzip install which contains everything required, including a lightweight application server and Java runtime. This lab uses the latter approach.

In this part of the lab, you will use ISA to understand its key concepts, and test some of the core functions.

_____ Validate that ISA is fully started. ISA was started in a command prompt window. Check that window and look for "IBM Support Assistant is ready to run". If this is not displayed wait for the startup to complete.

Image: Start IBM Support Assistant 5(2)
Image: Starting IBM Support Assistant v5.0 Beta 2...
Server isa started.
Starting Memory Analyzer...
Server mat started.
Memory Analyzer started
Now starting the IBM Support Assistant application
System resources and system load may affect the time required
to start the application. Please be patient...
IBM Support Assistant is ready to run.
Open a browser to:
"http://localhost:10911/isa5 OR http://(hostname):10911/isa5"



Launch the browser and use the bookmarks to load the ISA web interface.

🕗 IBM Support Assistant - Mozilla Firefox						_ 🗆 🗙		
Eile Edit View History Bookmarks Tools Help								
IBM Support Assistant +								
Contemporary Conte	♦ 🕲 localhost: 10911/isa5/index.html#							
IBM Support Assistant				6	ot Feedbac	ka Help IBM.		
Cases 🕨		•		Scan this Case	•	Global Filter - Off		
🖹 Files 🤷 Tools 📗 Reports 🗐	Overview 당 Symptom	s 🔞 Global Knowledg	e Base Matches	🕼 Data Collector				
Tree View				Search	i File Conter	٩		
	Name Filter 🚔		Filter	Reset				
Navigator	Name Syn	nptoms Knowledge Ba	First Timestam Last	Timestam Size	Туре	Modified (PST)		
d hd								

Note:

Like previous releases of ISA, the tool can be used to find problems, analyze data, and send information to IBM support. As it is a multi-user installation, ISA provides a case management component to help manage various problem determination activities that a team might require. A case is simply a container for a logical grouping of files and information. A typical practice would be to group artifacts pertaining to a single issue.

_ Create a new case by clicking the **Cases** button, and then **Add**.





	IBM Support Assistant	
Case Ma Add	Delete	Cases
Casi 🔺	Summary	bal Knowled
0000	Example Case	
0001	Testing	
		Knowledge B
Case II):	
Summa	ary:	
Descrip	otion:	

Complete the summary and description, and click the **green** tick.

Case ID:	[New]	✓ ×
Summary:	lab case 1	
Description	:	
test case	2	

_Shrink the cases dialog by clicking **Cases**.



Case Management Add Delete					
Casi 🔺	Summary	bal Knowle			
0000	Example Case				
0001	Testing				
0002	lab case 1				

_____ Having made a new case, diagnostic data can be added. This lab provides a sample WebSphere Application Server SystemOut log file that contains some error messages. Open the file on from the desktop, and note that it contains various WebSphere errors and Java stack traces.



🗄 SystemOut.log - WordPad
Eile Edit View Insert Format Help
<pre>*********** Start Display Current Environment *********** WebSphere Platform 7.0.0.23 [ND 7.0.0.23 cf231218.02] running with process name labNetwork\lab1prod\ Host Operating System is AIX, version 7.1 Java version = 1.6.0, Java Compiler = j9jit24, Java VM name = IBM J9 VM was.install.root = /usr/WebSphere/v7/AppServer user.install.root = /usr/WebSphere/v7/AppServer/profiles/AppSrvr Java Home = /usr/WebSphere/v7/AppServer/java/jre</pre>
<pre>ws.ext.dirs = /usr/WebSphere/v7/AppServer/java/lib:/usr/WebSphere/v7/AppServer/profiles/AppSrvr/clas Classpath = /usr/WebSphere/v7/AppServer/profiles/AppSrvr/properties:/usr/WebSphere/v7/AppServer/prop Java Library path = /usr/WebSphere/v7/AppServer/java/jre/lib/ppc:/usr/WebSphere/v7/AppServer/java/jr ************************************</pre>
<pre>[12/3/12 0:35:15:218 EST] 0000001e XARecoveryDat A WTRN01511: Preparing to call xa recover on XARe [12/3/12 0:35:15:232 EST] 0000001e J2CXAResource W J2CA0061W: Error creating XA Connection and Res at com.ibm.ejs.j2c.J2CXAResourceFactory\$1.run(J2CXAResourceFactory.java:253) at com.ibm.ws.security.util.AccessController.doPrivileged(AccessController.java:118) at com.ibm.ejs.j2c.J2CXAResourceFactory.getXAResource(J2CXAResourceFactory.java:190) at com.ibm.ws.Transaction.JTA.XARecoveryData.getXARminst(XARecoveryData.java:445) at com.ibm.ws.Transaction.JTA.XARecoveryData.recover(XARecoveryData.java:572) at com.ibm.tx.jta.PartnerLogTable.recover(PartnerLogTable.java:389) at com.ibm.tx.jta.RecoveryManager.resync(RecoveryManager.java:1530)</pre>
at com.ibm.tx.jta.RecoveryManager.performResync(RecoveryManager.java:2265) at com.ibm.ws.tx.jta.RecoveryManager.performResync(RecoveryManager.java:114) at com.ibm.tx.jta.RecoveryManager.run(RecoveryManager.java:2218) at java.lang.Thread.run(Thread.java:736)
[12/3/12 0:35:15:256 EST] 0000001e XARecoveryDat W WTRN0005W: The XAResource for a transaction par cfName = JMS\$null\$JMSManagedConnection@0 configProps = [Deployed Resource Adapter Properties]
TransactionResourceRegistration java.lang.String dynamic
For Help, press F1

Close the text editor.



_____ Add this sample SystemOut log to the ISA case by dragging it from the desktop to the ISA file list in the browser.

Add files	Or drag files into browser to add	Name Filter 뵭		<u>Filter</u>	Reset			
Navigator		Name	Symptoms	Knowledge Ba	First Timestam	Last Timestam	Size	Туре
+ 🗀 0002	Ş.	Copdtools					0 Bytes	directory
		(99939)						
		SystemOut.log						

_ Click the **yellow box** to dismiss the notification.

Successfully added the following (click this notification to hide):
 Collector tool output impact2013Cell.jar

_ Click the **refresh** icon in the Navigator to see the files in the case.

Navigator	
0001	Ŷ

__ Click the Status button. Notice the "Ticket" (case) has not been scanned.



х

IBM Software Accelerated Value Program



Click the Scan this Case button.

Problem Analysis

Run Automated Analysis

Input Files and Folders

/ISA5Beta2/ISA5/isa/cases/0002

Parameters

Parameter	Description	Value
force_rescan	This parameter controls the type of scan. Default is an incremental scan, scanning only new files. Check here to force a re-scan of ALL THE FILES within this case.	
ignore_failed	This parameter controls handling of previous failures. By default, a scan will not run if a previous scan has failed. Check here to force a scan regardless of the previously failed one. Note: Even with this setting, the scan will be terminated if it runs longer than 1 hour.	

Run as background task: 🗹

Submit Canc	el
-------------	----

_ Click **Submit** to start the Scan.



Note:

Scanning the case may take up to 5 minutes. In the meantime, let's investigate the tools that are installed into ISA 5.

_____ Click the **Tools** tab. Note that only the tools that will be used in this lab have already installed, more tools are available.

Note:

There are three key types of tools provided in ISA and each type has its benefits and compromises.

Report generator tools - process input data (e.g. log files or Java dumps) and generate a simple output file, usually in the form of an HTML or .txt report. These tools are not interactive but they are simple to run and have the benefit of consuming no local resources.

Web-based tools - run most of their analysis processing on the ISA server and provide a rich, interactive experience in the browser based user interface. These types of tools are ideal for activities where you want to off-load heavy processing of files to a more powerful server.

Desktop tools - typical desktop client-side applications that are launched via the ISA browser UI. By leveraging Java WebStart, the entire tool will be installed and run locally on your desktop. This type of tool has a few drawbacks - a Java plugin is required for your browser, local system resources are required to run the client tool, and you must have local access to the files you wish to analyze. However, some ISA tools are only available as desktop tools.

🗈 Files 🤷 Tools 🗽 Reports 🕻) Overview	🟭 Symptom	is 🔞	Global Knowledge Base Matches	🔡 Data Collector
Enter keyword	<u>et</u>				
Sort By: 📴 Tag: All Tags	•				
Garbage Collection and Memory Visualizer (GCMV) [Desktop]	V			Select a tool from t	ne list to display (
Garbage Collection and Memory Visualizer (GCMV) [Report]	√ <u>∎</u>		bead	mara shaut additional to	ole ausilable to in
THealth Center	V	<u>1</u>	<u>(eau</u>		
👕 HeapAnalyzer [Desktop]	6°) 🗊				
👕 Memory Analyzer [Desktop]	V 🗾				
T Memory Analyzer [Report]	🗸 🛄				
T Memory Analyzer Web Edition [Web]	69 💿				
Thread and Monitor Dump Analyzer (TMDA) [Desktop]	69 🗊				
Thread and Monitor Dump Analyzer (TMDA) [Report]	°§ 🛄				
WebSphere Application Server	69 🛄				

Impact2013

Business. In Motion.

Note:

By now the scanning of the case should be finished, let's check the results.

Click the **Overview** tab. It shows useful system information determined from the log file(s) in the case. (If the scan is not completed nothing will display and you will have to repeat this step until data is ready).

3M Software Accelerated Value Program Business. In Motion.			
Files 🧰 Tools 녪 Report:	s 📋 Overview 👪 Symptoms 懰 Global Knowledge Base Matches	🔡 Data Collector	
Collapse All	Scan Coverage	🕒 Printable ver	
✓ General Information From a Sci	an of This Ticket		
Number of files included in th Number of symptoms discover First Timestamp discovered:: Last Timestamp discovered:: • Product Versions	he most recent scan: 6 ered in the most recent scan: 3 : 12/03/12 00:35:15:218 12/03/12 10:09:30:545		
WebSphere Platform 7.0.0.23 [NI	D 7.0.0.23 df231218.02]		
✓ JVM Versions			
1.6.0, Java Compiler = j9jit24, Ja	ava VM name = IBM J9 VM		
▼ WebSphere Versions			
WebSphere Platform 7.0.0.23 [NI			
· · · · ·	D 7.0.0.23 cf231218.02]		
✓ OS Versions	D 7.0.0.23 cf231218.02]		
 ✓ OS Versions AIX, version 7.1 	D 7.0.0.23 cf231218.02]		
OS Versions <u>AIX, version 7.1</u> · iFix Information	D 7.0.0.23 cf231218.02]		
OS Versions <u>AIX, version 7.1</u> iFix Information <u>{no iFix information found}</u>	D 7.0.0.23 cf231218.02]		

Impact 2012

Click the **Symptoms** tab. It shows a list of the errors encountered in the log file(s).

