

Java Troubleshooting with ISA 5 using Health Center and Memory Analyzer



Java Troubleshooting with ISA 5 using Health Center and Memory Analyzer

What this lab	is about	2
Lab requirem	ients	3
What you sho	ould be able to do	3
Part 1:	Lab Set Up	4
Part 2:	IBM Support Assistant 5 Beta	5
Part 3:	Setup Health Center to Monitor a running WebSphere JVM	.15
Part 4:	Use Health Center to Investigate Application Errors	.21
Part 5:	WebSphere Application Server Health Management	.36
Part 6:	Trigger a the Memory Leak to activate the Heath Policy	.46
Part 7:	Using ISA and the Memory Analyzer to Analyze a Heapdump	60
Part 8:	(Optional) Using the IBM Extensions to Memory Analyzer for Further Memory Analysis	85
Part 9:	(Optional) Using the WebSphere Application Server Configuration Visualizer	100
Reference Li	nks	105

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 running WebSphere Application Server 8.5.

A badly implemented web application will be used to simulate common problems such as **memory leaks**, unexpected garbage collection cycles triggered by **System.gc()**, **large application objects** and **large HTTP session** sizes.

The lab demonstrates how ISA 5 facilitates team team-based collaboration, and provides server-level diagnostic tools to carry out analysis. It also describes the data needed to debug the issues, and introduces the Java problem determination tools available as part of the IBM Support Assistant including Health Center, and Memory Analyzer.

The lab also uses the new health management self protecting and self healing features of WebSphere Application Server 8.5 to dynamically monitor and manage servers, helping to preserve service even if something is about to go wrong.



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
 - ▶ IBM Monitoring and Diagnostic Tools for JavaTM Health Center (ISA 5 desktop tool)
 - ► IBM Monitoring and Diagnostic Tools for JavaTM Memory Analyzer (ISA 5 desktop tool, web edition 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:-

- Launch and configure Health Center to monitor a running WebSphere JVM
- Identify bugs in running code such as unnecessary calls to System.gc(), large object allocations and memory leaks
- Define a WebSphere Application Server health policy to automatically restart a server if a memory leak is detected
- Understand the basic techniques for debugging Java[™] memory issues with Memory Analyzer, using desktop, web and report editions
- Analyze a heap dump to determine those objects consuming the most heap space
- Use IBM extensions for Memory Analyzer to perform product specific memory analysis of a system dump

Part 1: Lab Set Up

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

Username :Administrator Password : Impact2013

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, or skip other sections according to your interests.



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.

_____ Launch the browser (Mozilla Firefox) and then use the bookmarks to load the ISA web interface as shown below.

Edit View Higtory Bookmarks Tools Help Mozilla Firefox Start Page Show All Bookmarks Ctrl+Shift+B Go to a Website Bookmark This Page Ctrl+D Subscribe to This Page Subscribe
I Mozilla Firefox Start Page Show All Bookmarks Ctrl+Shift+B So to a Website Bookmark This Page Ctrl+D Subscribe to This Page Subscribe to This Page Image: Ctrl+D
Bookmark This Page Ctrl+D Subscribe to This Page ••••••••••••••••••••••••••••••••••••
Bookmarks Toolbar
Recently Bookmarked Recent Tags
Mozilla Firefox
🔍 IBM Support Assistant 5
Plants By WebSphere
HebSphere Integrated Solutions Console
Unsorted Bookmarks



🕗 IBM Support Assistant - Mozilla Firefox 👘						
<u>File Edit View History Bookmarks Tools</u>	: <u>H</u> elp					
IBM Support Assistant	+					
Comparison of the second secon				🚖 ⊽ C 🚼 י	r Google	۶
IBM Support Assistant					Got Feedbac	a Help IBM.
Cases 🕨		•		Scan this Cas	e 🛛 🔻	Global Filter - Off 💌
🗈 Files 🧰 Tools 📗 Reports 📋	Overview 👪 Symptoms	🗐 Global Knowled	je Base Matches 🖁	🗜 Data Collector		
Tree View 🛡 🗀	E Tree View Carch File Conter					
	Name Filter 🐳		Filter	<u>Reset</u>		
Navigator	Name Symp	toms Knowledge Ba	First Timestam Last	Timestam, Size	Туре	Modified (PST)

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.



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

____ Complete the summary and description, and click the green tick as shown below.

Case ID:	[New]	×
Summary:	lab case 1	
Description	n:	
test case	e	
		:

_____ 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 as shown below on the Windows desktop. Open this file from the desktop, and note that it contains various WebSphere error messages and Java stack traces.



🗄 SystemOut.log - WordPad
<u>File Edit View Insert Format Help</u>
************* 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
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
the first and the property current Environment ************************************
[12/3/12 0:35:15:218 ESI] UUUUUUIE XARECOVETVUAT A WIRNUISII: Preparing to call xa recover on XARE
[12/3/12 0:35:15:232 ESI] UDUDDIE J2CAARESOURCE W J2CADUBIW: Error creating AR connection and Res
at com.ibm.ejs.jzc.zzc.akesourceracionys1.run(zzc.akesourceraciory.java.zzs)
at com ibm. ws.security.util.accesscontorite.uprivilega(accesscontorite.java.iio)
at com ibm.ejs.jzt.ozcaaresoutceractory.getaaresoutceractory.getaaresoutceractory.geta.iso)
at com ibm.ws.Transaction.JTA XARCOVEryData recover(X)RecoveryData java-572)
at com, ibm, tx, ita, PartnerLogTable, recover (PartnerLogTable, java; 389)
at com. ibm.tx. ita. RecoveryManager.resync (RecoveryManager.iava:1530)
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 Filter Reset							
Navigator			Name	Symptoms	Кг	nowledge Ba	First Timestam	Last Timestam	Size	Туре
+ 🗀 0002	Ŷ		Cpdtools						0 Bytes	directory
		r								
			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



___ Click the Scan this Case button as shown above.

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: 🔽

Click **Submit** to start the Scan.



Scanning the case takes a few 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	🏭 Sympto	oms	劗 Global Kr	owledge Base	Matches	🔡 Dat	a Collector
Enter keyword <u>Filter</u> Rese	<u>et</u>							
Sort By: 📴 Tag: All Tags	•							
arbage Collection and Memory Visualizer (GCMV) [Desktop]	V .			Se	lect a tool	from th	ie list t	o display (
Garbage Collection and Memory Visualizer (GCMV) [Report]	🗸 🛄							
👕 Health Center	V D		NCO		out additi		/15 0 V 01	
👕 HeapAnalyzer [Desktop]	69 🗊							
👕 Memory Analyzer [Desktop]	V							
👕 Memory Analyzer (Report)	🖌 🚹							
👔 Memory Analyzer Web Edition [Web]	65 💽							
Thread and Monitor Dump Analyzer (TMDA) [Desktop]	66 🗊							
Thread and Monitor Dump Analyzer (TMDA) [Report]	69 <u>III</u>							
WebSphere Application Server Configuration Visualizer	°9 🛄							

By now the scanning of the case should be finished, let's check the results. You can confirm this by clicking the **status** button as shown below.



Click the **Overview** tab. It shows useful system information determined from the log file(s) in the case.



🗎 Files 🧰 Tools 퉲 Repor	ts 📋 Overview 🟭 Symptom	is 🔞 Global Knowledge Base Matc	hes 📳 Data Collector		
Collapse All	Scan (Coverage	1	A Printable version	
	can of This Ticket				
The info in the following section: Click <u>Scan this Ticket</u> if you wish The most recent scan for thi Number of files included in t Number of symptoms discov First Timestamp discovered Last Timestamp discovered:	; was found by scanning the files i to re-scan the ticket. is ticket contains information the most recent scan: 6 rered in the most recent scan: :: 12/03/12 00:35:15:218 :: 12/03/12 10:09:30:545	in this ticket, that match the current related to files uploaded up to: : 3	filter. Please click the links Thu Dec 20 07:20:35 PST 2	to see the containing files. 2012	
WebSphere Platform 7.0.0.23 [N	ID 7.0.0.23 cf231218.02]				
<u> 1.6.0, Java Compiler = j9jit24, J</u>	Java VM name = IBM J9 VM				
✓ WebSphere Versions					
WebSphere Platform 7.0.0.23 [N	ID 7.0.0.23 cf231218.02]				
AIX, version 7.1					
▼ iFix Information					
{no iFix information found}					
- Database Information					

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

Files	💼 То	ols 📗 Reports	📋 Overview	👪 Symptoms	🗐 Global Knowledge Base Ma	tches 📳 Data Colle	ctor		
Symptom	Symptoms Filter 🔆 Enter keyword Filter Scan Coverage State Scan C								
Global Sco	Туре	Symptom				Symptom Occurrences	Knowledge Base Ma	First Occurren	Last Occuri
al.	WTRN0005W: The XAResource for a transaction participant could not be recreated and transaction recovery may not be able to complete properly. The resource was J2CXAResourceInfo: 10 12/03/12				12/03/12 09:07:18:039				
al 👘	AdHoc CWZZZ0001W: Possible abnormal startup - did not find 'open for e-business' 1 4			4	12/03/12 10:09:30:545	12/03/12 10:09:30:545			
đ	III ErrorMsg J2CA0061W: Error creating XA Connection and Resource java lang Exception: Parameter xaResIn lacks an RA wrapper and an RA wrapper could not be resolved 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.

IBM Software Accelerated Value Program



🗿 0002 - IBM Support Assistant 5.0 - Mozilla Firefo	4		
<u>File Edit View History Bookmarks Tools H</u> elp			
WebSphere Integrated Solutions Console 🛛 🗌 000	2 - IBM Support Assistant 5.0 × +		
Contemporary Contexter Con		🏠 ⊽ 😋 🚼 ▼ Google	۶ (م
IBM Support Assistant		Got Feedback	Help IBM.
Cases 🕨 [0002] lab case 2	▼	Scan this Case	Global Filter - Off
📔 Files 🧰 Tools 🗽 Reports 📋 Overv	iew 🚦 Symptoms 👹 Global Knowledge Base Matche	s 🔀 Data Collector	
Symptoms Filter 🚔 Enter keyword 🛛 👔	ter <u>Reset</u> Scan Coverage	l Showing 17 of 22	2 results 📃 Show Al
Global Sci Type Knowledge Base Entry	Symptom		Tool ID
APAR PM15719: THE TRANSACTION MAN AN XARESOURCE TO ROLLBACK	IAGER FAILS TO GET THE TRANSACTION. Multiple symptoms (2) matched by this entry	L	ocalKBSe:9
HI APAR FOR A TRANSACTION PARTICIPAN RECREATED.	T COULD NOT BE not be able to complete properly. The resource	ion participant could not be recreated and transaction recovery may use J2CXAResourceInfo :	ocalKBSe:16
II APAR PK81814: MESSAGE "OPEN FOR E- BEING PRINTED IN SYSTEMOUT.LO	BUSINESS" IS NOT G CWZZZ0001W: Possible abnormal startup - dia	I not find 'open for e-business'	ocalKBSe:1
PK83560: WLM RETURNING TARGE III APAR ENDPOINTS CAUSES SIB EXCEPTIC CWSIT0019E OR CWSIA0241E ERF	T WITH EMPTY INS RESULTING IN CWZZZ0001W : Possible abnormal startup - di ORS	I not find 'open for e-business'	ocalKBSe:3
Knowledge Base Matches Symptom Occurren	ces Symptom Details Containing Files		
Fype: APAR Found by Tool: LocalKBSearch Global Score: 1846 Label: PK83560: WLM RETURNING TARGET WITH Match ID: 3 Symptom IDs associated with this Match: 85 Description:	EMPTY ENDPOINTS CAUSES SIB EXCEPTIONS RESULTIN	S IN CWSIT0019E OR CWSIA0241E ERRORS	
Click on the link for more references: http://www.ibm.com/Search/?q=PK83560			
Abstract: WLM RETURNING TARGET WITH EMPTY EN	DPOINTS CAUSES SIB EXCEPTIONS RESULTING IN	I CWSIT0019E OR CWSIA0241E ERRORS.	2
Build ID: 5.0.0.0_Beta2_20121016-1409		© Copyright IBM Corp. 2011, 2012	. All rights reserved



Part 3: Setup Health Center to Monitor a running WebSphere JVM

Note:

In this section the Health Center tool will be used to profile the code. Health Center is a free low-overhead diagnostic tool for monitoring applications running on an IBM Java Virtual Machines. It consists of a workstation client and a JVM agent. The agent uses a small amount of processor time (less than 3%) and memory on the server. It is installed by default in an IBM JVMs at Java 5 SR8 or IBM Java 6 SR1 and above.

The easiest way to attach the Health Center client to the agent is to configure the JVM to run the agent at start up with JVM "-Xhealthcenter" generic argument. In this lab, that argument has already been configured for both application server JVMs.

It is also possible to use a feature called "late attach" to connect the Health Center client to an already running JVM "on the fly", without having pre-configured any command line properties. This lab does not demonstrate this technique, but more information can be found in the Health Center InfoCenter.

If not launched already, launch the browser and use the bookmarks to load the ISA web interface.

Switch to the Tools tab, select **Health Center** and click the **Launch** button.

👔 Files 🤷 Tools 📗 Reports 🛢	Overview	👪 Symptom:	; 🗐 Glo	bal Knowledge Base Matches	📳 Data
Enter keyword <u>Filter</u> Rese	<u>et</u>	He	ealth Ce	enter Version 2.0.0.2012	0508
Sort By: 📴 Tag: All Tags	•	0	Launch	(i) Tool Help	
Garbage Collection and Memory Visualizer (GCMV) [Desktop]	V 🗊	_			
Garbage Collection and Memory Visualizer (GCMV) [Report]	√ I I	(h Jai	e 18M Moni va(TM) - He	itoring and Diagnostic Tools to ealth Center is a lightweight to	or ool that
👕 Health Center	✓ □ monitors active IBM Virtual Machines for . minimal performance overhead and prov			ve 15m virtual machines for 30 ormance overhead and provid	ava with les live
👕 HeapAnalyzer [Desktop]	69 🗊	tur	ing recom	mendations and observations	
👕 Memory Analyzer [Desktop]	V D	Th	e Health Co	enter provides the following	
👕 Memory Analyzer [Report]	🗸 🚹	pe	rspectives:		
👔 Memory Analyzer Web Edition [Web]	69j 💿		• Classe	s - information about classes	heina
Thread and Monitor Dump Analyzer (TMDA) [Desktop]	66 🗊		loadec • Enviro	nment - details of the configu	vation
Thread and Monitor Dump Analyzer (TMDA) [Report]	69 🛄		and sy	estem of the monitored applic	ation
WebSphere Application Server Configuration Visualizer	65 🔟	0	more)		



Problem Analysis

х

Run Health Center (Version 2.0.0.20120508)

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**. The Health Center client will be downloaded and launched using Java Web Start – this will take a few minutes, then a new connection dialogue will appear. In the meantime, continue with the next steps.

Health Center: Connection wizard		
Enable an application for monitorin (Select Cancel to import an existing file.)	ng	
This wizard guides you through connecting the H	lealth Center to a currently running Java application.	
NOTE: Your application must be enabled for mon	itoring before Health Center can be connected.	
For instructions on how to setup an application t	o enable live monitoring, click on the link below before	continuing.
Enabling an application for monitoring		
	< <u>B</u> ack <u>N</u> ext > Einish	Cancel



In this lab, the Plants by WebSphere application will be accessed via the IBM HTTP Server. It is configured to load balance requests between the application servers, and maintain session affinity once a HTTP session has been established. Multiple instances of Health Center can be started to monitor multiple JVMs if necessary. However, for now, to keep things simple only one JVM will be monitored, and the other JVM will be stopped.

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 Types \rightarrow WebSphere Application Servers.



_ Select server1.



plication servers

Total 2

Application servers					
Use this page to view a list of the specific application server.	application servers in your	environment and the status	s of each of these servers. भ	'ou can also use this page to	change the status of a
🗄 Preferences					
New Delete Template	es Start Stop R	estart ImmediateStop	Terminate		
Select Name 🛟	Node 🗇	Host Name 💠	Version 🗘	Cluster Name 🖒	Status ሷ
You can administer the following	resources:				
Server1	server1 impact2013Node impact2013 ND 8.5.0.1 PlantsByWebSphereCluster #				
□ server2 impact2013Node impact2013 ND 8.5.0.1 PlantsByWebSphereCluster					
Total 2					

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

pplicatio	oplication servers 7				?	
 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. 						
Application servers Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.						
Nev	w Delete Template	s Start Stop R	estart ImmediateStop	Terminate		
Select	Name 🗢	Node 🗘	Host Name 💠	Version 🗘	Cluster Name 💲	Status ዕ
You can administer the following resources:						
	server1	impact2013Node	impact2013	ND 8.5.0.1	PlantsByWebSphereCluster	€
	server2	impact2013Node	impact2013	ND 8.5.0.1	PlantsByWebSphereCluster	8

Return to the Health Center client window. Click **Next** on the connection wizard.

IBM Software Accelerated Value Program

Health Center: Connection wizard	
JVM Connection Details Enter the details of the JVM you want to connect to (Select Cancel to import an existing file)	o.
Hostname: impact2013	_
Port: 1989	Scan next 100 ports for available connections
 No security MBean authentication SSL connection SSL connection with MBean authentication 	Username: Password: SSL keystore location: SSL keystore password: Browse
	< <u>B</u> ack <u>N</u> ext > <u>Fi</u> nish Cancel

The JVMs in this lab have been configured to run the Health Center agent on ports 1990 (server 1) and 1991 (server 2). Change the port to 1990 and Click **Next** to scan for these ports.

Highlight the Heath Center agent on port 1990 as soon as it appears and click **Finish** as shown below.

Health Center: Connection wizard				
Search for a JVM				_Ц
Select a detected JVM.				
Searching impact2013 on port 1994				
impact2013:1990				
impact2013:1991				
	< Back	Next >	Finish	Cancel
		Carrier C.		



_____ Switch to the connection tab and verify the connection status shows an active connection as shown below.





Part 4: Use Health Center to Investigate Application Errors

Note:

WebSphere Application Server is running the Plants by WebSphere sample web application which has been modified with some deliberate programming errors.

_____ First verify the Plants by WebSphere sample is running. Launch the browser and use the bookmarks to visit the Plants by WebSphere site in a new tab.

<u>B</u> ookmarks	<u>T</u> ools	Help		
Show All I	3ookmar	ks	Ctrl+Shift+B	
Bookmark	This Pa	ge	Ctrl+D	
Subscribe	to This	Page		
🛯 🖻 Bookmark	s Toolba	r		۲
Recently Bookmarked			×	
🔎 Recent Ta	Recent Tags			۲
Mozilla Firefox			F	
🔍 IBM Supp	IBM Support Assistant 5			
Plants By WebSphere				
HebSphere Integrated Solutions Console				
Unsorted	Bookma	rks		

Feel free to have a look around, but to avoid some deliberate mistakes **do not** click any products on the **Accessories** tab.





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 deliberate errors which you will diagnose using Health Center.

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



Click File->Open and navigate to "E:\Impact lab files\Jmeter scripts\Impact 2013-1.jmx". Click Open.



📟 Apache JMeter (2.8 r13931	62)
<u>File E</u> dit Search <u>R</u> un <u>O</u> p	otions <u>H</u> elp
📑 실 🤒 🛃	
Test Plan	Test Plan
	Namer Test Plan
upen 🗠	×
Look <u>i</u> n:	Jmeter scripts 🔹 🖬 🛱 🖪 📴
Impact 20	13-1.imx
File <u>N</u> ame:	Impact 2013-1.jmx
Files of Type:	JMeter [.imx]
The of The	current filmed
	Onon Cancol
	open Calcel

_____ Click **Run->Start.** Wait a few moments until the number of threads has reached 2, as indicated in the right hand corner of the Jmeter window.



_____ Switch to the Health Center window which should already be monitoring the WebSphere JVM (if not make a new connection with File->New Connection). Click the **Status** tab.



The Health Center status panel summarizes the main categories of data that Health Center is monitoring, and also summarizes current recommendations. Note that the data categories to be collected can also be customized from the Monitored JVM menu.

IBM Monitoring and Diagnostic Tools for Ja	va™- Health Center	
<u>File E</u> dit D <u>a</u> ta Monitored <u>J</u> VM <u>V</u> iew <u>H</u> elp		
] () 💐 🖂] 🚼 🛛 👄 🧻 🖻 🏦	🔤 🗞 🦝 🔤	
🛃 Status 🖾		
G <u>Classes</u>	Your application has loaded 223 classes.	
Environment d	The option -Xscmaxaot4M is not a supported option.	
🗊 Garbage Collection 🔇	The mean occupancy in the nursery is 9%. This is low, so the gencon policy is probably an optimal policy for this workload.	
<u> I/o</u>	No problems detected	
🔒 Locking	No problems detected.	
🗞 <u>Method Trace</u>	Currently tracing 1 methods	
IIII Native Memory	The current memory usage does not indicate any memory leaks.	
Profiling d	Method names could not be determined for an unknown reason.	
🗠 🗠 🕅	Your application has 116 threads	

_____ Start by analyzing where the WebSphere JVM is spending most of its time to see if any optimizations can be made. Click the **Profiling** link.

Note:

Health Center uses a "sampling based" method profiler. This means it takes a periodic sample of the methods running and reports which are consuming the most time in the JVM.

Sort the table of data by "Tree %" by clicking the **Tree %** column heading.



Within the Health Center, collections of methods are organized into structures called trees. You should see that in this case, a "**ThreadPool\$Worker.run()**" method represents the top of a tree which is consuming a very high percentage of the JVM's time.

However, also note the value in the "Self (%)" column, which indicates that the method "ThreadPool\$Worker.run()" is actually using a **low percentage** of the JVM's time. Therefore the problem must be in some code called by the "ThreadPool\$Worker.run()" method, i.e. further down the tree / method call stack.

As incoming HTTP requests are handled by WebSphere using the "ThreadPool\$Worker" class, this gives a clue that there could be something wrong in a running web application.

6 M	1ethod profile	8				
Filte	er methods:					Apply Gear
	Samples	Self (%)	Self	Tree (%) 🔻	Tree	Method 🔺
	2	0.59		66.7		com.ibm.ws.util.ThreadPool\$Worker.run()
	0	0.0		56.9		com.ibm.ws.util.ThreadPool.getTask(boolean)
	9	2.65	L	55.8		com.ibm.ws.util.BoundedBuffer.take()
	10	2.95	L	53.4		com.ibm.ws.util.BoundedBuffer.waitGet_(long)
	1	0.29		47.8		com.ibm.ws.util.BoundedBuffer\$GetQueueLock.await(long)
	105	31.0		47.5		java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject
	0	0.0		18.9		java.lang.Thread.run()
	23	6.78	1	15.9		com.ibm.ejs.util.am.AlarmManagerThreadCSLM.run()
	22	6.49	1	6.49	1	java.util.concurrent.locks.LockSupport.parkNanos(java.lang.Object, long
	0	0.0		5.9	1	com.ibm.io.async.ResultHandler\$2.run()
	6	1.77		5.9	1	com.ibm.io.async.ResultHandler.runEventProcessingLoop(boolean)
	0	0.0		5.6	1	com.ibm.ws.dcs.vri.common.JobsProcessorThread.run()
	0	0.0		5.6	1	com.ibm.ws.dcs.vri.common.JobsProcessorThread.executeJob(com.ibm.v
	8	2.36	L. C.	5.01	1	java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireQueued(j
	0	0.0		4.42	1	.com.ibm.ws.util.ThreadPool.execute(iava.lang.Ruppable)

Reorder the table to see results for individual methods by clicking Self %.

Note:

Now you can see the individual method "**deliberateSlowMethod**" in the **ShoppingBean** class is using a high percentage of the JVM's time. Note, the "Self" and "Tree" columns (without the % symbol) are a graphical indication of the same data.

Select the expensive method in the table by clicking it once.



🗞 Method profile	• X				
Filter methods:					Appiy Gear
Samples	Self (%) 🔻	Self	Tree (%)	Tree	Method
265	26.1		26.2		com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSlowMethod()

The "Invocation paths" tab shows what is calling the "deliberateSlowMethod". The "timeline" tab shows when the "deliberateSlowMethod" was invoked.

Also notice that Health Center has automatically identified the time consuming method and has highlighted this fact in the "Analysis and Recommendations" section.