	🗈 Files 🤷 Tools 📗 Reports 🗊 Overview 👪 Symptoms 🗌 Global Knowledge Base Matches 港 Data Collector									
s	Symptoms Filter Decot									
(Global Sco	Туре	Symptom				Symptom Occurrences	Knowledge Base Ma	First Occurren	Last Occuri
	al.	ErrorMsg	WTRN0005W: The XA recovery may not be a	Resource for a tran able to complete pro	nsaction participant perly. The resource	could not be recreated and transaction was J2CXAResourceInfo :	63	10	12/03/12 00:35:15:256	12/03/12 09:07:18:039
	al -	AdHoc	CWZZZ0001W: Possil	ble abnormal startu;	p - did not find 'open) for e-business'	1	4	12/03/12 10:09:30:545	12/03/12 10:09:30:545
	al.	ErrorMsg	J2CA0061W: Error cre lacks an RA wrapper a	eating XA Connectio and an RA wrappe	on and Resource jav r could not be resolv	va.lang.Exception: Parameter xaResInfo ved using RA key.	63	10	12/03/12 00:35:15:232	12/03/12 09:07:18:021

_____ Click the **Global Knowledge Base** tab. This compares the symptoms to a local database (XML file) to suggest possible resolutions including APARs (IBM fix references) and Technotes. Click on a suggestion to see a detailed description below.



Business. In Motion.

🕘 0002 - IBM Support Assistant 5.0 - Mozilla Firefox				
Elle Edit View Higtory Bookmarks Tools Help				
(a) WebSphere Integrated Solutions Console × (a) 0002 - IBM Support Assistant 5.0 × +				
(♦) @ localhost:10911/isa5/index.html#id=0002	<i>P</i> ^			
IBM Support Assistant	erdbeck? Help IBM.			
Cases 🕨 [0002] lab case 2 💌 💌	Global Filter - Off 💌			
🗈 Files 🤷 Tools 📗 Reports 🗉 Overview 🏭 Symptoms 🔮 Global Knowledge Base Matches 🔠 Data Collector				
Symptoms Filter 🔆 Enter keyword Filter Reset Scan Coverage	g 17 of 22 results 📃 Show All			
Global Sci Type Knowledge Base Entry Symptom	Tool ID			
APAR PM15719: THE TRANSACTION MANAGER FAILS TO GET AN XARESOURCE TO ROLLBACK THE TRANSACTION. Multiple symptoms (2) matched by this entry	LocalKBSer9			
PK91826: CSCP0007E, WTRN0005W THE XARESOURCE WTRN0005W THE XARESOURCE WTRN0005W THE XAResource for a transaction participant could not be recreated and transaction recovery not be able to complete properly. The resource was J2CXAResourceInfo :	y may LocalKBSer16			
II APAR PK81814: MESSAGE "OPEN FOR E-BUSINESS" IS NOT BEING PRINTED IN SYSTEMOUT.LOG CWZZZ0001W: Possible abnormal startup - did not find 'open for e-business'	LocalKBSe:1			
PK83560: WLM RETURNING TARGET WITH EMPTY II APAR ENDPOINTS CAUSES SIB EXCEPTIONS RESULTING IN CWST0019E OR CWSIA0241E ERRORS CWST0019E OR CWSIA0241E ERRORS	LocalKBSet3			
Knowledge Base Matches Symptom Occurrences Symptom Details Containing Files				
Type: APAR Found by Tool: LocalKBSearch Global Score: 1846 Label: PK83560: WLM RETURNING TARGET WITH EMPTY ENDPOINTS CAUSES SIB EXCEPTIONS RESULTING IN CWSIT0019E OR CWSIA0241E ERRORS Match ID: 3 Symptom IDs associated with this Match: 85 Description:				
Click on the link for more references: http://www.ibm.com/Search/?g=PK83560				
Abstract: WLM RETURNING TARGET WITH EMPTY ENDPOINTS CAUSES SIB EXCEPTIONS RESULTING IN CWSIT0019E OR CWSIA0241E ERRORS.	-			
Build ID: 5.0.0.0_Beta2_20121016-1409 © Copyright IBM Corp. 20	011, 2012. All rights reserved			

_____ To demonstrate one of the report generator tools, first switch to the "PlantsByWebSphere" case that has been prepared for this lab by selecting it from the dropdown.

n Support Assistant		
0003] PlantsByWebSphere		
0000] Example Case		
000211ab case 2 0003] PlantsByWebSphere		K
	0003] PlantsByWebSphere 0000] Example Case 00021 lab case 2 0003] PlantsByWebSphere	0003] PlantsByWebSphere 0000] Example Case 0002] lab case 2 0003] PlantsByWebSphere



Business. In Motion.

IBM Support Assistar	t						iot I
Cases [0003] PlantsBy WebSphere	▼				s	can this Case	,
Files 💼 Tools 📗 Reports 🕻	🕽 Overview 📲 Symptoms 🗌 Global H	(nowledge Base M	atches 📲	Data Collect	or		
E Tree View CASE:0003/*						Searci	h Fili
Add files Or drag files into browser to add	Name Filter		<u>Filter</u> B	<u>leset</u>			
Navigator	Name	Symptoms Kr	nowledge Ba	First Timestam	Last Timestam	Size	Typ
0003	Collector tool output impact2013Cell.jar					6 MB	jar

Note:

The case contains file "Collector tool output impact2013Cell.jar". This is output from the WebSphere collector tool which collects information about the WebSphere Application Server configuration – it is frequently requested by IBM support when investigating PMRs.

To demonstrate one of the report generator tools, return to the Tools tab and select **WebSphere Application Server Visualizer**. Then press the launch button.



🕙 0003 - IBM Support Assistant 5.0 - Mozilla Fire	efox			_ 🗆 ×
<u>Eile E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u>	lelp			
🕀 WebSphere Integrated Solutions Console 🛛 🗌	0003 - IBM Support Assistant 5.0	× +		
Contemporary Conte	8id=00038toolId=com.ibm.wsvisu	alizer	☆ マ C S - Google	<i>P</i> ^
IBM Support Assistant			Cot Feedback ? He	^{olp} IBM.
Cases [0003] PlantsByWebSphere		v	Scan this Case 🛛 🔻 Global Fil	ter - Off
Files 🗖 Tools 📴 Ov	erview 👪 Symptoms 🦉	Global Knowledge Base Matches 📳 Data Collector		
			Search Tool Help	Q
Enter keyword Filter Reset	WebS	Sphere Application Server Configurati	on Visualizer Version 1.0.0	
Sort By: 📮 Tag: All Tags	💌 🚺 Launa	ch 🕕 Tool Help		
Garbage Collection and Memory Visualizer (GCMV) [Desktop]	WebSph	ere Application Server Configuration Visualizer		
Visualizer (GCMV) [Report]	Generat	tes an interactive HTML visualization of a WebSphere		
Fleath Center	Applicat	ion Server configuration, including Service Integration		
heapAnaryzer [Jesktop]	containi	ng configuration directories. Configurations from		
Memory Analyzer [Beport]	multiple visualiza	nodes in a cell will be merged into a single ation, and multiple cells can be displayed in the		
Memory Analyzer Web Edition [Web]	output.			
Thread and Monitor Dump Analyzer			No Screenshot Available	
Thread and Monitor Dump Analyzer	3 <u>II.</u>			
WebSphere Application Server Configuration Visualizer	Tags:	🗽 Report Generator Tool 🥙 As-is Problem Area: We	bSphere Problem Area: Configuration	
	r:1- T			-
Build ID: 5.0.0.0_Beta2_20121016-1409			© Copyright IBM Corp. 2011, 2012. All righ	ts reserved .::
			OK Cancel	

Navigate to the file **Collector tool output impact2013Cell.jar** select it and press OK.



Submit

Cancel

Problem Analysis

х

Run WebSphere Application Server Configuration Visualizer (Version 1.0.0) Input Files and Folders *

/ISA5Beta2/ISA5/isa/cases/0003/Collector tool output impact2013Cell.jar Bro	owse
---	------

Parameters

Parameter	Description	Value
ReportName	Report Name	topology.html

Run as background task: 🔽

WebSphere Application Server

Configuration Visualizer Collector tool output impact2013Cell.jar

Ø

Click Submit

Click on the **Reports** tab, then click the **Refresh** icon a few times until the report is complete, as indicated by a green tick icon.

🗈 Files 🧰 Tools 📗 Reports 📋 Overview	v 🏭 Symp
Enter keyword <u>Filter</u> Reset	
Sort By:	S.
WebSphere Application Server 1/3/13 C	06:41:56
Collector tool output impact2013Cell.jar	
👔 Files 🧰 Tools 📗 Reports 📋 Overview	v 🏭 Sym
Enter keyword <u>Filter</u> Reset	
Sort By: 12 10	<u>%</u>

http://www-01.ibm.com/software/support/acceleratedvalue/

1/3/13 06:42:20



_ Select the completed report.

🗈 Files 🤷 Tools 🗽 Reports 🗐 👀	verview 👪 Sym	
Enter keyword <u>Filter</u> Reset		
Sort By: ↓ªZ ↓Ŏ	Ŷ	
WebSphere Application Server 1/3/13 06:42:20		
Collector tool output impact2013Cell.jar		

Note:

Take a look at the report in the right hand window. This visualizes the topology of WebSphere Application Server. For example, in this lab the topology consists of a single cell containing two nodes – one containing the deployment manager, and the second containing two application servers. Those servers are part of a cluster called "PlantsByWebSphereCluster". In addition, there are several datasources defined in the configuration.

_____ Try clicking on any of the topology components, the report is interactive and will pop up dialogues with more detailed information.







Part 3: Advanced Javacore Analysis with IBM Thread and Monitor Dump Analyzer Tool

Note:

Java dumps (ala javacore files or thread dumps) are useful for analyzing hung thread scenarios. In this section, you will use a javacore file help debug a "Possible Hung Thread" warning in the WebSphere log file. The following ISA 5 tools will be used:

IBM Thread and Monitor Dump Analyzer - used to interpret Javacore files

Memory Analyzer - used to inspect the Java objects associated with the hung thread

IBM Extensions to Memory Analyzer– provides addition functions to correlate thread Ids in Javacore files with other WebSphere traces

Launch the browser and use the bookmarks to load the WebSphere Integrated Solutions Console (admin console) in a new tab.

Login using a blank user name.

_ Navigate to Servers->Server Type->WebSphere Application Servers. Select server1.



Press the Start button and wait for the admin console to report the server has started.

Messages
Server impact2013Node/server1 started successfully. The collection may need to be refreshed to show the current server status. <u>View JVM logs</u> for further details.