IBM Monitoring and Diagnostic Tools for Ja	ava™- Health Cei	nter				
File Edit Data Monitored JVM View Help						
] 0 *] 🖂] 🖹 🛛 🖷 🗊 🚨 🔒	🎟 🐁 🖪 🍇	₩				
🚼 Status 🖾 🛄 Connection 📃 🗆	🔏 Method profile	• 🛛 🗌				
⊙ <u>Classes</u>	Filter methods:					Apply Gear
Environment (1)	Samples	Self (%) 📼	Self	Tree (%)	Tree	Method
Garbage Collection	265	26.1		26.2	1100	com.ibm.websphere.samples.phw.war.SboppingBean.deliberateSlowMethod()
	111	10.9		16.6		java.util.concurrent.locks.AbstractOueuedSynchronizer\$ConditionObject.await(long, jav
<u> 1/0</u>	23	2.26	1	2.26	1	java.util.concurrent.locks.LockSupport.parkNanos(java.lang.Object, long)
	23	2.26	10	5.61		com.ibm.ejs.util.am.AlarmManagerThreadCSLM.run()
i Locking V	14	1.38		1.38		java.lang.String.hashCode()
A Method Trace	13	1.28		18.9		com.ibm.ws.util.BoundedBuffer.waitGet_(long)
	11	1.08		2.36	1	java.util.HashMap.hash(java.lang.Object)
I Native Memory S	10	0.98		1.67		javax.faces.component.UIComponentBase.isIdValid(java.lang.String)
	9	0.89		19.6		com.ibm.ws.util.BoundedBuffer.take()
Profiling 💧	9	0.89		2.56	1	javax.faces.componentComponentAttributesMap.getPropertyDescriptor(java.lang.Str
1.	8	0.79		1.67		java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireQueued(java.util.concuri
🗠 <u>Threads</u>	8	0.79		1.67		javax.faces.component.UIComponent.pushComponentToEL(javax.faces.context.Faces
	7	0.69		3.05		org.apache.myfaces.view.facelets.tag.jsf.ComponentSupport.findChildByTagId(javax.f
🚍 Analysis and Barran de Name 🕅 🖳 🖻	6	0.59		1.48	_	java.util.concurrent.ConcurrentSkipListMap.doRemoveFirstEntry()
Analysis and Recommendations 🖓 👘	h h	11.59		14.4		org.apache.mytaces.view.tacelets.tag.ist.(omponent lagHapdieri)elegate.apply(iavax
A The method						
Contraction of the second of t	0			0		
ShoppingBean.deliberateSlowMethod() is	Sinvocation pa	iths 23 🧠 Call	led methods	🔏 I meline 🔛	Method t	race summary
consuming approximately 26% of the CPU	Methods that call	ShoppingBean.de	liberateSlow	Method()		
cycles. It may be a good candidate for	E 🚳 Shonning	Bean.deliberate	SlowMethod			
opennización.	E- M Shor	pindBean perform	ProductDet	sil (100%)		
		phigodaniporron		-l-0 (1000)		
			essormphin	UKEU (100 %)		
	±	NativeMethod	Accessor1mp	I.INVOKE (100%)		
	54 <u>-</u>					178M of 474M 🔟

_____As the Plants sample is clearly suffering with at least one slow method, type "com.ibm.websphere.samples" in the "Filter Methods" box and click Apply.

Note:

You can see only the "**ShoppingBean.deliberateSlowMethod**" in the Plants sample has a high value for the "Self (%)".



🗞 Method	l profile	23				
Filter met	hods:	com.ibm.websph	ere.samples			Apply Clear
Sar	mples	Self (%) 🔻	Self	Tree (%)	Tree	Method
	2623	50.9		50.9		com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSlowMethod()
	8	0.16		0.41		com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateLargeObjectAllocation()
	0	0.0		0.56		com.ibm.websphere.samples.pbw.war.ShoppingBean_\$\$_javassist_1.performShopping(
	0	0.0		0.33		com.ibm.websphere.samples.pbw.ejb.CatalogMgr.getItemInventory(java.lang.String)
	0	0.0		0.54		com.ibm.websphere.samples.pbw.ejb.EJSLocalNSFShoppingCartBean_f66be8da.getSize
	0	0.0		0.058		com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC()
	0	0.0		0.019		com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateLargeSession(javax.face
	0	0.0		51.8		com.ibm.websphere.samples.pbw.war.ShoppingBean_\$\$_javassist_1.performProductDe
_	0	0.0		0.64		com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean_\$\$_javassist_2.getSize()
	0	0.0		0.56		com.ibm.websphere.samples.pbw.war.ShoppingBean.performShopping()
	0	0.0		0.33		com.ibm.websphere.samples.pbw.ejb.EJSLocalNSLCatalogMgr_c8d70688.getItemInvent
	0	0.0		0.52		com.ibm.websphere.samples.pbw.ejb.CatalogMgr.getItemsByCategory(int)
	0	0.0		51.8		com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail()
	0	0.0		0.35		com.ibm.websphere.samples.pbw.war.ShoppingBean_\$\$_javassist_1.getProducts()
	Π	0.0		0.078		.com.ibm.websnbere.samples.nbw.war.ShonningBean \$\$_iavassist_1.getCart()

_ Press the Clear button.

Optional Steps:

_____ Double click the desktop shortcut as shown below to **ShoppingBean.java** to inspect the programming error.



_____ Click **Edit->Find** and search for "deliberateSlowMethod". Click **Find Next** to find the second occurrence of the search string.

Note:

The "deliberateSlowMethod" is invoked every time the user clicks on the tulips. The "deliberateSlowMethod" executes a tight loop which does not end until a 10 second wait time has passed. You have found the first deliberate mistake.



```
System.out.println("==> STARTING SLOW METHOD");
long timestamp = System.currentTimeMillis();
long target = timestamp + 10000;
System.out.println("timestamp="+timestamp);
System.out.println("resume at="+target);
while(timestamp < target) {
    timestamp = System.currentTimeMillis();
}
System.out.println("==> ENDING SLOW METHOD");
```

_____ Return to the Health Center window and select the **Garbage Collection** link to monitor the performance of garbage collection and memory usage



_____ Observe the current JVM heap size and used heap size after collection. After starting the load generator, you will notice the heap size and heap usage has increased but by now should have leveled out. There is currently no evidence of a memory leak.



Garbage Collection (GC) affects the entire application and tuning GC correctly can potentially deliver significant performance gains. Health Center identifies where garbage collection is causing performance problems and suggests more appropriate command line options.

_____ Observe the Analysis and Recommendations window. It warns of large object allocations which of course are likely to trigger frequent garbage collections and may indicate the application code can be optimized.



Note:

Health Center allows you to view the size, time and code location of a large object allocation request that meets specific threshold criteria.

To investigate this further, select Monitored JVM->Garbage Collection and allocation data collection





GC Data Collection Settings	_ 🗆 🗙
Configure Collection of GC and Allocation Data	
(i) Use these settings to control collection of verbose GC data and data for the analysis of object allocations requests	
🔲 Weite verkaar CC data ta filo	
Write Verbose GC data to file	
Enable collection of call stacks for sampled object allocation events	
Enable collection of object allocation events within thresholds	
Low threshold (bytes) High threshold (bytes) Upda	
Maximum number of stack entries to collect per event	
<u> </u>	ancel

Select the box **Enable collection of object allocation events within thresholds**, and set a threshold to focus on the biggest objects. The threshold values can be entered in bytes, kilobytes or megabytes. Enter **2m** for the low threshold and **10m** for the high threshold.

Enable collection of object allocation	n events within thresholds
Low threshold (bytes) 2m	High threshold (bytes) 10m Undo

____ Click Finish.

_____ Select the **Object Allocations** tab. Wait a few moments until large object allocation data is parsed by the Health Center client – **this could take up to 1 minute.**



6	o Heap and p	ause times	📖 Object allocati	ons 🖾 🔲 Samples by	request site 🔲 Samples by object	
	Filter by reque	est site:				Apply Gear
	Count 🔺	%	Allocations	Average size (KB)	Request site	
	64	20.0		5000	java.lang.String. <init>(String.java:</init>	311)
	64	20.0		5000	java.lang.StringCoding.decode (Strin	gCoding.java:505)
	64	20.0		2500	com.ibm.websphere.samples.pbw.wa	ar.ShoppingBean.deliberateLargeObjectAlloca
	127	39.7		7520	java.lang.StringBuilder.ensureCapaci	ityImpl (StringBuilder.java:342)

Some large object allocations met the defined size threshold. They are associated with creating a very large String.

_____Select the **Call Hierarchy** tab, then the **java.lang.StringBuilder** row in the table to see the stack trace leading to this large String allocation.

Summary 🔩 Call hierarchy 🖾 🎡 Timeline	
Image: StringBuilder.ensureCapacityImpl (StringBuilder.java:342) Image: StringBuilder.append (StringBuilder.java:208) (100%) Image: StringBuilder.append (StringBuilder.java:208) (100%) Image: StringBuilder.append (StringBuilder.java:183) (50.0%) Image: StringBuilder.append (StringBuilder.java:161) (100%) Image: StringBuilder.append (StringBuilder.java:161) (100%) Image: StringBuilder.append (StringBuilder.java:161) (100%) Image: StringBuilder.append (StringBuilder.append (StringBuilder.append (StringBuilder.java:346) (50.0%) Image: StringBuilder.append (StringBuilder.append (StringBuilder.appen	

Note:

Once again you will see the ShoppingBean class seems to be responsible, specifically a method named "deliberateLargeObjectAllocation". You have identified another deliberate error in the plants sample.

Optional Steps:

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



_____ Click Edit->Find and search for "deliberateLargeObjectAllocation". Click Find Next to find the second occurrence of the search string.



The "deliberateLargeObjectAllocation" is invoked every time the user clicks on the grapes. The "deliberateLargeObjectAllocation" creates a large Array and fills it with a String of characters.

The variables used are local to the method so once the request has finished; the large objects are eligible for garbage collection. Therefore this is not a memory leak, but the creation of this large object makes unnecessary work for the JVM's garbage collector.

```
System.out.println("==> STARTING LARGE OBJECT ALLOCATION");
// Handle to a large object. Not a memory leak, just a LOA that will be GC'd
HashSet largeObject = null;
largeObject = new HashSet();
long timestamp = System.currentTimeMillis();
byte[] array = new byte[2560[00];
Arrays.fill(array, (byte) 66);
largeObject.add(new String(array) + (timestamp));
```

System.out.println("==> ENDING LARGE OBJECT ALLOCATION");

_____ Return to the Heath Center window and click the **Summary** tab. This shows that System (forced) garbage collection is being called by some application code running in the JVM. There may also be a warning in the Analysis and Recommendations window, depending on how many times System (forced) garbage collection has been called.

📰 Summary 🔀 📲 Call hierarchy 🤬 Timeline					
<u>ـ</u>					
Minor collections - Mean garbage collection pause	35.6 ms				
Minor collections - Mean interval between collections	2325 ms				
Minor collections - Number of collections	457				
Minor collections - Total amount flipped 188475					
Number of collections triggered by allocation failure 463					
Proportion of time spent in Garbage Collection pauses	3.93%				
Proportion of time spent unpaused 96.1%					
System (forced) garbage collection count	84				

Note:

The Java code "System.gc()" forces a full garbage collection cycle. This is generally not recommended as the garbage collector should manage its own schedule of garbage collection, and does not always need to execute the compaction phase of GC which is the most CPU intensive. An application calling System.gc() will always trigger the most expensive compaction phase.

Health Center can be used to filter method innovations (for example, looking for System.gc). However, the method profiling works on a sampling basis, so infrequent method calls might not appear in Health Center's profiling results. Therefore, the best way to achieve this goal is to set a JVM trace, and use Health Center to extract this information from the JVM.

Impact

Business, In Moti

The JVM has been pre-configured with the following command line property:

-Xtrace:print=mt,methods={java/lang/System.gc},trigger=method{java/lang/System.gc,jstacktrace}

This causes the JVM to print a stack trace each time the java/lang/System.gc is executed. Normally, this output goes to the native_stderr.log. However, if the Health Center agent is also configured on the JVM (with -Xhealthcenter, like in this lab), then the output is instead directed to the Health Center Client.