Note:

Next you will use the Jmeter load generating tool to simulate some user requests to the Plants web application. Some of these user requests will trigger a deliberate error which you will diagnose using ISA 5 tools.

Double click the **Jmeter** shortcut on the desktop.



Click File->Open and navigate to E:\Impact lab files\Jmeter scripts\Impact 2013-2.jmx. Click Open.

Eile Edit Search Dun Ontione Hole				
	· 🔸 🕨 🔍 📽 🗞 😪 🖼 🌌 🏘 🏷 📰 🛛			
Test Plan WorkBench	Test Plan Name: Test Plan			
	Comments:			
	User Defined Variables			
	Name:			
	🖙 Open 🔀			
	Look In: 🗇 Jmeter scripts 🔽 🖬 🛱 🛱 📴 📴			
	Impact 2013-1.jmx Impact 2013-2.jmx File Name: Impact 2013-2.jmx Files of Type: JMeter [.jmx]			
	Open Cancel Bun Thread Groups consecutively (i.e. run groups one at a time)			
	- Train Thiroda or only consecutively (ner run groups one at a time)			



Impact

Business. In Motior



__ From the desktop, double click the **Tail** icon.



_____ Close any open files in Tail.

_____ Use Tail to monitor the SystemOut.log of the **server1**. The file is located at:

C:\IBM\WebSphere\AppServer\profiles\AppSrv01\logs\server1

Monitor the file for a hung thread warning message similar to that shown below. It may take up to 3 minutes to appear (you may read the "Note:" section below while you wait). Make a note of the thread title, e.g. "WebContainer : n" – the actual value of "n" may very with the screenshots in this lab document.

Note: Be sure to watch the tail carefully as the error will scroll off the screen.

🐚 SystemO	ut.log				
[2/11/13	6:09:55:578 PST]	000000a7	SystemOut	0	==> STARTING SLOW METHOD
[2/11/13	6:09:55:578 PST]	000000a7	SystemOut	0	timestamp=1360591795578
[2/11/13	6:09:55:578 PST]	000000a7	SystemOut	0	resume at=1360591805578
[2/11/13	6:09:58:015 PST]	00000097	SystemOut	0	==> ENDING SLOW METHOD
[2/11/13	6:09:58:125 PST]	000000aa	HtmlGridRende	W	PanelGrid shopping:j_id164039580_461abe83 has not enough children. Child count
[2/11/13	6:09:58:156 PST]	00000097	SystemOut	0	performProductDetail : itemID=F0018
[2/11/13	6:09:58:156 PST]	00000097	SystemOut	0	==> STARTING DELIBERATE LARGE SESSION
[2/11/13	6:09:58:156 PST]	00000097	SystemOut	0	==> ENDING DELIBERATE LARGE SESSION
[2/11/13	6:10:05:578 PST]	000000a7	SystemOut	0	==> ENDING SLOW METHOD
[2/11/13	6:10:07:218 PST]	00000073	ThreadMonitor	W	WSVR0605W: Thread "WebContainer : 0" (00000096) has been active for 147406 mil
at	java.lang.Thread	.sleep(Nat	tive Method)		
at	java.lang.Thread	.sleep(Th	cead.java:896)		
at	com.ibm.webspher	e.samples	.pbw.war.Shopp:	ing	gBean.deliberateHungThread(ShoppingBean.java:415)
at	com.ibm.webspher	e.samples	.pbw.war.Shopp:	ing	gBean.performProductDetail(ShoppingBean.java:182)
at	sun.reflect.Gene	ratedMeth	dAccessor166.:	inv	/oke(Unknown Source)
at	sun.reflect.Dele	gatingMetl	hodAccessorImp.	l.i	invoke (DelegatingMethodAccessorImpl.java:37)
at	java.lang.reflec	t.Method.:	invoke(Method.;	jav	7a:611)
at	org.apache.webbe	ans.inter	cept.Intercept	orF	Handler.invoke(InterceptorHandler.java:287)
at	org.apache.webbe	ans.inter	cept.NormalScop	ped	dBeanInterceptorHandler.invoke(NormalScopedBeanInterceptorHandler.java:98)
at	com.ibm.webspher	e.samples	pbw.war.Shopp:	inc	γBean \$\$ javassist 1.performProductDetail(ShoppingBean \$\$ javassist 1.java)



Note:

WebSphere Application Server has a built in heuristic for detecting threads that could be potentially hung. To speed up this lab, some custom properties have been set to influence the behavior of the hung thread detection with the goal of making it trigger more quickly than normal. In addition, automatic generation of Javadumps on hung thread detection has been configured. The properties configured for the lab are summarized in the table below. No action is required.

Custom WebSphere Property	Description	Default	Lab Setting
com.ibm.websphere.threadmonitor.interval	The frequency, in seconds, at which managed threads in the selected application server will be interrogated	180 seconds	60 seconds
com.ibm.websphere.threadmonitor.threshold	The length of time, in seconds, in which a thread can be active before it is considered hung. Any thread that is detected as active for longer than this length of time is reported as hung	600 seconds	120 seconds
com.ibm.websphere.threadmonitor.false.alarm.threshold	The number of times that false alarms can occur before automatically increasing the threshold	100	0 (disabled)
com.ibm.websphere.threadmonitor.dump.java	Set to true to cause a javacore to be created when a hung thread is detected and a WSVR0605W message is printed	false	True



After the hung thread warning has been made in the SystemOut log, switch to directory "C:\IBM\WebSphere\AppServer\profiles\AppSrv01". Verify that a Javacore.txt file has been automatically written by WebSphere Application Server in response to the hung thread notification.

Stop the Jmeter script by clicking **Run->Stop**. Close the Jmeter window.

Note:

When analyzing hung threads or other performance problems, it is often useful to have several Javadumps over a period of a few minutes. This can show that while a thread has been active for a long time, its current stack frame is changing meaning it is still busy processing its work, and actually not hung at all.

In the next steps you will request the JVM to write a 2nd Javadump, and a system dump. There are several ways to do this including:

- 1. wsadmin scripting interface
- 2. Health Center tool
- 3. WebSphere Administration Console (for WebSphere Application Server 8.0 and above)

In this lab, you will use the WebSphere Administration Console.

_____ Launch the browser and use the bookmarks to load the WebSphere Integrated Solutions Console (admin console).

Navigate to Troubleshooting->Java dumps and cores.

ava dum	ps and cores					?	
Java dumps and cores							
Use th are pla	is panel to generate heap aced on the local file syster	dumps, Java cores or system o m.	lumps for a running process	. The files resulting from	these op	erations	
🗄 Pret	ferences						
Hea	ap dump 🛛 Java core 🛛 🤅	System dump					
D	D # \$						
Select	Server 🛟	Node 🗘	Host Name 🗘	Version 🗘	Туре 🗘	Status ሷ	
You c	an administer the following) resources:					
	dmgr	impact2013CellManager	impact2013	ND 8.5.0.1	servers	€	
	nodeagent	impact2013Node	impact2013	ND 8.5.0.1	servers	€	
	server1	impact2013Node	impact2013	ND 8.5.0.1	servers	€	
	server2	impact2013Node	impact2013	ND 8.5.0.1	servers	*	
Total	4						

Select server1 and press Java core. Wait a few minutes for the response.



🖃 Messages

Java core request was sent successfully to server server1.

_____ Select **server1** and press System dump. Wait a few minutes for the response. A system dump takes longer than the Java dump in the previous step.

🖃 Messages

🗓 System dump request was sent successfully to server server1.

The output file for the operation is C:\IBM\WebSphere\AppServer\profiles\AppSrv01
\.\core.20130207.032006.7740.0002.dmp.

Note:

In previous versions of the IBM SDK shipped with WebSphere Application Server, it was necessary to post-process a system core file using a command called jextract before the system core could be used by tools such as Memory Analyzer. However, newer SDKs do not require this jextract step.

For reference, the IBM SDKs 1.6.0 SR9 or later, or any version of Java 1.7, no longer require jextract to be run. A technote in the references sections correlates the WebSphere Application Server versions to the SDK shipped.

This shows the SDKs which do not require jextract were shipped with WebSphere Application Server 7.0.0.15 and above.



_____ Navigate to **Servers->Server Type->WebSphere application servers**. Select the running server. and press the Stop button.

opplication servers				?
Application servers				
Use this page to view a list of the ap servers. You can also use this page	plication servers to change the st	in your envi atus of a spe	ronment and the status of ea cific application server.	ch of these
🛨 Preferences				
New Delete Templates	Start Sto	p Restart	: ImmediateStop Term	inate
Select Name 🛟 Node 🗘	Host Name 🗘	Version 🗘	Cluster Name 🗘	Status ሷ
You can administer the following re-	sources:			
server1 impact2013Node	impact2013	ND 8.5.0.1	PlantsByWebSphereCluster	€
server2 impact2013Node	impact2013	ND 8.5.0.1	PlantsByWebSphereCluster	8
Total 2			·	

Server stat Server : Server s	us feedback status feedback status provides information about events that occur while the server stops.	<u>, 1</u>
•	impact2013Node:server1	
OK		



_____ Launch the browser and use the bookmarks to load the ISA web interface.

🕹 IBM Support Assistant - Mozilla Firefox									
Eile Edit View History Bookmarks Tools	Help								
IBM Support Assistant	BM Support Assistant +								
♦ 🕑 localhost: 10911/isa5/index.html#									
IBM Support Assistant		6at Feedback? Help IBM.							
Cases ▶ Scan this Case ♥ ▼ Global Filter - Off ▼									
🖹 Files 🤷 Tools 📗 Reports 🗐	Overview 📲 Symptoms 📔 Global Knowledge Base Match	nes 📳 Data Collector							
Search File Conter									
Name Filter									
Navigator	Name Symptoms Knowledge Ba First Timestam	Last Timestam Size Type Modified (PST)							

Create a new case by clicking the Cases button, and then Add.

2 I	BM Sup
Cases 🕨	

	IBM Support Assistant	
Case Mar Add	nagement Delete	Cases
Casi 📤	Summary)bal Knowle
0000	Example Case	
0001	Testing	Knowledge B
Case ID Summa Descrip	: ry: tion:	



_ Complete the summary and description, and click the green tick.

Case ID:	[New]	~	×
Summary:	PlantsByWebSphere Hung Thread		
Description	:		
Hung thre	ead warning seen.		

Shrink the cases dialog by clicking **Cases**.