In the Health Center window, click File->Export JVM Trace



____ Export the trace to "E:\trace.trc"

____ Open a command prompt, and issue command: E: (to change to the E: drive).

___ Then issue the following command:

java com.ibm.jvm.format.TraceFormat trace.trc



📾 Command Prompt	_ 🗆 🗵
Error: j9jit.4436 not in dat file: dat files old or from incorrect UM. Error: j9jit.9715 not in dat file: dat files old or from incorrect UM. Error: j9jit.14352 not in dat file: dat files old or from incorrect UM. 50% Error: j9jit.13584 not in dat file: dat files old or from incorrect UM. 50% Error: j9jit.3844 not in dat file: dat files old or from incorrect UM. Error: j9jit.3344 not in dat file: dat files old or from incorrect UM. Error: j9jit.3344 not in dat file: dat files old or from incorrect UM. Error: j9jit.11024 not in dat file: dat files old or from incorrect UM. Error: 10.10593 not in dat file: dat files old or from incorrect UM. Error: I0.13915 not in dat file: dat files old or from incorrect UM. Error: I0.13915 not in dat file: dat files old or from incorrect UM. Error: I0.7010 not in dat file: dat files old or from incorrect UM. Error: I0.7010 not in dat file: dat files old or from incorrect UM. Error: I0.8306 not in dat file: dat files old or from incorrect UM. Error: j9jit.12587 not in dat file: dat files old or from incorrect UM. Error: j9jit.12587 not in dat file: dat files old or from incorrect UM. Error: J9jit.12587 not in dat file: dat files old or from incorrect UM. Error: j9jit.2052 not in dat file: dat files old or from incorrect UM. Error: j9jit.2052 not in dat file: dat files old or from incorrect UM. Error: j9jit.12052 not in dat file: dat files old or from incorrect UM. 60% Error: I01 not in dat file: dat files old or from incorrect UM. 80% 90% 100% **** Number of formatted tracepoints = 505075 **** 1 buffers were discarded during trace data generation	
*** Formatting complete *** Formatted output written to file: trace.trc.fmt C:\temp>_	•

Note: This may take a few minutes.

_ Open the resulting trace.trc.fmt in an editor such as Notepad++

_ Search for java.lang.System.gc

File Edit Search View Encoding Language Settings Macro Run Plugir Image: System Find.n Files Ctrl+Shift+F Find In Files Ctrl+Shift+F Find Next F3 >java/lang/System.gc()V Bytecode static method jstacktrace: [1] java.lang.System.gc (System.java:312) [2] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jav) [3] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jav) [4] sun.reflect.Na [5] sun.reflect.Na [6] SystemGC start: ne Exclusive access: GlobalGC start: g1 Mark start Mark end Class unloading en Sweep start Sweep start Sweep end Commet starti rea Warg around	🔐 E:\trace.trc.fmt - Notepad++	
Find Ctrl+F Find in Files Ctrl+Shift+F Find Next F3 > java/lang/System.gc() V Bytecode static method jstacktrace: [1] java.lang.System.gc (System.java:312) [2] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.java) [3] com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.java) [4] sun.reflect.Na [5] sun.reflect.Na Find Find Replace Find in Files SystemGC start: ne Exclusive access: GlobalGC start: gl Mark start Mark end Class unloading st Class unloading en Sweep start Sweep end Comment Class Virag around	File Edit Search View Encoding	Language Settings Macro Run Plugin
Find in Files Ctrl+Shift+F Find Next F3 > java/lang/System.gc()V Bytecode static method jstacktrace: [1] java.lang.System.gc (System.java:312) [2] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jav [3] com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.jav [4] sun.reflect.Na [5] sun.reflect.Na ****** Context chan SystemGC start: ne Exclusive access: GlobalGC start: gl Mark start Mark end Class unloading st Class unloading st Sweep start Sweep start Sweep start Sweep start Sweep start Sweep start Sweep start i real	📔 🔓 📙 Eind	Ctrl+F 🧤 🧤 🧤 😪 🤜
<pre>>java/lang/System.gc()V Bytecode static method jstacktrace: [1] java.lang.System.gc (System.java:312) [2] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jav [3] com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.j [4] sun.reflect.Na [5] sun.reflect</pre>	Find in Files	Ctrl+Shift+F
<pre>>java/lang/System.gc()V Bytecode static method jstacktrace: [1] java.lang.System.gc (System.java:312) [2] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jav [3] com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.jav [4] sun.reflect.Na [5] sun.refle</pre>	🔚 System 🛛 Find Next	F3 pingBean.java
<pre>>java/lang/System.gc()V Bytecode static method jstacktrace: [1] java.lang.System.gc (System.java:312) [2] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jav [3] com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.j [4] sun.reflect.Na [5] sun.reflect.Na [6] Find Replace Find in Files Mark [5] sun.reflect.Na [6] Find Replace Find in Files Mark [6] Find Replace Find in Files Mark [6] Find Replace Find in Files Mark [6] Find All in All Opened [6] Documents [6] Document [6] Sweep start [6] Match whole word only [6] Match gase [6] Wing around [6] Find All in Current [6] Close [6] Sunce find and file file file file file file file file</pre>		
[1] java.lang.System.gc (System.java:312) [2] com.ibm.websphere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jav [3] com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.j [4] sun.reflect.Na [5] sun.reflect.Na Find [6] reflect.Na Find Replace Find Replace Find Next Count Find Next Count Mark start Mark end Class unloading st Class unloading en Sweep end Compact start: ree Wrag around	>java/lang/System.g	C()V Bytecode static method
[1] jata: Hing; Sybelang (Gybelang) (Gybelang) (Gybelang) [2] com. ibm. websphere. samples. pbw. war. ShoppingBean. deliberateSystemGC (ShoppingBean. jav [3] com. ibm. websphere. samples. pbw. war. ShoppingBean. performProductDetail (ShoppingBean. j [4] sun.reflect.Na [5] sun.reflect.Na ****** Context chan SystemGC start: ne Exclusive access: GlobalGC start: gl Mark end Class unloading en Sweep start Sweep end Commact start: rea Watch whole word only Match gase V Wrag around	[1] java lang Syst	em de (System java-312)
[3] com.ibm.websphere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.j [4] sun.reflect.Na [5] sun.reflect.Na [5] sun.reflect.Na ****** Context chan SystemGC start: ne Exclusive access: GlobalGC start: gl Mark start Mark end Class unloading st Class unloading st Sweep start Sweep end Compact start: rea Watch whole word only Match gase V Wrag around	[2] com.ibm.websph	ere.samples.pbw.war.ShoppingBean.deliberateSystemGC (ShoppingBean.jay
[4] sun.reflect.Na [5] sun.reflect.Na [5] sun.reflect.Na ****** Context chan SystemGC start: ne Exclusive access: GlobalGC start: gl Mark start Mark end Class unloading st Class unloading st Sweep start Sweep end Compact start: rea ✓ Wrag around	[3] com.ibm.websph	ere.samples.pbw.war.ShoppingBean.performProductDetail (ShoppingBean.j
[5] sun.reflect.Na ****** Context chan SystemGC start: ne Exclusive access: GlobalGC start: gl Mark start Mark end Class unloading st Class unloading en Sweep start Sweep start Sweep end Comment Watch gase ✓ Wrag around	[4] sun.reflect.Na	
<pre>****** Context chan SystemGC start: ne Exclusive access: GlobalGC start: gl Mark start Mark end Class unloading st Class unloading en Sweep start Sweep end Commact start: rea </pre>	[5] sun.reflect.Na	Find
SystemGC start: ne Eind what: java.lang.System.gc Find Next Exclusive access: GlobalGC start: gl Count Mark start Mark end Find All in All Opened Documents Class unloading st Match whole word only Find All in Current Document Sweep start Match gase Close Sweep end ✓ Wrag around Close	***** Context chan	Replace Find in Files Mark
Exclusive access: GlobalGC start: gl Count GlobalGC start: gl Count Mark start Find All in All Opened Mark end Find All in All Opened Class unloading st Find All in Current Sweep start Match whole word only Sweep end Match gase Commact start: rea Wrap around	SystemGC start: ne	Find what : java.lang.System.gc
GlobalGC start: gl Count Mark start Find All in All Opened Mark end Class unloading st Class unloading en Find All in Current Sweep start Match whole word only Sweep end Match gase Commact start: rea Wrap around	Exclusive access:	
Mark start Find All in All Opened Documents Mark end Find All in All Opened Documents Class unloading en Find All in Current Document Sweep start Match whole word only Sweep end Match case Commact start, real Wrap around	GlobalGC start: gl	Coun <u>t</u>
Mark end Documents Class unloading st Find All in Current Class unloading en Match whole word only Sweep start Match case Sweep end Close Commact start: real Wrap around	Mark start	Find All in All Opened
Class unloading st Find All in Current Class unloading en Match whole word only Sweep start Match case Sweep end Close Commact start, real Wrap around	Mark end	Documents
Sweep start Match whole word only Document Sweep end Match gase Close	Class unloading st	Find All in Current
Sweep end Compact start, rea Wrag around Cose	Sween start	Match whole word only Document
Compact start, rea Virag around	Sween end	Match gase Close
COMPACE STALL, ICA	Compact start: rea	✓ Wrap around



The formatted Java trace contains a stack trace for each time the java.lang.System.gc method was called.

Once again the source of the problem is ShoppingBean which calls a method "**deliberateSystemGC**". You have found another deliberate mistake in the plants sample.

Optional Steps:

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



_____ Click **Edit->Find** and search for "deliberateSystemGC". Click **Find Next** to find the second occurrence of the search string. This "deliberateSystemGC" method is invoked every time the user clicks on the gloves. The "deliberateSystemGC" method calls System.gc().

System.out.println("==> STARTING SYSTEM.GC");

System.gc();

System.out.println("==> ENDING SYSTEM.GC");

Switch to the Jmeter load generator window. Stop the load test by clicking File->Exit.

Note:

In subsequent parts of this lab, you will manually trigger the final deliberate error in the plants sample, but first you will set up a WebSphere Application Server Health Policy to help minimize the business impact of the error.

Part 5: WebSphere Application Server Health Management

Note:

WebSphere Application Server 8.5 provides significant new features for administration and operations staff. This includes incorporating the Intelligent Management functions from IBM WebSphere Virtual Enterprise which can minimize end-user outages, and use operational monitoring to control the production environment. One such feature is health management which monitors the status of the application servers, as well as sense and respond to problem areas before an outage occurs.

The health of an application server can be managed with a policy-driven approach that enables specific actions to occur when monitored criteria are met. In this section of the lab, a **health policy** will be defined which is triggered when memory usage exceeds a percentage of the heap size for a specified time, and **health actions** will be defined to correct the situation.

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

WebSphere. software		Welcome
View Alltacka	Cell=impact2013Cell, Profile=Dmgr	
	Global Health Controller Parameters	
Welcome		
Guided Activities Guided Activities Second Secon	Global Health Controller Parameters	· · · · · · · · · · · · · · · · · · ·
± Servers	Controller in cooperation with the defined Health Policies.	controller parameters. These par
Applications	Configuration Runtime	
± Jobs		
+ Services	General Properties	
+ Resources	Enable health monitoring	
+ Runtime Operations	Control cycle length	
⊞ Security	1 Minutes	
Operational policies	Maximum consecutive restarts	
Service policies	5	
Service policy topology	Restart timeout	
Health Policies	1 Minutes	
Custom Action	Minimum restart internal	
Autonomic Managers		
Application Placement Controller	U Minutes 💌	
Autonomic Request Flow Manager		
- (Health Collidoller)	Prohibited restart times	
± Environment	Add Remove	
\boxdot System administration	Start End Sun M	Aon Tue Wed Thu Fri Sat
🛨 Users and Groups		
🛨 Monitoring and Tuning		
E Service integration	Apply OK Reset Cancel	
+ UDDI		

Navigate to Operational Policies->Autonomic Managers->Health Controller


Note that health monitoring is already enabled. The Control cycle length specifies the time between consecutive health checks, it has been set to the lowest possible value for this lab.

Navigate to **Operational Policies->Health Policies**.

WebSphere. software			Welcome		
	Cell=impact2013Cell, Profile=Dm	gr			
View: All tasks	Health Policies		2 =		
Welcome					
Guided Activities Guided Activi	Health Policies				
	A health policy defines runt	ime behaviors to monitor and take corr	ective actions when these behaviors are		
Servers					
	+ Preferences				
± Jobs	New Delete				
+ Services	R R # 9				
+ Runtime Operations	Select Name 🗘	Reaction mode 🗘	Description 🛇		
± Security	None				
Operational policies	lotal U				
Service policies					
Service policy topology					
Health Policies					
Custom Action					
🖃 Autonomic Managers					
Application Placement Controller					
Autonomic Request Flow Manager					
Health Controller					

_____ Click **New** and name the policy "plantsHealthPolicy". Select the pre-defined health condition **Memory** condition: excessive memory usage as shown below and then click **Next**.



Create a new health policy	? =
Create a new health policy clusters, and dynamic clus	. Define the general properties, including the health condition, and the servers, ters to be monitored.
 Step 1: Define health policy general properties Step 2: Define health policy health condition properties Step 3: Specify members to be monitored Step 4: Confirm health policy creation 	Define health policy general properties * Name plantsHealthPolicy Description : Health condition © Predefined health condition Memory condition: excessive memory usage
	C Custom health condition
Next Cancel	

_____ First set the condition which should trigger the policy. Set the JVM heap size % to **75%**, and the offending time period to **1 second** (this is very aggressive, but we want the policy to be triggered quickly in this lab environment).

Define health policy health condition properties					
The memory condition: excessive memory usage will look for excessive memory utilization for each server that is a member of the policy. It detects general memory consumption by detecting if a JVM's heap size has grown over a configured percentage of the maximum heap size for a configured period of time.					
Health condition properties					
* JVM heap size 75 %					
 * Offending time period 1 Seconds • 					

_____ Next specify the actions the Health Manager should take if the policy is triggered. Select a Reaction Mode of **Automatic**.



	Health ma	anagement monitor reaction
	Reacti	on mode matic 💌
	Press the Add Action I	outton. Select the predefined action Take JVM heap dumps .
È	ake the following action	is when the health condition breaches
L	Add Action Remove	Action Move Up Move Down
Cr	reate a new health policy	? =
	Create a new health policy, clusters, and dynamic clust	. Define the general properties, including the health condition, and the servers, ters to be monitored.
	Step 1: Define bealth policy general	Select action
	Step 2: Define health policy health condition properties	Select action Predefined health policy action Take JVM heap dumps
	Step 2.2: Select action Step 2.2: Select target	C Custom health policy action Create new custom action 💌
	Step 3: Specify members to be monitored	
	Step 4: Confirm health policy creation	
	Previous Next Ca	ancel

__ Click Next



health policy general	Define he	efine health policy health condition properties								
properties Step 2: Define health policy health condition properties Step 3: Specify members to be	The mem utilization memory c configured	The memory condition: excessive memory usage will look for excessive memory utilization for each server that is a member of the policy. It detects general memory consumption by detecting if a JVM's heap size has grown over a configured percentage of the maximum heap size for a configured period of time.								
monitored	Health	condi	tion properties							
Step 4: Confirm health policy	* 3	VM hea	np size %							
creation	* (* Offending time period								
	1	1 Seconds 💌								
	Health management monitor reaction									
	Reaction mode Automatic									
	Take	the f	ollowing actions wh	en the healtl	condition breaches					
	Ac	ld Actio	n Remove Action	Move Up	Move Down					
	D	D								
	Sele	t Step	Action	Target server	Target node					
		1	Restart server	Sick server	Node hosting sick server					
		2	Take JVM heap dumps	Sick server	Node hosting sick server					

_____ To ensure heap dumps are triggered before the server is restarted, select **Take JVM heap dumps** and press the **Move Up** button.



	Step 1: Define bealth policy general	Defin	Define health policy health condition properties						
properties									
	Step 2: Define health policy health condition properties	The utiliz men confi	memor ation fo nory cor igured p	y cond or ead osump oercen	lition: excessive memo h server that is a mem tion by detecting if a J tage of the maximum	ory usage will ber of the pol VM's heap siz heap size for	look for excessive memory icy. It detects general e has grown over a a configured period of time		
	Step 3: Specify members to be monitored	He	ealth c	ondit	ion properties				
	Step 4: Confirm health policy creation		* J∨ľ 75	M hea	p size %				
		* Offending time period 1 Seconds							
		Health management monitor reaction							
		Reaction mode Supervise 🔽							
		L G	Take t	he fo	llowing actions whe	en the health	n condition breaches		
			Add	Actior	n Remove Action	Move Up	Move Down		
				D					
			Select	Step	Action	Target server	Target node		
				1	Take JVM heap dumps	Sick server	Node hosting sick server		
				2	Restart server	Sick server	Node hosting sick server		

____ Click Next



Create a new health policy Create a new health policy. Define the general properties, including the health condition, and the servers, clusters, and dynamic clusters to be monitored. Step 1: Define health policy general properties Specify members to be monitored Select the members to monitor with this health policy. If you defined health rules that only apply to certain Step 2: Define health policy health condition properties member types, this list of available members is automatically filtered based on your previously defined health rule for this health policy. Step 3: Specify members to be monitored Memberships Filter by Select a member type 💌 Step 4: Confirm health policy creation Available for membership Members of plantsHealthPolicy: ----------Add >> << Remove Previous Next Cancel

Select Filter by Clusters, and highlight PlantsByWebSphereCluster

Create a new health policy Create a new health polic be monitored.	Define the general properties, including the health condition, and the servers, clusters, and dynamic clusters	? _
Step 1: Define	Specify members to be monitored	
Step 2: Define health policy health condition properties	Select the members to monitor with this health policy. If you defined health rules that only apply to certain member types, this list of available members is automatically filtered based on your previously defined health rule for this health policy.	
→ Step 3: Specify members to be monitored	Memberships Filter by Clusters	
Step 4: Confirm health policy creation	Available for membership Members of plants Health Policy:	
Previous Next (ncel	

_ To add all members of the PlantsByWebSphereCluster to the health policy, click Add



Create a new health policy Create a new health policy. Define the general properties, including the health condition, and the servers, clusters, and dynamic clusters to be monitored. Step 1: Define health policy general properties Specify members to be monitored Select the members to monitor with this health policy. If you defined health rules that only apply to certain Step 2: Define health policy health condition properties member types, this list of available members is automatically filtered based on your previously defined health rule for this health policy. Step 3: Specify members to be monitored Memberships Filter by Clusters -Step 4: Confirm health policy creation Available for membership Members of plantsHealthPolicy: -----. PlantsByWebSphereCluster (Clusters) ۸ Add >> << Remove Previous Next Cancel

_ Click Next, then Finish

_ Click the Save link to save the changes to the Deployment Manager master repository.

Select System Administration->Nodes

ides				?				
Nodes								
Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to this list by clicking Add Node.								
 Preferences 								
Add Node Remove Node Forc	e Delete Synchronize	Full Resynchronize Stop						
Select Name 🛟	Host Name 🗘	Version 🗘	Discovery Protocol 🗘	Status ሷ				
You can administer the following resou	irces:							
impact2013CellManager_	impact2013	ND 8.5.0.1	тср	↔				
impact2013Node	impact2013	ND 8.5.0.1	тср					
Total 2								

Select **impact2013Node**, and press **Synchronize**. (The Nodes may have already synchronized automatically).

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

pplication servers				? -				
Application servers								
Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.								
🛨 Preferences								
New Delete Templates Start Stop Restart ImmediateStop Terminate								
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	us feedback	
Servers	status feedback	
Server s	status provides information about events that occur while the server stops.	
■ i	impact2013Node:server1 🛛 🔓 🔓 🐨	
-	Server impact2013Node/server1 stopped successfully. The collection may need to be refreshed to show the current server status. View JVM logs for further details.	
OK		

____ Wait for the server to stop, then Click **OK**



In the next section of the lab, you will cause a memory leak which triggers the health policy and its associated actions. To demonstrate how diagnostic data (heap dumps) can be collected, without a significant impact to the application's users, all cluster members should be running.

Select Servers->Clusters->WebSphere Application Server Clusters.

____Select **PlantsByWebSphereCluster**, and press Start. This starts both servers in parallel.

Wait for the servers to start which will take a few minutes. You may read ahead, but you must wait for the servers to start before attempting any actions in the lab. You can check the status on the Servers->Server Types->Application Servers page, by pressing the refresh icon.

Applicatio	on servers					?.			
Applic	Application servers								
Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.									
🗄 Pre	ferences								
Ne	New Delete Templates Start Stop Restart ImmediateStop Terminate								
R	6 # 2								
Select	Name 🛟	Node 🗘	Host Name 🗘	Version 🗘	Cluster Name 🗘	Status ሷ			
You	an administer the followin	ig resources:							
	server1	impact2013Node	impact2013	ND 8.5.0.1	PlantsByWebSphereCluster	€>			
	server2	impact2013Node	impact2013	ND 8.5.0.1	PlantsByWebSphereCluster	€			
Total	2								



Part 6: Trigger a the Memory Leak to activate the Heath Policy

Note:

In this section you will trigger a deliberate error in the application which leaks memory. This will trigger the Health Policy defined in the previous section, causing heap dumps to be written, **before** the JVM exhausts all the available heap space. As the application runs in a cluster of two application servers, the corrective actions of the health policy (heap dumps followed by JVM restart) will have minimal impact to the user experience. Having the health policy action restart the application server is more desirable than letting the application server run completely out of memory, causing an out of memory exception. For example, when stopping the application server in a controlled way, it will be 'quiesced' meaning current requests will be completed before the server is stopped.

The heap dumps produced should be adequate to diagnose the memory leak in the next section.

_____ Return to the existing Health Center window and click the **connection** tab - the connection will have been dropped when server1 was restarted in the previous section.



Click **File->New** Connection to re-establish a connection.

Health Center: Connection wizard	- 🗆 🗙
Enable an application for monitoring (Select Cancel to import an existing file.)	
This wizard guides you through connecting the Health Center to a currently running Java application.	
NOTE: Your application must be enabled for monitoring before Health Center can be connected.	
For instructions on how to setup an application to enable live monitoring, click on the link below before con	ntinuing.
Enabling an application for monitoring	
< Back Next > Einish Ca	ncel

__Click Next

Health Center: Connection wizard JVM Connection Details Enter the details of the JVM you want to connect to (Select Cancel to import an existing file) Hostname: impact2013	□.
Port: 1989 Security No security MBean authentication SSL connection SSL connection with MBean authentication	Scan next 100 ports for available connections Username: Password: S5L keystore location: Browse S5L keystore password:
	< Back Next > Einish Cancel



____ Click Next

_ Highlight the Heath Center agent on port 1990 (server1) and click **Finish**.

Health Center: Connection wizard				
Search for a JVM				_11
Select a detected JVM.				
Searching impact2013 on port 1994				
impact2013:1990				
impact2013:1991				
				_
	e Davela	Marchin	Cipirk	
	<u> </u>	Mext >		

Note:

The goal of this section is to monitor the memory footprint, and see the health policy trigger its actions. However, with two JVMs it is more difficult to know which one serviced the client request, and therefore which JVM could be leaking memory.

To determine this, you will use two instances of Health Center, and connect to each JVM on different port numbers. After further clicks on the Wheelbarrow, it should become obvious which JVM is experiencing the memory leak. Note the IBM HTTP Server is configured with session persistence, so all requests in the current session will be serviced by the same JVM.



_____ Return to the ISA tab in the browser. Launch a second instance of Health Center by selecting the **Tools** tab, choosing **Health Center** and clicking the **launch** button.