Case Management Add Delete		
Casi 🔺	Summary	bal Knowled
0000	Example Case	
0001	PlantsByWebSphere	
0002	PlantsByWebSphere Hung Thread	

Add the two javacore.txt files, and one system dump file to the case by dragging the files from Windows Explorer into the ISA files list. The location of the files is:

C:\IBM\WebSphere\AppServer\profiles\AppSrv01



🖹 Files 🙆 Tools 📗 Reports 🗐 O	verview 将 Symptoms 🗌 Glo	bal Knowledge Base №	1atches 📳 Data Collecto	or		
E Tree View CASE:0002/*						Sea
Add files Or drag files into browser to add	Name Filter 🚔		Filter Reset			
Navigator	Name			Symptoms	Knowledge Ba First Timestam	Last Timestam _l Size
		Ele Edit View Calance Edit View Calance Edit View Calance Edit View Calance Edit View Folders B B B B B B B B B B B B B	ere\AppServer\profiles\Ap Favortes Iools Help P Search P Folders WebSphere\AppServer\profiles blafiles I lafiles I l	AppSrv01	P :::: • re.20130207.053039.7740.0003.0130207.032006.7740.0002.dmp re.20130207.031900.7740.0001.07022013.0315.19.txt p ace uration s	↓ ↓ 60 xt 2, · 353, xt 2, ·
Cases 🕨 [0002] Plan	tsByWebSphere⊢					
Files 💼 Tools	<u>llı</u> Reports 📋					
🗄 Tree View 🔻 🗲	CASE:0002/*					
14%						
Navigator						
0002	Ŷ					

Wait for the file add progress bar to read 100%, and for the yellow message box to appear.



Successfully added the following (click this notification to hide):

- core.20130207.032006.7740.0002.dmp
- javacore.20130207.031900.7740.0001.txt
- javacore.20130207.053039.7740.0003.txt

Cick the yellow box to dismiss the message

Note:

The files are now safely stored in the ISA folder where they can be accessed by a team of people.

The information in a Javadump is human readable, but the TDMA tool helps interpret the data. The primary benefits this tool provides are:

1. Summary of the Javadump, including any warnings, thread histogram by type and state, memory segment information, etc.

2. User interface for navigating thread stacks and monitors.

3. User interface for comparing multiple Javadumps and their respective thread stacks and monitors.

To make an initial assessment of the files, it can be convenient to run a TMDA report on the ISA server. This avoids the need to download the files & TMDA tool to a local workstation. More detailed analysis can be performed later using the desktop version of TMDA if required.

In the next steps you will generate a basic TMDA report to see high level details contained in the Javadump file.

___ Ensure the relevant case is selected, then choose Thread and Monitor Dump Analyzer (TMDA) Report from the tools tab. Click **Launch**.



Impact2013

Business. In Motion

Problem Analysis	х
Run Thread and Monitor Dump Analyzer (TMDA) [Report] (Version 4.3.4)	
Input Files and Folders *	
Brows	se
Run as background task: 🗹	
Submit Can	cel



Click Browse and select one of the Javacore,txt files.

Select a File	х
0002	
eore.20130207.032006.7740.0002.dmp	
📄 📄 javacore.20130207.031900.7740.0001.txt	
Javacore.20130207.053039.7740.0003.txt	
OKCan	el :
Click OK	



Problem Analysis	х
Run Thread and Monitor Dump Analyzer [Report] (Version 4.3.4) Input Files and Folders *	(TMDA)
/ISA5Beta2/ISA5/isa/cases/0002/javacore.20130207.053039.7740.0003.txt	Browse
Run as background task: 🔽	
	Submit Cancel
Click Submit .	
BM Support Assistant The Thread and Monitor Dump Analyzer (TMDA) [Report] tool request has been submitted. Go to output folder.	lin
ses) [0002] PlantsByWebSphere Hung Thread 💌	Scan this Case

____ Click the **yellow box** (not the link) to dismiss it, then click the **reports** tab.

Enter keyword	<u>Reset</u>
Sort By: ↓ª ↓Ŏ	\$
Thread and Monitor Dump Analyz (TMDA) [Report]	er 2/7/13 06:29:30
javacore.20130207.031900.7740.0001.tx	t

____ Click the **refresh** icon until the report's status shows a green tick. Then select the **report**.

Enter keyword <u>Filter</u> Reset	Thread and Monitor Dump Analyzer (TMDA) [Report]
Sort By: Jaz Jo	
Thread and Monitor Dump Analyzer 2/7/13 06:29:34 (TMDA) [Report]	
javacore.20130207.031900.7740.0001.txt	Java Thread and Monitor Dump Analysis

_____ Take a brief look at the report on the right hand side. You should be able to find the following general information:



Cause of thread dump – This helps determine if the JVM experienced a crash or out of memory. In this case, the dump was triggered by a user signal, i.e. WebSphere's hung thread detection and/or the Health Center tool.

Java Version – The exact java version is displayed making finding IBM fixes easier.

System Classpath & User Arguments – Gives information about the Java environment and its configuration.

Free & Allocated Java Heap Size – Learn if the heap is fully expanded, and what percentage occupied it is.

Thread Status Analysis – Shows the status and number of running threads. In TDMA, the different states are color coded. The meaning of each state is explained later in this lab.

Thread Method Analysis – Shows what code is being executed by each thread at the time of the Javacore

Note:

To diagnose the hung thread problem presented in this lab, the more advanced and interactive features of the desktop TDMA tool are required.

_ Choose Thread and Monitor Dump Analyzer (TMDA) Desktop from the tools tab. Click Launch.



Impact2013

Business. In Motion



Submit

Cancel

Problem Analysis

х

Run Thread and Monitor Dump Analyzer (TMDA) [Desktop] (Version 4.3.4)

This tool is a desktop application. It will be launched using Java Web Start and will run on your workstation. Using the tool with files associated with ticket 0002 will require that you have access to the files from the workstation. If a file is located on a remote server, you can download the file to a local file system location or access the file through a shared storage area. Any existing local file may also be accessed by the tool.

In some cases, analysis of files on your workstation can noticeably degrade performance of other applications running on your workstation.

Click 'Submit' below to begin.

Click **Submit**. The TMDA desktop tool will be downloaded and launched using Java Web Start – this will take a few moments. In the meantime, read the note below in preparation for using the TMDA desktop tool.





🧾 IBM Thread and Monitor Dump Analyz	er for Java	
<u>F</u> ile <u>A</u> nalysis ⊻iew <u>H</u> elp		
🗅 X 🗶 🔺 🎍 🍓 🦧	🧐 🚈 🛄 💷 🕢 🏈 🗸 Floatable	
	Throad Dump List	
Iname Inmestamp	Runnable/Total T Free/Allocated He AF(SC)/GC Counter	Monitor Contention
	Console	
Open Open Javacore or Thread Dump		

Note:

The SystemOut log file reported a "possible hung thread" warning, i.e. a thread that has been active for a period of time longer than considered normal by the hung thread detection heuristic. The log file also showed the name of the thread (something like "WebContainer : 5") and a stack trace of the currently executing code.

You can learn the naming convention of common WebSphere and JVM threads by referring to a technote in the the references section. Threads starting with "WebContainer" are WebSphere threads dealing with incoming HTTP requests that come from the client.



The Java stacktrace showed that a ShoppingBean object in the PlantsByWebSphere application had executed a Thread.sleep, meaning the thread did not end in a timely manner. However, it is likely that the WebSphere Application Server will have many "WebContainer" threads that are handling incoming HTTP requests, and many would be executing the ShoppingBean code. The remaining steps of this lab will investigate why that particular thread encountered a (deliberate) problem.

____ In the TMDA tool. Select File->Open thread dumps

Navigate to C:\ISA5Beta2\ISA5\isa\cases\0002

遵 Open		×
Look <u>I</u> n: 👔	0002	
 imetadat javacore. .toolscor core.201 javacore. javacore. 	a 20130207.031900.7740.0001.bxt-jca_ISA_PD text 30207.032006.7740.0002.dmp 20130207.031900.7740.0001.bxt 20130207.053039.7740.0003.bxt	
File <u>N</u> ame:	C:\ISA5Beta2\ISA5\isa\cases\0002	
Files of <u>T</u> ype:	All Files	•
		Open Cancel

_____ Select the first **Javacore.txt** file, and click **Open** (look at the last few characters of the filename to determine which was created first).



_____ Select the **Javacore.txt** file in the Thread Dump List. Below you will see the general overview information that you saw in the previous TMDA report.

____ Right click the **file** and choose **Thread Detail** from the popup menu.

		Thread Du	mp List
Name	Timestamp	Runnable/Total T F	Free/Allo
javacore.201302	Feb 7 03:19:00 (🤨 <u>T</u> hread Detail	5,1
		Monitor Detail	
		🥏 <u>C</u> ompare Threads	
A ¥		💿 C <u>o</u> mpare Monitors	
 File name 	: C:)ISA5Beta2)IS/	X Close Thread Dumps	30.
- The hame	. 0.10/10/00/02/10/	🏾 💥 Close <u>A</u> ll Thread Dum	ps