Files	💼 Tools	<u> R</u> eports	📋 Overview	👪 Symptoms	🗐 Glob	al Knowledge Base Matches	📳 Data
🛟 Enter ko	eyword	<u> </u>	<u>eset</u>	Hea	alth Ce	enter Version 2.0.0.2012)508
Sort By:	↓ <mark>a</mark> Tag:	All Tags	•	04	aunch	(i) Tool Help	
T Garbage Visualize	Collection an er (GCMV) [De	d Memory :sktop]	V]		-		
l Garbage Visualize	Collection an er (GCMV) [Re	d Memory :port]	🗸 🛄	The Java	IBM Monit (TM) - He	toring and Diagnostic Tools fo alth Center is a lightweight to	or ool that
👕 Health C	enter		√]	mon	itors activ mal perfoi	e IBM Virtual Machines for Ja rmance overhead and provid	iva with les live
👕 HeapAna	alyzer (Deskto	P[65 🗊	tunir	ng recomm	nendations and observations	
👔 Memory	Analyzer (De	sktop]	V 🗾	The	Health Ce	nter provides the following	
👔 Memory	Analyzer [Re	port]	🗸 🛄	pers	pectives:		
👔 Memory	Analyzer Wel	b Edition [Web]	69 💽		 Classes 	s - information about classes	beina
Thread a (TMDA) [and Monitor D [Desktop]	ump Analyzer	69 🗊		loaded Enviror	ment - details of the configu	ration
Thread a (TMDA)	and Monitor D [Report]	ump Analyzer	69 🛄		and sy:	stem of the monitored application of the control of the monitored application of the monitored applicat	ation
T WebSph Configur	ere Applicatio ration Visualiz	on Server er	65 🛄	(m	ore)		

___ Click Submit

_____ A second Health Center client will be launched using Java Web Start. This will take a few moments, wait for the new connection dialogue to appear.

Problem Analysis



Run Health Center (Version 2.0.0.20120508)

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.



Health Center: Connection wizard				
Enable an application for monitori (Select Cancel to import an existing file.)	ing			
This wizard guides you through connecting the NOTE: Your application must be enabled for mo	Health Center to nitoring before H	a currently runn lealth Center can	ing Java applicat be connected.	ion.
For instructions on how to setup an application	to enable live mo	onitoring, click on	the link below be	fore continuing.
	< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel
Click Next				

_ U ICI

Health C JVM Con Enter the d (Select Car Hostname:	enter: Connection wizard nection Details letails of the JVM you want to connect to icel to import an existing file) impact2013	o.	
Port: Security - No si O MBea O SSL (SSL (1989 ecurity an authentication connection connection with MBean authentication	Scan next 100 ports for av Username: Password: S5L keystore location: S5L keystore password:	Browse,
		< <u>B</u> ack <u>N</u> ext > <u>Fi</u> nish	Cancel



Highlight the Heath Center agent on port 1991 (server2) and click finish.

Health Center: Connection wizard				
Search for a JVM				
Select a detected JVM.				
Searching impact2013 on port 1997				
impact2013:1990 (in use)				
impact2013:1991				
		1 1		
	< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel

Ensure that the Plants by WebSphere application can be accessed through the browser as follows.(Use the bookmark to access it if needed)





PLANTS BY WEBSPHERE Your shopping cart is currently empty Flowers Fruits & Vegetables Trees Accessories SHOPPING CART LOGIN HELE Gardens 🔲 IBM Monitoring and Diagnostic Tools for Java™ - Health C _ 🗆 X _ 🗆 X File Edit Data Monitored JVM View Help File Edit Data Monitored JVM View Help 0 🜒 🖂 🚼 🕒 🖷 🗓 🔒 📖 🗞 🚵 🖄 0 🕴 🖂] 🚼 | 🖸 🖷 🛍 🔟 🔒 📖 🗞 🚵 🗠 🖹 Status 🗙 🖳 Connec... 🕆 🗖 🆓 Heap and p... 💥 📟 Object alloc... 🚍 Samples by ... 🚍 Samples by ... 🗮 🖬 Status 🕱 🎩 Connec... 🕆 🗖 🆓 Heap and p... 🕱 📟 Object alloc... 🚍 Samples by ... 🚍 Samples by ... 🖓 🗖 **O** <u>Classes</u> **O** Classes -Used heap (after collection) -Used heap (after collection) Environment ۵ ----Heap size Environment ۵ ----Heap size -Pause time -Pause time 📋 Garbage Collection 📀 🗓 Garbage Collection 📀 120 -120 120 I/O 0 I/O 800 (s 100 80 60 -100 -100 Locking 0 🔒 Locking 0 (Yau 600 -60 [≦] SIS -60 ime Method Trace Method Trace 60 🗟 MB Pe -60 400 0 INALIVE Memory INALIVE Memory 40 -40 4N b Profiling ۵ 🔥 Profiling ۵ 200 20 -20 -20 0 0 M Threads M Threads 0 0 100:00 120:00 140:00 160:00 180:00 200:00 220:00 10:00 20:00 30:00 40:00 50:00 60:00 70:00 🗖 Analysis and Recom... 🛛 🗖 🗖 🗖 Analysis and Recom... 🔀 🖵 🗖 elapsed time (minutes) elapsed time (minutes) ۸ 😵 The mean occupancy in the 😵 The mean occupancy in the - 8 - 8 📰 Summary 🛛 📲 Call hierarchy 🚇 Timeline 📰 Summary 🔀 🔩 Call hierarchy 🚇 Timeline nursery is 8%. This is low, so the nursery is 9%. This is low, so the gencon policy is probably an optimal gencon policy is probably an optimal . . olicy for this workload. oolicy for this workload. Concurrent collection count Concurrent collection count Default (gencon) GC Mode Default (gencon) GC Mode Global collections - Mean garbage collection pause Global collections - Mean garbage collection pause 132 ms 0 ms Global collections - Number of collections Global collections - Mean interval between collections 11752 ms Largest memory request 104 KB Global collections - Number of collections 104 KB Minor collections - Mean garbage collection pause 19.8 ms • Largest memory request • 126M of 141M 📋 69M of 101M 🛛 📋 Connected to impact2013:1990. 18 MB received. Connected to impact2013:1991. 17 MB received.

____ Align the windows so both Health Center clients, and the browser can be viewed simultaneously.

In both Health Center client windows, click the **status** tab, click the **Garbage Collector** link, and select the **Heap and Pause Time** tab on the right hand side (if necessary).

_____ Use the browser to access the Accessories tab and then click the **Wheelbarrow** product on the Accessories page of the Plants by WebSphere application as shown below. This action triggers a deliberate error which causes the application's memory footprint to increase.



🕗 Plants By WebSphere Product Detail - Mozilla Firefox <u>File Edit View History Bookmarks</u> Tools Help Plants By WebSphere Product Detail х 0003 - IBM Support Assistant 5.0 😔 WebSphere Integrated Solutions Console +(Incalhost/PlantsByWebSphere/shopping.jsf ∰ ▼ C PLANTS BY WEBSPHERE Flowers Fruits & Vegetables Trees Accessories Home >Accessories Wheelbarrow Just like Grandpa used to have! Shiny red wheelbarrow with epoxy coated steel bin and wooden handles. Tire is solid with thick treads that grip rough, wet surfaces. Large capacity - 3 Cu.Ft. capacity, 150 Lb. maximum load Item Selection: ITEM# DESCRIPTION PRICE QUANTITY A0011 Assembled \$29.00 1 Add to cart

_____ In the browser window, click the **Accessories** tab. Click the **Wheelbarrow** to view the product, then return to the accessories page by clicking the **Accessories** tab. Repeat this step a further 3 times, giving a total of **4 wheelbarrow views** in this step. There is no need to add the item to the cart, just selecting it from the tools menu will trigger the memory leak.





Return to the Health Center windows and wait for them to refresh their data a couple of times (there is a 10 second pause between each refresh). Take a look at the Garbage Collection statistics, you should see the used heap has increased significantly for one of the JVMs. It seems there is a large memory leak over a short period of time in that JVM.







Health Center can also be used to trigger a heap dump which is useful for analyzing memory leaks. However in general it is often beneficial to let the memory leak grow as large as possible, making the problem as easy to diagnose as possible,

In the health center window where the memory increase was observed, click the **status summary** and make a note of the port number and therefore server. Recall port 1990 is server1 and 1991 is server2.



__ Close both Health Center windows, disconnecting them from the JVMs.

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





_ Close any open files in Tail.

_____ Use Tail to monitor the SystemOut.log of the **deployment manager** and **either server1 or server2** (the one you have identified as having a memory leak). The files are located at:

C:\IBM\WebSphere\AppServer\profiles\Dmgr\logs\dmgr C:\IBM\WebSphere\AppServer\profiles\AppSrv01\logs\server1 C:\IBM\WebSphere\AppServer\profiles\AppSrv01\logs\server2

Т	ail for	Win3	2 - [Sys te	emOut.lo	og]								
<u>F</u> ile	<u>E</u> dit	⊻iew	<u>S</u> ettings	<u>P</u> lugins	Tools	<u>W</u> ind	ow <u>H</u> elp						
▶	٠												
					[1/14	ł/13	6:29:0	7:053	PST]	000000da	ActionServlet	Ι	org.apache.:
					[1/14	ł/13	6:29:4	0:928	PST]	000000da	servlet	Ι	com.ibm.ws.u
	ti M	orkspa	tes		[1/14	ł/13	6:29:4	1:131	PST]	000000da	TilesRequestP	Ι	org.apache.:
	÷ 🗄	Non-	Workspace	Files	[1/14	ł/13	6:29:4	5:194	PST]	000000ba	WsmmConfigFac	Ι	ARFM5007I
		-13	:\IBM\Web	oSpher	11/14	¥/13	6:29:4	5:725	PST1		WsmmConfigFac	Ι	ARFM5007I
		11	:\IBM\Web	oSphere\/	AppServ	er\prof	iles\Dmgr\l	ogs\dmgr	System	Out.log	WsmmConfigFac	Ι	ARFM5007I
					[1/14	ł/13	6:29:4	5:912	PST]	dd00000bb	WsmmConfigFac	I	ARFM5007I
					[1/14	ł/13	6:29:4	6:678	PST]	000000ed	WsmmConfigFac	Ι	ARFM5007I
					[1/14	ł/13	6:29:4	6:678	PST]	000000eb	WsmmConfigFac	I	ARFM5007I

In the browser, return to the Accessories tab and click the **Wheelbarrow** product **one more time**. This should be sufficient to trigger the health policy actions as the JVM occupancy (after a garbage collection cycle) will have exceeded 85%.

In the tail window, monitor the **Dmgr log** and **wait for up to up to 60 seconds** until the logs similar to these are seen indicating that the Health Management controller has detected the condition as entered for the policy earlier and will be taking the appropriate actions:

JVMMemorySubS W WXDH3004W: The memory consumption limit specified by policy **plantsMemoryHealthPolicy** was exceeded by server server2 on node impact2013Node. The limit is 85 % and the current heap size is 90 % of the maximum of 131072 KB.

HeapDumpTask I WXDH1014I: Heap dumps for server impact2013Cell/impact2013Node/server2 are being issued automatically.

Note:

The health management controller is managed by the WebSphere High Availability Manager which means it could typically be running on any node agent, or deployment manager. For this lab, the health management controller has been configured to always start on the deployment manager, meaning it is easier to locate its logging output. For details of how to configure where such controllers start, see the references.



The logs indicate the JVM has occupied more than 85% of the available heap space and the health policy has been triggered. At this time, heap dumps are triggered and then the application server JVM is restarted.

When the heapdumps are generated, you may notice that Health Center pops up three dialog boxes to alert you. You can ignore these.

_____ In the tail window, monitor the SystemOut log for the server identified previously (i.e. server1 or server2).

Note:

If you timed it correctly, you will see the logs show the application server is busy restarting – you may have to wait a few moments until the restart process begins. When it does, continue with the steps to see what happens to further application requests from the browser client.

_____ In the browser, click a product – preferably not the wheelbarrow! You can expect to see the warning below:



Note:

The IBM HTTP Server plugin has detected that one of the application servers is unavailable, and has routed the request to the other server in the cluster.

In this lab environment, HTTP session objects are not shared between the JVMs as session persistence has not been configured, although this is possible using either memory to memory replication or database persistence.

For this reason, the plants application displays a warning that the session data is unavailable, and to continue the user should start a new session.



Start a new session by returning to the plants application homepage using browser favorites or typing URL http://localhost/PlantsByWebSphere



_____ Click a few links to satisfy yourself that you can continue to use the application even though one of the application servers is being restarted.

_____ At the admin console for the Application Server, refresh the status of the servers as shown below to ensure that the stopped server has restarted.