🔄 IBM Thread and	d Monitor Dump /	Analyzer for Java										
<u>File Analysis Vi</u>	iew <u>H</u> elp											
🗅 X X 🗡	< 👌 🌞 🧔	s 🤌 🕲		•) 🗸 🧐 🗹 Float	table						
0					Thread D	Detail : javao	ore.20130207.031900.7	740.0001.txt				
Name 🔺	State	NativeID	vlethod		Waiting Threads : 0							1
AIO Timer Thre	🍸 Waiting on c	. 0x1f88 j	ava/lang/Thr		· · · · · · · · · · · · · · · · · · ·	. •	Thread Status Analysis					
Approximate Ti	X Waiting on c	Ox1eb4 j	ava/lang/Thr	J,	,	•			-			
AsyncPmiBBLIs	.X vvaiting on c	UXITEC J	ava/lang/Obj				Status	Number of Threads : 108	Percentage			
Attach API wait I	Runnable	Oxfile) OxfieaD r	avariang/Obj				Deadlock	0	0 (%)			
BBSon:0	😵 Waiting on c	. Ox1fd0 j	ava/lang/Obj				đ	1.0				
BBSon : 1	🟅° Waiting on c	. Ox1fcc j	ava/lang/Obj				P Runnable	18	17 (%)			
Bundle File Clo	🏅 Waiting on c	Ox1e9c j	ava/lang/Obj				X ⁰ Maiting on condition	68	63 (%)			T
Completion Pro	🎢 Runnable	0x1010 c	om/ibm/io/a				La vivalang on conduction					
Concurrent Mar	X Waiting on c	Ux1e58 N	VU JAVA STA				³ Waiting on monitor	0	0 (%)			
Default : 0	B Runnable	0x1130 5	sunninocrivvi				U. Quanandad	0	0.(%)			
Default : 1	Vaiting on c	Ox10dc i	ava/lang/Obi				Suspended	0	0 (70)			
Default : 2	Rarked	0x1494 s	sun/misc/Un				Cbject.wait()	0	0 (%)			
Default : 3	🔚 Parked	0x13b0 s	sun/misc/Un					0	0.001			
Default : 4	🏅 Waiting on c	. 0x13a0 c	:om/ibm/io/a				Blocked	U	0 (%)			
Default : 5	🖉 Runnable	0x1388 c	om/ibm/jvm/				X ^B Parked	22	20 (%)			
Deferrable Alar	≚≅ Parked	Ox1efD s	sun/misc/Un									
Deferrable Alar	All Parked	UXTUC4 s	sun/misc/Un									
Deferrable Alar	A Parked	0x000 s	sunimisciOn			•	Thread Method Analysis					
Deferred Alarm	Vaiting on c	Ox1ed0 i	ava/land/Ohi					Mathead Name			Number of	Descentario
Finalizer thread	Vaiting on c.	Ox1ea8 M	NO JAVA STA					method Nam			Threads : 108	Percentage
Framework Eve	To Waiting on c	. Ox1ebO j	ava/lang/Obj				iovo/long(Object.woit/N/	ativo Mothod)			46	42 (%)
GC Slave	🏅 Waiting on c	. 0x1e64 M	NO JAVA STA				Java/lang/Object.wait(iva	auve metriou)			40	43 (20)
HAManager.thr	🏹 Waiting on c	. 0x157c s	sun/misc/Un		Blocked by : U		sun/misc/Unsafe.park()	Native Method)			27	25 (%)
HAManager.thr	Waiting on c	0x410 s	sun/misc/Un				iava/lang/Thread sleen/	Native Method)			9	8 (%)
Health Center tr	Vvaiting on c	. Ux1e/U P	NU JAVA STA				jaranangi medalereep,	ruare meaneay				
HungThreadDe	Valuing on c	Dv1ec6 j	ava/lang/Thr				com/ibm/io/async/Async	:Library.aio_getioev2(Native	Method)		6	6 (%)
IProfiler	Waiting on c	0x1e50 h	IO JAVA STA				NO JAVA STACK				6	6 (%)
JIT Compilation	Vaiting on c	0x1a48 M	NO JAVA STA								-	
LT=0:P=71965	🔊 Runnable	0x1e84 j	ava/net/Plain				Javainet/PlainSocketimp	oi.socketAccept(Native Meth	5a)		5	5 (%)
LT=1:P=71965	🔊 Runnable	0x1f10 j	ava/net/Plain				sun/nio/ch/WindowsSel	ectorimpl\$SubSelector.poll	D(Native Metho	d)	3	3 (%)
LT=2:P=71965	🖉 Runnable	Ox1ffD j	ava/net/Plain				com/ihm/misc/SignalDi	enstcher waitForSignal/Not	ve Method)		1	1 (%)
LocalNotificatio	Vvaiting on c	UX318 s	sun/misc/Un				Common and sublightable	opaterier.waitr Urolynal(Nat	ve meulou)		<u></u>	1 (10)
MechoryFuulMA	Waiting on c	Oviencia Oviencia	,onn/lum/lang availangi∩hi				com/ibm/lang/manager	nentMemoryNotificationThr	ead.processNo	otificationLoop(Native	1	1 (%)
Mtl Tx EventAn	Vaiting on c	0x1f64 i	ava/lang/Obj				Method)				<u> </u>	
Non-Deferred A	Vaiting on c	Ox1ecc j	ava/lang/Obj				com/ibm/jvm/Dump.JavaDumpImpl(Native Method) 1 (%)			1 (%)		
Non-deferrable	Rarked	Ox1668 s	sun/misc/Un	۷								

_____ Maximize the tool's windows and expand the columns in the TDMA "Thread Details" window to see more information.



Click the **Name** column heading to sort the threads by name. As we already know there is a problem with a WebSphere web application, it seems reasonable to only investigate WebSphere threads, specifically the "WebContainer" threads.

IBM Thread and Monitor	Dump Analyzer for Java			
File Analysis View Help				
	🐞 🔯 🤌 🐚 🗄 🚈		🚺 🔽 Floatable	
<u>.</u>		Thread Detail : javaco	re.20130211.064806.9068.00	001.txt
Name 🔺	State	NativeID	Method	Waiting Threads : 0
ThreadManager.JobsPro	🏅 Waiting on condition	0x24e4	java/lang/Object.wait(N 🔺	
TrLogger	🎖 Waiting on condition 🚽	0x2490	java/lang/Object.wait(N 🏲	•
UpstreamPing	🎖 Waiting on condition 🚽	0x25fc	java/lang/Thread.sleep	
VEUtilWorkDispatcher:0	🔏 Parked	0x27dc	sun/misc/Unsafe.park(
WLMMonitorSleeper: 0	🍸 Waiting on condition 👘	0x2474	java/lang/Object.wait(N	
WMQJCAResourceAdapt	🏹 Waiting on condition 👘	Ox1a44	java/lang/Object.wait(N	
WMQJCAResourceAdapt	🏅 Waiting on condition 👘	0x21e4	java/lang/Object.wait(N	
WMQJCAResourceAdapt	🏅 Waiting on condition 👘	0x256c	java/lang/Object.wait(N	
WMQJCAResourceAdapt	🍸 Waiting on condition 👘	Ox1 cbc	java/lang/Object.wait(N	
WMQJCAResourceAdapt	🏅 Waiting on condition 👘	0x25ac	java/lang/Object.wait(N	
WT=0	🍸 Waiting on condition 👘	0x1a30	java/lang/Object.wait(N	
WebContainer: 0	🔏 Parked	0x26b8	sun/misc/Unsafe.park(
WebContainer:1	🏅 Waiting on condition 👘	0x2768	sun/misc/Unsafe.park(
WebContainer:10	🍸 Waiting on condition 👘	0x27a4	com/ibm/io/async/Asyn	
WebContainer:11	🔏 Parked	0x275c	sun/misc/Unsafe.park(in in
WebContainer: 2	🔏 Parked	0x2758	sun/misc/Unsafe.park(1 <u>1</u>
WebContainer: 3	🏅 Waiting on condition 👘	0x2750	java/lang/Object.wait(N	
WebContainer: 4	🔏 Parked	0x25a0	sun/misc/Unsafe.park(
WebContainer: 5	🍸 Waiting on condition 👘	0x2784	java/lang/Thread.sleep	
WebContainer: 6	🏂 Runnable	0x260c	com/ibm/io/async/Asyn	Disaked bus 0
WebContainer: 7	🍸 Waiting on condition 👘	0x25dc	com/ibm/websphere/s	Blocked by . 0
WebContainer: 8	🍸 Waiting on condition 👘	0x2690	com/ibm/websphere/s	
WebContainer: 9	🍸 Waiting on condition 👘	0x25e8	com/ibm/io/async/Asyn	
Worker-0	🍸 Waiting on condition 👘	Ox15ec	java/lang/Object.wait(N	
XDTimer	🍸 Waiting on condition 👘	0x25e4	java/lang/Object.wait(N	
com.ibm.son.mesh.Peer	🏂 Runnable	0x1a5c	sun/nio/ch/WindowsSe	
java.net.MulticastSocket	🌋 Runnable	0x25f0	java/net/PlainDatagra	
pool-2-thread-1	📕 Parked	0x1504	sun/misc/Unsafe.park(🗸	
server.startup : 0	🕍 Parked	0x2330	sun/misc/Unsafe.park(
sonInThreadPool : 0	🎊 Runnable	0x2388	com/ibm/io/async/Asyn 🔽	
-			7.	

The exact states of the "WebContainer" threads in your Javacore may vary. In the screenshot above, the majority of thread states are purple in color and one is green. Purple threads are threads in state "Condition Wait" and green threads are "Runnable". An explanation of these states follows in the note below:

Note: Definition of Thread States



State	Name	TDMA Default Colour	Description	Tips and Tricks
R	Runnable	Green	The thread is able to run when given the chance	Many threads that you may expect to be "runnable" are actually shown as "condition wait" in Javacores from JDK 5 and above. This is actually by design as the JVM sets many threads to "Condition Wait" as it writes the Javacore. See "Unexpected Conditional Wait thread states in IBM Javacore" in the references for more details.
CW	Condition Wait	Purple	The thread is waiting. For example, because:	Common methods at the top of the stack trace for threads in this state include:
			A sleep() call is made	Object.wait: Waiting for a notify from another thread
			The thread has been blocked for I/O	Thread.sleep: Operating system
			A wait() method is called to wait on a monitor being notified	AsyncLibrary.aio_getioev2: Async I/O waiting on new work
			The thread is synchronizing with another thread with a join() call	SocketInputStream.socketRead0: Socket established, processing or waiting for data
S	Suspended	Salmon	The thread has been suspended by another thread	
Z	Zombie		The thread has been killed	
Ρ	Parked	Cyan	The thread has been parked by the Java concurrency API (java.util.concurrent).	

Select the **web container thread** that caused the hung thread warning in the WebSphere log file (refer to your SystemOut.log to confirm the exact title which may vary from the one shown in this lab document).



WebContainer : 5

🏹 🖓 Waiting on condition 🚽

0x2784 java/lang/Thread.sleep...

Verify the state of the thread and confirm it is not deadlocked. Deadlocking would be a possible cause of a hung thread and the JVM has automatic detection of this which would be reported in both the Javadump, and by TDMA. If the thread was in deadlock it would be colored grey in TDMA, and be entitled "deadlocked". In this lab, the hung thread is actually in state "Condition Wait" (purple). In addition, note that the first line of the Java stack shows "java/lang/Thread.sleep" which matches the stack trace shown in the SystemOut log file.

_____ Note the information in the panels on the right hand side. This shows "Waiting Threads", i.e. if the currently highlighted thread held a Java monitor and other threads were waiting for it, they would be displayed in this section. In this lab, no threads are blocked by the hung thread under investigation. In addition, the full Java and native stack traces are shown in the rightmost panel.

Waiting Threade : 0		A
wailing inreads . 0	Thread Name	WebContainer : 5
	State	Waiting on condition
Blocked by : 0	State	<pre>ivanting on condition at java/lang/Thread.sleep(Native Method) at java/lang/Thread.sleep(Thread.java:896) at com/lbm/websphere/samples/pbw/war/ShoppingBean.deliberateHungThread(ShoppingBean.java:415) at com/lbm/websphere/samples/pbw/war/ShoppingBean.performProductDetail(ShoppingBean.java:415) at com/lbm/websphere/samples/pbw/war/ShoppingBean.performProductDetail(ShoppingBean.java:415) at sun/reflect/NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:60(Compiled Code)) at sun/reflect/DelegatingMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:37(Compiled Code)) at sun/reflect/DelegatingMethodAccessorImpl.invoke(NetiveMethodAccessorImpl.java:37(Compiled Code)) at org/apache/webbeans/intercept/InterceptorHandler.invoke(InterceptorHandler.java:287) at org/apache/webbeans/intercept/InterceptorHandler.invoke(InterceptorHandler.invoke(NormalScopedBeanInterce at com/lbm/websphere/samples/pbw/war/ShoppingBean_\$\$_javassist_1.performProductDetail(ShoppingBea at sun/reflect/NativeMethodAccessorImpl.invoke(NativeMethod) at sun/reflect/NativeMethodAccessorImpl.invoke(NativeMethod) at sun/reflect/NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:37(Compiled Code)) at sun/reflect/Method.invoke(Method.java:611(Compiled Code)) at sun/reflect/MethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:37(Compiled Code)) at sun/reflect/MethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:37(Compiled Code)) at sun/reflect/MethodAccessorImpl.invoke(MethodExpressionImpl.java:37(Compiled Code)) at org/apache/wl/parser/AstValue.invoke(AstValue.java:266) at org/apache/wl/parser/AstValue.invoke(AstValue.java:266) at org/apache/myfaces/view/facelets/el/TagMethodExpression.invoke(TagMethodExpressionToMethodBinding at org/apache/myfaces/application/ActionListenerImpl.processAction/ActionListenerImpl.java:100) at javax/faces/component/Ul/NewRoot_proadcastUl/Ormmand.java:120) at javax/faces/component/Ul/NewRoot_broadcastEvents(Ul/NewRoot_java:275) at javax/faces/component/Ul/NewRoot_broadcastEvents(Ul/NewRoot_</pre>
	Java Stook	at javax/faces/component/UIViewRoot.processApplication(UIViewRoot.java:711)

Note:

It is clear that the when the Javadump was generated, the suspected hung thread was performing a Thread.sleep. However, this might be part of its design and perhaps after a short Thread.sleep it would



continue and finish its work, ending normally. A good way to see if a thread is actually progressing (albeit slowly) is to use multiple Javacores over a period of time. If the stack trace for this thread never shows a Java stack associated with incoming HTTP requests and only ever shows Thread.sleep, it will be increasing likely that the thread will not complete in a timely manner, meaning an application defect has been found.

Click File->Close all thread dumps, and close the "Thread Detail" window.



"File->Open thread dumps" and select both **Javacore.txt** files and click **Open**. TDMA will compare the Javacore file written by WebSphere's hung thread detection, and the other triggered manually a few minutes later via the WebSphere administration console.

📄 📄 javac	ore.20130211.064806.9068.0001.txt	🕒 Snap.20130211.040017.7648				
📄 javac	ore.20130211.064859.9068.0002.txt	📄 Snap.20130211.040130.7648				
📄 orbtro	📄 orbtrc.08022013.0824.37.txt 📄 Snap.20130211.040136.7648					
📄 orbtro	📄 orbtrc.11022013.0538.16.txt 📄 Snap.20130211.040156.7648					
np 🕒 orbtro	.11022013.0644.46.bt					
-						
File <u>N</u> ame:	211.064806.9068.0001.txt" "javacor	e.20130211.064859.9068.0002.txt"				
File <u>N</u> ame: Files of <u>T</u> ype:	211.064806.9068.0001.txt" "javacor All Files	e.20130211.064859.9068.0002.txt"				

Highlight both Javacores and right click. Select Compare Threads from the popup menu.

					Thread D	ump List
l	Name		Timestamp	Rur	nable/Total T	Free/Allocated He.
	javacore.20139 javacore.2013	D I	Fob 11.08:40:08 hread Detail	0474	23 23	64,807,488/134,2. 66,907,360/134,2.
I	(<u> N</u>	lonitor Detail			
I		🥜 <u>C</u>	ompare Threads			
ľ	A ¥	🖲 C	ompare Monitors			-
	• File r	X C	lose Thread Dumps		erver\profiles\App	Srv01\javacore.201
l)	🗶 C	lose <u>A</u> ll Thread Dum	ps		
1	 Cause 	ofth	hread dump : Dump F	Requi	ested By User (0)	0100000) Through (

As before, maximize the windows and expand out the columns. Order the first column by name to locate the "WebContainer" threads.

_____ The two columns show the state of the threads a few minutes apart. You will probably find that some or all of the "WebContainer" threads show a different stack in each column indicating the threads were executing different code at each point in time. However, the thread that has been identified as a hang suspect will definitely be highlighted in red meaning that TDMA also considers it a hang suspect. This is because it is showing exactly the same Java stack in both Javadumps. Therefore the thread is still paused in a Thread.sleep, confirming the suspicion that it really is hung and not just proceeding slowly.

WebContainer : 6	🌋 com/ibm/io/async/Async🔽 sun/misc/Unsafe.park(
WebContainer : 5	😰 java/lang/Thread.sleep(🝸 java/lang/Thread.sleep(
WebContainer : 4	🔀 sun/misc/Unsafe.park(🍸 java/lang/Object.wait(N

Note:

To find more detail about this hung thread, it can often be useful to relate the thread to a system dump. In addition this provides a correlation between the thread name seen in the Javadump, and the thread lds that would be traced by WebSphere's logging and tracing mechanism. The technique to achieve this mapping will be demonstrated later in this lab.

Impact²

Business. In Motion

A system dump contains information about the Java threads whereas an IBM heap dump (.phd) does not. The Memory Analyzer can present the same thread information when inspecting an IBM heap dump if both the heap dump and a Javacore from the same point in time are loaded into the tool simultaneously.

However a system dump is the only format that includes the full details about Java field names and values. Therefore this type of dump is the most useful for debugging Java issues, especially hung threads or other performance issues as demonstrated in this lab.

_ Close the TDMA windows, pressing the Yes button to exit.

_ Return to the ISA 5 tab in the browser.

_ From the Tools tab, select **Memory Analyzer (Desktop)** and click **Launch**.

Enter keyword <u>Filter</u> Reset	Memory Analyzer [Desktop]
Sort By: 📴 Tag: All Tags 🔻	🜔 Launch 🧻 Tool Help
↓ Garbage Collection and Memory↓ Visualizer (GCMV) [Desktop]	
Garbage Collection and Memory Visualizer (GCMV) [Report]	IBM Monitoring and Diagnostic Tools for Java
THealth Center	Memory Analyzer is a feature-rich Java heap leaks and reduce memory consumption.
👕 HeapAnalyzer [Desktop] 😚 🗊	
👕 Memory Analyzer [Desktop] 🛛 🗸 🗸 🗸	This tool is provided in three versions:
👕 Memory Analyzer [Report] 🛛 🗸 🛄	 as a report generating version that re
👕 Memory Analyzer Web Edition [Web] 🛛 😚 💽	and generates some zipped HTML rep as an interactive GUI version running
Thread and Monitor Dump Analyzer 😚 🗊	an interac



Problem Analysis

х

Run Memory Analyzer [Desktop] (Version 1.2.0.201208221220)

This tool is a desktop application. It will be launched using Java Web Start and will run on your workstation. Using the tool with files associated with ticket will require that you have access to the files from the workstation. If a file is located on a remote server, you can download the file to a local file system location or access the file through a shared storage area. Any existing local file may also be accessed by the tool.

In some cases, analysis of files on your workstation can noticeably degrade performance of other applications running on your workstation.

Click 'Submit' below to begin.

			[Submit	Cancel	
	 	 				-

_____ Click **Submit** and wait for the Memory Analyzer tool to download and launch. Note that the window does not take focus, so click the program on the taskbar.

🕒 IBM Monitoring and Diag...

__ Click File->Open Heap Dump.

_____ Navigate to path below and and open the system dump you previously triggered via the WebSphere admin console. The system dump file will end ".dmp".

C:\ISA5Beta2\ISA5\isa\cases\0002

Note:



Please be patient while Memory Analyzer parses the dump, approximately 3-5 minutes may pass with little nothing reported by the progress bar.

__ A warning may be displayed about a missing snapshot. Dismiss this warning by pressing OK

_ Click Cancel on the Getting Started Wizard.





_____ Click the Open Query Browser icon and select Java Basics->Thread Overview and Stacks

😫 core.20130211.064941.	9068.0003.dmp 🖾	
i 🖬 🖫 💀 🕅 🕶 🗸	월 • Q	
i Overview ⊠ ▼ Details	List objects Show objects by class by Path To GC Roots	
Size: 74.3 MB Classes: 2	Merge Shortest Paths to GC Roots Eclipse IBM Extensions	
66.8 MB	Java Basics Java Collections Leak Identification Immediate Dominators Show Retained Set Search Queries History	 References Class Loader Explorer Class Loader Explorer Customized Retained Set Duplicate Classes Finalizer Overview Find Strings Go GC Roots Group By Value Open In Dominator Tree Show As Histogram Surber Dreparties
	т	System Properties



Java Basics / Thread Stacks					
Thread Stacks (i) Enter a class na	me pattern (java.util.*)				
Argument	Value				
objects	0				
	\Box include class instance (if defined by a pattern)				
	more options				
1					
0	Finish Cancel				

__ Click Finish to show all Threads

In the table, locate the ThreadPoolWorker thread with name "WebContainer" that was reported as hung. Remember, the exact Id may differ from the screenshots in this lab.

i Overview 🍓 Thread Overview and Stacks 🛛	
Object / Stack Frame	Name
🔆 <regex></regex>	<regex></regex>
😐 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x3100e10	[HUNG] WebContainer : 7

_____ Highlight the thread and right click it. Try launching **Java Basics->Thread Details** to inspect all details about the thread.

+	ן 🎡 com.ibm.ws.util.ThreadPool\$Worker @ 0x307d6a8 א				
Ŧ	🏽 🍓 com.ibm.ws.util.Threa	na turna turna ante en	5] Wel		
+	🍓 com.ibm.rmm.ptl.tcha	inread Details	han_F		
+	🍓 com.ibm.rmm.ptl.tcha	List objects	han_F		
	ера :L	,			



👔 Overview 🌆 Thread Overview and Stacks 🔀 Details: Thread Details 💥

Thread Details

Thread Details

Thread [HUNG] WebContainer : 7

🔻 Thread Properties 🛐

Object / Stack Frame	com.ibm.ws.util.ThreadPool\$Worker @ 0x3100e10
Name	[HUNG] WebContainer : 7
Shallow Heap	144
Retained Heap	193,784
Context Class Loader	com.ibm.ws.classloader.CompoundClassLoader @ 0x267e0b8
Is Daemon	true
DTFJ Name	WebContainer : 7
JNIEnv	0x8d45800
Priority	5
State	[alive, sleeping, waiting]
State value	0xc1
Native id	7576
∑ Total: 12 entries	

Thread Stack

[HUNG] WebContainer : 7

at java.lang.Thread.sleep(JI)V (Native Method)

at java.lang.Thread.sleep(J)V (Thread.java:896)

at com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateHungThread()V (ShoppingBean.java:415)

at com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail()Ljava/lang/String; (Shoppi)

Note:

Key point to note here is that the thread stack that is hung as shown above points to the method "ShoppingBean.performProductDetail" which in turn calls the method "ShoppingBean.deliberateHungThread" as being responsible for calling Thread.sleep.