New	New Delete Templates Start Stop Restart ImmediateStop Terminate											
Select	Select Name 🔅 Node 🗘 🛛 Host Name 🗘 Version 🗘 Cluster Name 🗘 🗘 Status											
You c	an administ	er the following res	ources:									
	server1	impact2013Node	impact2013	ND 8.5.0.1	PlantsByWebSphereCluster	€						
	server2 impact2013Node impact2013 ND 8.5.0.1 PlantsByWebSphereCluster ♣											
Total	2		·									

Use Windows explorer to verify that the health policy actions triggered some heap dumps. By default they are written to the JVM's working directory as shown below:

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

🝂 C:\IBM\WebSphere\AppServer\profiles\AppSrv01	
<u>File E</u> dit <u>V</u> iew F <u>a</u> vorites <u>T</u> ools <u>H</u> elp	🥂 🖉
😋 Back 🝷 🕥 👻 🦻 Search 🏷 Folders	
Address C:\IBM\WebSphere\AppServer\profiles\AppSrv01	💌 🏓 Go
Folders	× Name
🛅 lafiles	heapdump.20130108.082850.1764.0001.phd
🕀 🛅 lib	eapdump.20130108.082854.1764.0002.phd
🛅 links	heapdump.20130108.082901.1764.0003.phd
🗉 🧰 logs	📃 orbtrc.14012013.0629.06.txt
🗉 🫅 optionalLibraries	📃 orbtrc.08012013.0817.55.txt
🗉 🧰 plugins	🧾 orbtrc.08012013.0639.50.txt
🖃 🫅 profiles	🧾 orbtrc.10012013.0843.38.txt
🖂 🗁 AppSrv01	🧾 orbtrc.10012013.0842.31.txt
- ~ .	E orbite 10012013 0321 37 Evit



In the next section, these heap dumps will be used to diagnose the memory leak. While they represent a Java heap that was only 85% occupied, this should be sufficient to identify the objects that have contributed to the memory leak.



Part 7: Using ISA and the Memory Analyzer to Analyze a Heapdump

Note:

IBM Memory Analyzer is a powerful and flexible tool for analyzing Java heap memory, using heap dumps or system dumps.

ISA 5.0 provides three ways to use this tool – HTML report, web interface and desktop tool. This lab will demonstrate all three.

You will use the heap dumps generated in the previous section. However, as every heap dump is different, you may see some slight variation from the screenshots in this lab document, e.g. exact number of bytes for the object size or number of objects in a data structure etc.

If you prefer, use the ready made heap dump in "E:<u>\Impact</u> lab files\heap dumps" – this files has previously been opened by Memory Analyzer which creates "index files". Using these files will slightly reduce the amount of time required to complete the lab – it's up to you.

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

🐸 IBM Support Assistant - Mozilla Firefox											
<u>File E</u> dit <u>V</u> iew History <u>B</u> ookmarks <u>T</u> ools	Help										
IBM Support Assistant	+										
Contemporary Conte						⊽ C" 🚼 - I	Google	۶ 🎤			
IBM Support Assistant						6	ot Feedbac	<mark>1.7</mark> Help <u>IBM</u> .			
Cases 🕨	Cases > Scan this Case Scan this Case Global Filter - Off 💌										
🗈 Files 🤷 Tools 📗 Reports 💼	Overview 🏭 Sympt	toms 🗐 Gl	obal Knowledg	e Base Matche	es 📳 Data	Collector					
ta Tree View 🛡 🗀						Search	File Conter	٩			
	Name Filter										
Navigator	Name	Symptoms	Knowledge Ba	First Timestamı	Last Timestam	Size	Туре	Modified (PST)			

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



	IBM Support Assistant	
Case Ma	anagement Delete	Cases
Casi 📤	Summary	bal Knowled
0000	Example Case	
0001	Testing	
		Knowledge B
Case II	D:	
Summa	ary:	
Descrip	iption:	

Complete the summary and description, click the green tick.

Cas	e ID:		[New]]				~	×
Sun	nmar	y:	PlantsByW	/ebSph	ere OO	M				
Des	cript	ion								
An	out	of	memory	cond	ition	for	the	plantsCluster		
										:

Shrink the cases dialog by clicking Cases.

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

_____Add one of the heap dumps to the case by dragging the file from Windows Explorer to the ISA case. The location of the heapdump file is:

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

1BM Support Assistan					Got Feedba	ck? He	^{elp} IBM.	profilerCyclesBetweenSaves=15 m
						-		code cache
Cases [0003] PlantsByWebSphereO	ОМ	•		Scan this Cas	e 🚱 🔻	Global Fil	ter - Off 🔽 🔻	me=MarkSweepCompact
	1	-						load protection is enabled.
🗈 Files 🤷 Tools 📗 Reports 🗐	Overview 👪 Symptom:	; 懫 Global	Knowledge Base Matches 🔡 Data (ollector				r cell impact2013Cell is { arfm
								Service has started chain WCIn
Tree View V CASE:0003/*				Searc	h File Conte	1	Q	Service has started chain Http
								Service has started chain Http
Add files Or drag files into browser to add	Name Filter 🔆		Filter Reset					Service has started chain WCIn
	Name	Symptome	Knowledge Ba, First Timestam, Last Times	ami Size	Type	Modified	(PST)	Service has started chain SOAP
Navigator	Traine	o y nipromo	The mode by That The stand East These	and bize	1300	modified	((0))	Service has started chain SOAP
🗀 0003 🛷								Service has started chain SOAP
								is starting the Schedulers.
	beandump 20130	117 005630						has completed starting the Sch
		11/1000000111						successfully
	20	Et\Impact lab	files\bean dumps					
		lo Edit Viou						
	0	Back v 🔿 v	🕂 😰 Search 🍋 Folders	× 🖌 📖 -				
	Ad	dress 🛅 C:\In	npact lab files\beap dumps					
	Fr	lders	**-			×	Name 🔺	Size Type
			🛱 lafiles			- 1	🖬 heapdump.2	20130117.005656 13,625 KB PHD File
			🗉 🧰 lib				heapdump.2	20130117.005652 13,612 KB PHD File
			inks				🖬 heapdump.2 🖬 heapdump.2	20130117.005649 13,612 KB PHD File 20130117.005643 13,759 KB PHD File
			± 🔲 logs			Ī	incapdump.2	20130117.005637 14,053 KB PHD File
			E C plugins				heapdump.2	20130117.005630 14,328 KB PHD File
Build ID: 5.0.0.0_Beta2_20121016-1409			🗉 🚞 profiles					
	_							
IBM Support Assi	stant	🔐 Su	crossfully added the fr	llowing (click thi	ie notif	fication	to

IBM Support Assistant	Successfully added the following (click this notification to hide):	GU
[0003] PlantsByWebSphereOOM	heapdump.20130117.005630.5356.0001.phd	Scan this Case

_ Cick the yellow box to dismiss the message



To make an initial assessment of a heap dump, it can be convenient to run a report on the ISA server. This avoids the need to download the heap dump or system dump to a local workstation. In the next steps you will generate a HTML leak suspects report.

Ensure the relevant case is selected, then choose **Memory Analyzer Report** from the tools tab. Click **Launch**.

Cases [0003] PlantsByWebSphere OOM	
Files Tools Reports Overview	/ 🚦 Symptoms 🖄 Global Knowledge Base Matches
Enter keyword <u>Filter</u> <u>Reset</u>	Memory Analyzer [Report]
Sort By: Tag: All Tags	Launch (1) Tool Help
Garbage Collection and Memory Visualizer (GCMV) [Desktop]	
Garbage Collection and Memory	IBM Monitoring and Diagnostic Tools for Java
Thealth Center	Memory Analyzer is a feature-rich Java hea
🍟 HeapAnalyzer [Desktop] 🛛 😚 🗊	This tool is provided in three versions:
T Memory Analyzer [Desktop]	 as a report generating version that re
👕 Memory Analyzer [Report] 🛛 🗸 🛄	 as an interactive GUI version running
👕 Memory Analyzer Web Edition [Web] 🚱 💿	• an interac

IBM Software Accelerated Value Program



х **Problem Analysis** Run Memory Analyzer [Report] (Version 1.2.0.201208221220) Input Files and Folders * Browse Parameters Value Parameter Description Generate Leak Suspects Report1 Leak Suspects Generate Top Components V Report2 Top Components report 3 e.g. Report3 org.eclipse.mat.api:overview Report4 Additional report/option Keep unreachable garbage keep_unreachable_objects objects in the heap Do not use any existing reparse indexes from a previous run Run as background task: 🗹 Submit Cancel

Click **Browse** and select the heap dump in this ISA case.



Pro	Problem Analysis					
	Select a File	х				
R 1 Ir	 © 0003 © heapdump.20130117.005630.5356.0001.phd 					
_						
Ρ						
F						
R						
R						
R						
R						
k						
re						
Ru	OK	el				

_ Click OK (**but don't press Submit yet**)

Note:

The reports to run are already selected.

The leak suspects report is a key feature of Memory Analyzer. It has capabilities to look for probable memory leak suspects including large objects or collections of objects that contribute significantly to the Java heap usage, and displays this information in the form of a pie chart.

The top components report outlines the top memory consumers, i.e. the object at the top of the 'tree' which is responsible for keeping other objects alive in the heap. Top consumers are not necessarily the cause of memory leaks, but could provide information on potential memory inefficiencies in the components.



_ The Top Components Report is not required for this lab, so de-select it.

Problem Analysis		Х				
Run Memory Analyzer [Report] (Version 1.2.0.201208221220) Input Files and Folders * /ISA5Beta2/ISA5/isa/cases/0003/heapdump.20130117.005630.5356.0001.phd Browse						
Parameters						
Parameter	Description	¥alue				
Report1	Leak Suspects	Generate Leak Suspects 💌				
Report2	Top Components	No Top Components report 💌				
Report3	report 3 e.g. org.eclipse.mat.api:overview					
Report4	Additional report/option					
keep_unreachable_objects Keep unreachable garbage						
reparse Do not use any existing indexes from a previous run						
Run as background task: 🗹]					
Submit Cancel						
Click Submit .						
IBM Support Assistant		The Memory Analyzer [Report] tool request has been submitted.				
Cases [0001] PlantsByWebSphere	▼	<u>So to output folder</u>				
🗈 Files 🧰 Tools 🗽 Reports 🗎 Overv	iew 🚦 Symptoms 懰 Global Knowledge Base M	atches 🔛 Data Collector				

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



Digital IBM Support Assistant		tot feedbook? Help 頂麗.
Cases [0003] PlantsByWebSphereOOM	v	Scan this Case Global Filter - Off 💌
🗈 Files 🛛 🖻 Tools 🚺 Reports 🗐 Overview 👪 Symp	otoms 🗐 Global Knowledge Base Matches	He Data Collector
Enter keyword <u>Filter</u> Reset		
Sort By: $\downarrow_{\mathbf{Z}}^{\mathbf{Z}} \downarrow_{0}^{0}$	0 5 🖗 📥	$\Delta \Leftrightarrow \Rightarrow$
	The Reports view provides acce Select a report from	ess to reports generated from tool executions. the list to view the report contents.

The Memory Analyzer tool running on the ISA server will parse the heap dump, and generate the requested reports. This takes a few moments so please be patient.

_____ Click **refresh** icon until the report is listed and the green tick appears. This can takes around three or four minutes so please be patient.

IBM Support Assistant						
Cases [0003] PlantsByWebSphereOOM						
👔 Files 🧰 Tools 📗 Reports	📋 Overview 👪 Sympt					
Enter keyword <u>Filter</u> R	eset					
Sort By: 📴 🔯						
Memory Analyzer [Report] heapdump.20130117.005630.5356.0001.phd	1/17/13 01:37:24					

Select the **report**, then click **the Leak Suspects** Link as shown below.

IBM Software Accelerated Value Program



🗈 Files 🙆 Tools 녪 Reports 🖹 Overview 🏭 Symp	ptoms Global Knowledge Base Matches 🔛 Data Collector
Enter keyword Filter Reset	Memory Analyzer [Report]
Sort By: 📴 🧑	0 🗗 🗟 📛
Memory Analyzer [Report] 1/17/13 01:37:24 heapdump.20130117.005630.5356.0001.phd 1/17/13 01:37:24	
	Dump file /ISA5Beta2/ISA5/isa/cases/0003/heapdump.20130117.005630.5356.0001.phd
	Options -output_folder="/DOCUME~1/ADMINI~1/LOCALS~1/Temp/1/com.ibm.java.diagno
	Memory Analyzer version: 1.2.0.201208221220
	Analysis completed: January 17 2013 01:37
	 Report: heapdump.20130117.005630.5356.0001 Leak Suspects Log file(standard out stream): t0117.013534.421583.log Log file(standard err stream): t0117.013534.421583-err.log





Examine the overview pie chart. It indicates that a problem suspect occupies around 40Mb of the total heap occupancy. The exact statistics in your heap dump may vary.