_____ Next, return to the "thread_stacks" window or tab and ensure the hang suspect thread is still highlighted as shown below. Look at the information in the "Inspector" panel on the left hand side and note that the thread has an attribute "isHung" which is set to true. The IBM Extensions for Memory Analyzer have also annotated the Thread Stacks report with the text **[Hung]** to further emphasize this. This confirms that the thread was still considered hung by WebSphere Application Server at the time the system dump was triggered.



i Overview 🍓 Thread Overview and Stacks 🔀				
Object / Stack Frame	Name			
<regex></regex>	<regex></regex>			
🖪 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x3100e10	[HUNG] WebContainer : 7			
🕀 🍓 com.ibm.rmm.ptl.tchan.receiver.PacketProcessor @ 0x1d36590	Pti_Tchan_PacketProcessor			
🛨 🍓 com.ibm.rmm.ptl.tchan.transmitter.PacketFireout @ 0x1d36138	Ptl_Tchan_Fireout			
🛨 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x26b9880	server.startup : 1			
🛨 🗄 🕀 🕀 🕀 🕀 🕀 🕀 😳 🕀	MessageAnnouncer for com.ibm.rmm.mtl.receiver.MStreamSetR@b2d4b17d			
🛨 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd58b8	WebContainer : 3			
🛨 🎕 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd5a68	WebContainer : 0			

🔍 Insp	ector 🛛					<	\$		
@ 0x3100e10									
🎝 Thre	🕗 ThreadPool\$Worker								
🖶 com	🖶 com.ibm.ws.util								
🖸 clas	s com.ibm.v	vs.util."	ThreadPoo	l\$Work	er@0	xe966	e8		
C _s java	a.lang.Threa	ъd							
👌 org.	eclipse.osg	i.intern	al.basead	aptor.D	efault	ClassL	oad	er @	
144	(shallow siz	e)							
193	,784 (retain	ned size	e)						
	o GC root: Thread								
o GC I	root: Threa	d		-					
o GC I Statics	root: Threa Attributes	d Class	Hierarchy	Value				ø	
o GC I Statics Type	root: Threa Attributes Name	d Class	Hierarchy Value	Value				ø	
o GC I Statics Type ref	root: Threa Attributes Name firstTask	d Class	Hierarchy Value	Value				\$	
GC I Statics Type ref ref	root: Threa Attributes Name firstTask wsThrea	d Class	Hierarchy Value null java.lan	Value g.Objec	t[44] (۵ 0×3	035	ø da8	
 GC I Statics Type ref ref long 	root: Threa Attributes Name firstTask wsThrea startTim	d Class Class	Hierarchy Value null java.lan 1360767	Value g.Objec 250781	t[44] (@ 0x3	035	ø da8	
 GC I Statics Type ref ref long long 	root: Threa Attributes Name firstTask wsThrea startTim created	d Class dL e Time	Hierarchy Value null java.lang 1360767	Value 9.Objec 250781 231406	t[44] (۵ Ox3	035	ø da8	
 GC I Statics Type ref ref long long long 	Attributes Name firstTask wsThrea startTim created isHung	d Class dL e Time	Hierarchy Value null java.lan 1360767 1360767 true	Value 9. Objec 250781 231406	t[44] (@ 0x3	035	ø da8	
 GC I Statics Type ref long long bool ref 	Attributes Name firstTask wsThrea startTim created isHung threadM	d Class dL e Time	Hierarchy Null java.lano 1360767 1360767 true	Value 9.Objec 250781 231406 2	t[44] (۵ 0x3	035	ø da8	
o GC I Statics Type ref ref long bool. ref ref	Attributes Name firstTask wsThrea startTim created isHung threadN activeW	d Class dL e Time um asT	Hierarchy Null java.land 1360767 1360767 true 000000a com.ibm.	Value g.Objec 250781 231406 Z ws.util	t[44] (; Threa	@ 0x3 dPool\$	035 Wa	ø da8 sThreac	

In the "thread_stacks" window, expand the thread by clicking on the "+" next to it and look for the method "ShoppingBean.performProductDetail" as seen in the stack earlier since it was responsible for calling Thread.sleep. Inspect the local variables and their values – in this case an instance of ShoppingBean and a RequestParameterMap as shown below.



🖃 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x3100e10
at java.lang.Thread.sleep(JI)V (Native Method)
at java.lang.Thread.sleep(J)V (Thread.java:896)
at com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateHungThread()V (ShoppingBean.java:41
😑 at com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail()Ljava/lang/String; (Shoppi
표 🗋 <local> com.ibm.websphere.samples.pbw.war.ShoppingBean @ 0x2d88120</local>
표 🗋 <local> org.apache.myfaces.context.servlet.RequestParameterMap @ 0x303e058</local>
∑ Total: 2 entries

This technique normally gives a lot of information about what the thread was processing. For example if it has opened a socket, the values will show the destination IP. Similarly, if the thread is performing a database query, the internal structures will reveal the SQL. This is the reason a system dump is recommended for debugging performance related issues such as hung threads.

In this scenario, looking for the HTTP request values could be of interest. Take a look at the screenshot and expand your stack frame to find the request parameters, specifically the "key" itemID. Select "value" below to see the value in the Inspector window on the left hand side.

🖸 Inspector 🛛 😫 🗖	😫 core.20130213.065809.7844.0002.dmp ⊠					
@ 0x3078698						
11 String[]						
🖶 java.lang	i Overview 🍓 Thread Overview and Stacks 🕱					
🔊 class java.lang.String[] @ 0x6a	Object / Stack Frame					
🔍 java.lang.Object	→ <regex></regex>					
🔊 com.ibm.oti.vm.BootstrapClass	😑 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x3100e10					
16 (shallow size)	at java.lang.Thread.sleep(JI)V (Native Method)					
164 (retained size)	at java.lang.Thread.sleep(J)V (Thread.java:896)					
o no GC root	at com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateHungThread()V (ShoppingBean.java:415)					
Statics Attributes >>2	at committee and the second					
Type Name Value	E 🚨 <a>local com.ibm.websphere.samples.pbw.war.ShoppingBean @ 0x2d88120					
ref [0] V0006	Image: A state of the state					
	servletRequest om.ibm.ws.webcontainer.srt.SRTServletRequest @ 0x2d06c68					
	class com.ibm.ws.webcontainer.srt.SRTServletRequest @ 0x2639b68					
	🗈 🛄 _paramStack com.ibm.ws.webcontainer.util.UnsynchronizedStack @ 0x2d07d80					
	🔄 🔄 🔄requestContext com.ibm.ws.webcontainer.srt.SRTRequestContext @ 0x2d07d98					
	J in com.ibm.ws.webcontainer.srt.http.HttpInputStream @ 0x2d07db0					
	J					
	🕒 🔄					
	🖃 🚺srtRequestHelper om.ibm.ws.webcontainer.srt.SRTServletRequest\$SRTServletRequestHelper @ 0x3035540					
	class> class com.ibm.ws.webcontainer.srt.SRTServletRequest\$SRTServletRequestHelper @ 0x1c805d0					
	this\$0 com.ibm.ws.webcontainer.srt.SRTServletRequest @ 0x2d06c68					
	Image:					
	Image:					
	🖻 📋 _parameters ava.util.Hashtable @ 0x3039170					
	🗄 🛵 <class> class java.util.Hashtable @ 0x6a1540 System Class</class>					
	□ □ elementData java.util.Hashtable\$Entry[11] @ 0x3078598					
	End Class > class ava.util.Hashtable≵Lntry[] @ Ux6a2248 System Llass					
	I ☐ [1] java.uti.Hashtable\$Entry @ 0x30/785e8					
	□ [2] java.util.Hashtable\$Entry @ 0x30/2600					
	H					
	C Tatal 2 aptrice					
	E L [U] Java.udi.hashcable\$Entry @ UX3U78618					

The sequence of references may look quite complicated, but to someone familiar with the JSF framework they would probably be quite familiar. You can see that by looking at JSF references relating to the request, eventually you find a "_parameters" Hashtable that has an "itemID" with value "V0006".

This may relate to a product Id in the ShoppingBean, and could determine the exact HTTP request that caused this particular thread to become hung while other "WebContainer" threads did not.

Collapse the "ShoppingBean.performProductDetail" stack frame you were inspecting.

Impact2013

Business. In Motion.



_____ In the stack frame of the same thread, scroll down and double click the **25 out of 51 entries** (your values may vary)

It javax.faces.component.UIViewRoot._process(Ljavax/faces/context/FacesContext;Ljavax/faces/event/PhaseId;Ljavax/faces/component/ at javax_faces_component_UIViewRoot.processApplication(Ljavax/faces/context/FacesContext;)V (UIViewRoot.java:711) Total: 25 of 51 entries

Scroll down to a method earlier in the servlet lifecyle. Highlight the method "com.ibm.ws.WebContainer.handleRequest". Note that there are similar methods above and below, so be sure to locate the correct method. In this method we should be able to find the HTTP request object, which can help to determine the URI that was being processed by the Web Container thread.

💼 at com.ibm.ws.webcontainer.WebContainer.handleRequest(Lcom/ibm/websphere/servlet/request/IRequest;Lcom/ibm/w	ebsphere/servlet/res
🖂 et com ihm wis webconteiner WSWebConteiner hendleDeauest() com/ihm/webenhere/servlet/reauest/IDeauest-) com/ihm	n luiahenhara learulati

Expand the references until the variable "request" of type

"com.ibm.ws.channel.imp.HttpRequestMessageImpl" is located. Of course it takes some experience to locate the right data structures.

	at com.ibm.ws.webcontainer.webapp.WebGroup.handleRequest(Liavax/servlet/ServletRequest:Liavax/servlet/ServletResponse:	
	at com.ibm.ws.webcontainer.WebContainer.handleRequest(Lcom/ibm/websphere/servlet/request/IRequest;Lcom/ibm/websphere/	
-		
	Image: Comparison of the second and the second a	
	E Channel com.ibm.ws.webcontainer.channel.WCChannel @ 0x2e980f8	
	🗉 🗋 serverName java.lang.String @ 0x2f8e130 luxor.hursley.ibm.com	
	🗉 🗋 link com.ibm.ws.webcontainer.channel.WCChannelLink @ 0x3094f88	
	🗉 🗋 iso com.ibm.ws.http.channel.inbound.impl.HttpInboundServiceContextImpl @ 0x3094fe8	
	🗊 🗋 response com.ibm.ws.webcontainer.channel.WCCResponseImpl @ 0x3095398	
	🖃 📄 request com.ibm.ws.http.channel.impl.HttpRequestMessageImpl @ 0x3095430	
	🗉 👩 <class> class com.ibm.ws.http.channel.impl.HttpRequestMessageImpl @ 0x1a3de68</class>	
	🕀 🗋 my¥ersion com.ibm.wsspi.http.channel.values.VersionValues @ 0x183be40	
	🗉 🗋 myScheme com.ibm.wsspi.http.channel.values.SchemeValues @ 0x183d470	
	Image:	
	Image:	
	Image: Image: The second se	
_	F C XCT HTTPOUTBOUNDREOUEST iava.lano.String @ 0x1a3e0a0 OutboundRequest	



Refer to the Inspector window on the left hand side. The request parameter includes the URI "/PlantsByWebSphere/shopping.jsf"

🔍 Insp	ector 🛛				⊈ □ □
 Inspector of a processing of the second secon					
Statics	Attributes	Class Hierarchy	Value		ø
Туре	Name		Value	•	▲
ref ref	myMeth myScher	od ne	com.	ibm.wsspi.http.channel.values.MethodV ibm.wsspi.http.channel.values.SchemeV	/alues /alues
ref	ef myURIBytes		/PlantsByWebSphere/shopping.jsf		
ref	ref myURIString		null		

Optionally. inspect some of the other "WebContainer" threads. They may have completely different thread stacks to the hung thread. If you find one that has a similar stack, i.e. with ShoppingBean.performProduct detail() close to the top, repeat the previous steps to locate the HTTP request information again. You will find that no other threads are dealing with requests where the item id is "V0006". It is becoming increasing clear that the HTTP request that caused the hung thread was for a PlantsByWebSphere item with an id of "V0006".

Optional Steps:

____ Double click the desktop shortcut to **ShoppingBean.java** to inspect the programming error.



Click Edit->Find and search for "V0006".



The deliberate error is indeed related to a product with Id "V0006". When a request for this product is received, the method "deliberateHungThread" is called which puts the thread to sleep for an hour.

```
} else if ("V0006".equals(invID)) {
       // ------
                                _____
       // User clicked on the Strawberries, let's pause
       // a webcontainer thread for a berry long time
       // ------
       // Comment out the method call below to remove this deliberate mistake
       this.deliberateHungThread();
    3
private void deliberateHungThread() {
   // -----
   // User clicked on the Strawberries, let's pause
   // a webcontainer thread for a berry long time
   // -----
                                           _____
  try {
      System.out.println("==> STARTING LONG SLEEP from Thread Id="
             +Thread.currentThread().getId()+" Thread name="
             +Thread.currentThread().getName());
      // Sleep for an hour
      Thread.sleep(3600000);
      System.out.println("==> ENDING LONG SLEEP from Thread Id="
             +Thread.currentThread().getId()
             +" Thread name="+Thread.currentThread().getName());
     } catch (InterruptedException e) {
        e.printStackTrace();
     }
}
```



Part 4: (Optional) Mapping a Thread Id in a Javadump to the output of WebSphere logging or trace (10 minutes)

Note:

If you need to correlate a thread Id from a Javadump to the output of WebSphere logging or trace file, there are different options according to the version of WebSphere Application Server being used. In a default configuration, WAS logging and tracing both print out a "thread identifier" with each log entry. This identifier has no relationship to the internal or native thread identifiers for that Java thread. Therefore, this identifier by itself is only useful to correlate messages on the same thread within logs and trace, but not to Javadumps. This behavior is by design, but thread identifiers may be aligned in future versions.

Correlation between Javadumps and WebSphere logging or trace can be achieved using one of these methods:

Method #1

Thread identifiers may be correlated at any one point in time using a system dump, like the one used in the previous section of this lab. The IBM Extensions for Memory Analyzer can be used to display an additional column, the "WAS Thread ID". You will try out this method in this part of the lab.

Method #2

You can also configure a server's trace output with the "Advanced" file formatting option to see more detailed trace information for use in troubleshooting and problem determination. Below is an example of the advanced format which prints the thread name with each trace entry:

[9/21/11 12:22:44:507 PDT] 00000022 I UOW=null source=com.ibm.ws.webcontainer.servlet class=com.ibm.ws.webcontainer.servlet.ServletWrapper method=init org=null prod=null component=null thread=[WebContainer : 0]

Method #3

Beginning in WebSphere Application Server Version 8.0 you can configure the server to use the High Performance Extensible Logging (HPEL) log and trace infrastructure instead of using SystemOut.log, SystemErr.log, trace.log, and activity.log files. If you are using HPEL, you can access all of your log and trace information using the LogViewer command-line tool from your server profile bin directory.

With HPEL enabled on a server, the logViewer command can be used to print both log and trace entries in the advanced format. In method #2, we had to explicitly change to advanced format. With HPEL, that information is always recorded, but we use different formatting options with the logViewer command to display the advanced format. For example:

<WAS_PROFILE_ROOT>/bin/logViewer -format advanced

Prints:

[9/26/11 14:37:06:493 PDT] 00000026 I UOW= source=com.ibm.ws.webcontainer.servlet class=com.ibm.ws.webcontainer.servlet.ServletWrapper method=init org= prod= component= thread=[WebContainer : 0]

SRVE02421: [WasSwat] [/swat] [/index.jsp]: Initialization successful.



Note: Methods 2 and 3 assume that thread names are unique and unchanging, which is not always the case. Method 3 is only available for WebSphere Application Server 8.0 and above.

For more information on these methods, see the references.

_____ In the Memory Analyzer window, return to the "Thread Overview and Stacks" window. (If you have closed this, open the report again with Query Browser (button) -> Java Basics -> Thread Overview and Stacks, then click **Finish**)

Locate the ThreadPoolWorker thread with the thread name that was previously reported as hung in the WebSphere logs, for example "WebContainer : 7".

i Overview 🍓 Thread Overview and Stacks 🛛	
Object / Stack Frame	Name
🛟 <regex></regex>	<regex></regex>
🛨 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x3100e10	[HUNG] WebContainer : 7
王 🎡 com.ibm.rmm.ptl.tchan.receiver.PacketProcessor @ 0x1d36590	Ptl_Tchan_PacketProcessor
王 🎡 com.ibm.rmm.ptl.tchan.transmitter.PacketFireout @ 0x1d36138	Ptl_Tchan_Fireout
🛨 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x26b9880	server.startup : 1
王 籀 com.ibm.rmm.mtl.receiver.MessageAnnouncer @ 0x1d36448	MessageAnnouncer for com.ibm.rmm.mtl.receiver.MStreamSetR@b2d4b17
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd58b8	WebContainer : 3
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd5a68	WebContainer : 0
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd59d8	WebContainer : 1
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd5708	WebContainer : 6
🛨 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd5ca8	ProcessDiscovery : 0
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd5798	WebContainer : 5
🛨 🍓 com.ibm.ws.util.ThreadPool\$Worker @ 0x2e86190	WebContainer : 13
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x308ed60	WebContainer : 10
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2e89ef8	WebContainer : 14
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x3100d80	WebContainer : 8
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x3100cf0	WebContainer : 9
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2e86220	WebContainer : 12
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2e862b0	WebContainer : 11
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2e727e0	WebContainer : 15
王 🇌 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd5828	WebContainer : 4
🛨 🏧 com.ibm.ws.util.ThreadPool\$Worker @ 0x2fd5948	WebContainer : 2
🛨 iava.lang.Thread @ 0x81e9d8	P=83234:O=1:CT
🛨 🇌 com.ibm.rmm.intrn.util.TaskManager @ 0x1d360a0	Rmm_TaskManager
	<u> </u>

Scroll to the right to reveal the "WAS Thread ID" column. This additional column has been added to this default report by the IBM Extensions for Memory Analyzer. (Note the column order has been rearranged in the screenshot below)

Name	WAS Thread ID	Native id	Shallow Heap
WebContainer : 11	0x0000009b	7716	144
WebContainer : 2	0×00000091	7872	144
WebContainer : 3	0x00000092	7868	144
WebContainer : 4	0x00000093	6852	144
WebContainer : 5	0x00000094	7880	144
WebContainer : 6	0×00000095	4656	144
WebContainer : 8	0x00000098	7900	144
WebContainer : 9	0x00000099	6808	144
Worker-0	0x0000001d	4524	104
XDTimer	0x0000007c	4860	104
[HUNG] WebContainer : 7	0x00000097	7892	144
com.ibm.son.mesh.Peer-tcp-port-11004	0x00000052	4836	248
java.net.MulticastSocket@f2a9de7c	0x0000007e	5156	96
pool-2-thread-1	0×00000077	6724	96
server startun · 1	0×0000006c	5216	144

The system dump shows the hung thread uses WAS Thread ID **0x00000097**.

This can be correlated to any log statements in the SystemOut.log, e.g.:

[2/13/13 6:54:11:343 PST] 00000097 SystemOut O ==> STARTING LONG SLEEP

The same id would be present in a WebSphere trace file.



Finally, to link the WebSphere logs to a thread in the Javadump, again you need the system dump to tie this together. Either use the thread name, "WebContainer : 7" or the Native Id. Note that the thread name is not necessarily constant and could be changed by the code. To use the Native Id, convert the Native Id column from Memory Analyzer to Hex, and locate this in the Javadump. In this case, the value **7892** is **1ED4** in hex, and can be located in the Javadump:

3XMTHREADINFO "WebContainer : 7" J9VMThread:0x15358400, j9thread_t:0x08EE1D40, java/lang/Thread:0x03131050, state:CW, prio=5				
3XMTHREADINFO1	(native thread ID:0x1ED4, native priority:0x5, native policy:UNKNOWN)			
3XMTHREADINFO3	Java callstack:			
4XESTACKTRACE	at java/lang/Thread.sleep(Native Method)			
4XESTACKTRACE	at java/lang/Thread.sleep(Thread.java:896)			
4XESTACKTRACE com/ibm/websphere/sample	at es/pbw/war/ShoppingBean.deliberateHungThread(ShoppingBean.java:415)			

_____ Optionally try to link together output from your hung thread in SystemOut.log with the Javadump, using the system dump as described above.



Reference Links

• IBM Support Assistant Information and Downloads:

http://www-01.ibm.com/software/support/isa/

- Verify Java SDK version shipped with IBM WebSphere Application Server fix packs http://www-01.ibm.com/support/docview.wss?uid=swg27005002
- Problem determination for javacore files from WebSphere Application Server http://www-01.ibm.com/support/docview.wss?uid=swg21181068
- Mapping Underlying Java Thread Identifiers to those in Logging and Trace http://www-01.ibm.com/support/docview.wss?uid=swg21418557