_____ Scroll down to examine the detailed description. It identifies an object called **ShoppingBean** which responsible for a large percentage of memory, and that a single **HashMap\$Entry[]** is involved.

▼ ⊗ Problem Suspect 1

The class "com.ibm.websphere.samples.pbw.war.ShoppingBean", loaded by "<system class loader>", occupies 49,004,464 (49.48%) bytes. The memory is accumulated in one instance of "java.util.HashMap\$Entry[]" loaded by "<system class loader>".

Keywords com.ibm.websphere.samples.pbw.war.ShoppingBean java.util.HashMap\$Entry[]

<u>Details »</u>



The object syntax "HashMap\$Entry[]" refers to a HashMap object that contains an array of type Entry. The dollar symbol indicates that Entry is an inner class of HashMap. This appears to be the JVM's internal representation of a data structure used in the Plants application code.

Click on the **Details** link to display the shortest paths to an accumulation point.

Note:

An accumulation point is simply a reference in the chain that is suddenly responsible for keeping lots of heap space alive. In this case, the single HashMap at the top of the table has been identified as the single object that is responsible for the large accumulation of further objects.

The shortest path to this accumulation point shows what is responsible for keeping that accumulation object alive. In this case an instance of the ShoppingBean class references the HashMap\$Entry[].

▼ Shortest Paths To the Accumulation Point

Class Name	Shallow Heap	Retained Heap
n java.util.HashMap\$Entry[131072] @ 0x67220b0	524,304	40,924,304
iava.util.HashMap @ 0x36d3fc0	48	40,924,368
	16	40,924,384
Lass com.ibm.websphere.samples.pbw.war.ShoppingBean @ 0x29484e8 System Class	80	40,924,464

Scroll down to view the accumulated objects.

Accumulated Objects

Class Name	Shallow Heap	Retained Heap	Percentage
Cass com.ibm.websphere.samples.pbw.war.ShoppingBean @ 0x29484e8	80	40,924,464	44.57%
iDiava.util.HashSet @ 0x36bcba8	16	40,924,384	44.57%
iDiava.util.HashMap @ 0x36d3fc0	48	40,924,368	44.57%
i miava.util.HashMap\$Entry[131072] @ 0x67220b0	524,304	40,924,304	44.57%
<mark>jiava.util.HashMap\$Entry @ 0x322c1b0</mark>	24	5,656	0.01%
Diava.util.HashMap\$Entry @ 0x3abbbf8	24	5,656	0.01%
Diava.util.HashMap\$Entry @ 0x2e2af18	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x3158810	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x32f4410	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x34040f0	24	4,848	0.01%
<mark>jiava.util.HashMap\$Entry @ 0x3932fb0</mark>	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x3d46a28	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x3e91c08	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x3f6c288	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x3ff6d60	24	4,848	0.01%
Diava.util.HashMap\$Entry @ 0x420c588	24	4,848	0.01%

Note:

This view shows the objects referred to by the HashMap\$Entry[] accumulation point. You will see the HashMap\$Entry[] contains entries that total approximately 40Mb or 44% of the occupied heap space. Each object is around 4Mb or more in size. Only the first 20 entries from the HashMap\$Entry[] are shown in this report.

Shallow heap refers to the size of an individual object in isolation, and retained heap includes all the objects that are referenced (and kept alive) by that object.

Scroll down to view the "Accumulated Object by Class" table.

Accumulated Objects by Class

Label	Number of Objects	Used Heap Size	Retained Heap Size
java.util.HashMap\$Entry First 10 of 37,170 objects	37,170	892,080	40,400,000



This shows the total number of objects referred to by the HashMap\$Entry[] accumulation point, in this case over 37,000 objects with a retained size of over 40Mb. The exact statistics in your heap dump may vary.

Diagnosing the memory leak would require further analysis of the HashMap\$Entry[], to determine exactly what is stored in that data structure. However, the Memory Analyzer Report is static and it is not possible to follow the links to examine the chain of references between the objects. To facilitate this, either the desktop or web version of Memory Analyzer could be used. In the remainder of this lab section the web version will be used. In the final section of the lab, the desktop edition will be used.

Cases [0003] PlantsByWebSphere	DOM	•			
Files 🌆 Tools 📗 Reports 🕻	🔋 Overview 🕌	Symptoms	劕 Global	Knowledge Base Matches	Data Collector
Enter keyword Filter Reset Memory Analyzer Web Edition [Web] Versic					
Sort By: 📴 Tag: All Tags	▼	🚺 La	unch	(i) Tool Help	
Garbage Collection and Memory Visualizer (GCMV) [Desktop]	V]				
Garbage Collection and Memory Visualizer (GCMV) [Report]	🗸 🛄	IBM I	IBM Monitoring and Diagnostic Tools for Java(TM) - Memory Anal, Memory Analyzer is a feature-rich Java heap analyzer that helps		
👕 Health Center	√ □	Memo			
👕 HeapAnalyzer (Desktop)	69 🗊	This a	This tool is provided in three versions:		
👕 Memory Analyzer [Desktop]	√ □		 as a report generating version that reads the dump, build: 		
Memory Analyzer [Report]	<u>√ l</u>		 as an interactive GUI version running on the desktop 		
Themory Analyzer Web Edition [Web]	66 💽	(ma	ore)		

_ Click the ISA Tools tab. Select the Memory Analyzer Web Edition. Click Launch.


Memory Ana	lyzer Web Edition [Web] Version 1.1.0.201203301819	
🜔 Launch	1) Tool Help	
IBM Monitoring an	Problem Analysis X	
Memory Analyzer	Run Memory Analyzer Web Edition [Web] (Version	
This tool is provid	1.1.0.201203301819)	
 as a report as an inter 	Input Files and Folders *	
(more)	Browse	
Tags: 💽 Web-b	ar	ea;
Restrictions:	Submit Cancel	
	 Ine tool supports only one user session per prowser. Some links in the exported or downloaded reports will not work when reports are opened outside the to 	ol.

____ On the "Problem Analysis" window that pops up as shown above, Click **Browse** and select the same heap dump in the ISA case as shown below. Then, Click **OK**.

Select a File	х
O O	
heapdump.20130117.005630.5356.0001.phd	

____ Click **Submit**. The browser window may remain blank for a few moments.

Problem Analysis	Х
Run Memory Analyzer Web Edition [Web] (Vers 1.1.0.201203301819) Input Files and Folders *	ion
/ISA5Beta2/ISA5/isa/cases/0003/heapdump.20130117.005630.5356.0001.phd	Browse
Subm	it Cancel



🥹 Eclipse Memory Analyzer - Mozilla Firefox					
Ele Edit View Higtory Bookmarks Tools Help					
Plants By WebSphere Shopping ×	VebSphere Integrated Solutions Console 🛛 💭 0003 - IBM Support Assistant 5.0 🛛 🖄 💭 Eclipse Memory Analyzer 🛛 🗙 🕂				
(impact2013:10912/mat/mat?startup=org.eclip	se.mat.ui.rcp.application&mainInputFile=/ISA58eta2/ISA5/isa/cases/0003/heapdump.20130117.005630.5356.0001.phd&outputFolder=/ISA58eta2/ISA5/isa/cases/0003/heapdump.20 🏫 🛡 🕻				
IBM Monitoring and Diagnostic Tools for Jav	a - Memory Analyzer Web Edition - 1.1.0.201203301819				
Edit Window Help					
🖾 Inspector 🛛 🤹 🗖 🗖	🖁 neepolong aatalaat aaddaa aasaa aata goo 🗴				
Statics Attributes » * Type Name Value	Opening Parsing heap dump from '\ISA5Beta2\ISA5\isa\cases\0003\heapdump.20130117.005630.5356.0001.phd' Cancel Parsing heap dump from '\ISA5Beta2\ISA5\isa\cases\0003\heapdump from '\ISA5Beta2\ISA5\Isa\ca				

Wait for the heap dump to be parsed, this time the operation will be quicker as Memory Analyzer 'index files' already exist on the filesystem from the previous analysis.



Getting Started Wizard	o x
Getting Started	
Choose one of the common reports below. Press Escape to close this dialog.	
C Leak Suspects Report	
Automatically check the heap dump for leak suspects. Report what objects are kept alive and why they are not garbage collected.	
🔿 Component Report	
Analyze a set of objects for suspected memory issues: duplicate strings, empty collections, finalizer, weak references, etc.	
O Re-open previously run reports	
Existing reports are stored in ZIP files next to the heap dump.	
✓ Show this dialog when opening a heap dump.	
< Back Next > Finish Ca	ncel

_____ Select the **Leak Suspects** Report to generate the same report you generated previously with the Memory Analyzer Report tool. Click **Finish**.





Click the details link and scroll down to the "Accumulated Objects" table.

_____ To determine what is stored in each of these HashMap entries, click the **accumulation point link**, i.e. the HashMapEntry **with the array icon**

🔻 Accumulated Objects

Class Name	Shallow Heap	Retained Heap	Percentage
Cass com.ibm.websphere.samples.pbw.war.ShoppingBean @ 0x2987be0	80	49,004,464	49.48%
الس <mark>ارة (java.util.HashSet @ 0x3968288</mark>	16	49,004,384	49.48%
i <mark>Niava.util.HashMap @ 0x3af9e20</mark>	48	49,004,368	49.48%
i <mark>[]</mark> iava.util.HashMap\$Entry[131072] @ 0x3872a60]	524,304	49,004,304	49.48%
Diava.util.HashMap\$Entry @ 0x3f19510	24	6,464	0.01%
Diava.util.HashMap\$Entry @ 0x317d538	24	5,656	0.01%
Diava.util.HashMap\$Entry @ 0x3a8f400	24	5,656	0.01%



Note:

The left Inspector panel show details for the HashMap, including its attributes as shown below.

🔍 Insp	ector 🛛		€ 5 °	- 🗆
@ 0x38	372a60			
<u> 1</u> Hasł	nMap\$Entry[]		
🖶 java	.util			
🔎 class	s java.util.Ha	shMap\$Entry[] @ 0x6a1948		=
🕒 java	.lang.Object			
🔏 java	lang.ClassL	oader @ 0x0		
1 524,	304 (shallow	/ size)		
‡ <mark>1</mark> 49,0	04,304 (reta	ined size)		-
Statics	Attributes	Class Hierarchy Value		*
Туре	Name	Value		
ref	[0]	java.util.HashMap\$Entry @	0x4fd	f9 📥
ref	[1]	java.util.HashMap\$Entry @	0x674	47
ref	[2]	java.util.HashMap\$Entry @	0x674	48
ref	[3]	java.util.HashMap\$Entry @	0x532	28
ref	[4]	java.util.HashMap\$Entry @	Ox410	:15
ref	[5]	java.util.HashMap\$Entry @	0x674	48

_____ Right click the object reference in the Inspector tab, and select **List Objects->with outgoing references** to display all references from the selected HashMap object.

Note:

There are other more advanced ways to view outgoing references such as "Immediate Dominators" which can help simplify the chain of references by showing only the most significant references, i.e. only those that keep an object alive in the heap.



Expand the top level HashMap object to show the thousands of entries that compose this HashMap.



8	heapdump.20130117.005630.5356.0001.phd 🖾		
i	║ 🗄 💀 📂 र 🍪 र 🔍 🔤 र 🖄 र 🖣		
i	Overview 😂 default_report org.eclipse.mat.api:	suspects 📘 li	st_objects [context
Cla	ess Name	Shallow Heap	Retained Heap
	🕨 🗋 [6138] java.util.HashMap\$Entry @ 0x2d5f06	24	808
	🕨 🗋 [34127] java.util.HashMap\$Entry @ 0x2d5f3	24	1,616
	🕨 🗋 [1488] java.util.HashMap\$Entry @ 0x2d5f6b	24	808
	🕨 🗋 [20006] java.util.HashMap\$Entry @ 0x2d5f6	24	808
	[10747] java.util.HashMap\$Entry @ 0x2d5fc	24	808
	🕨 🗋 [1489] java.util.HashMap\$Entry @ 0x2d600	24	1,616
	🕨 🗋 [6140] java.util.HashMap\$Entry @ 0x2d603	24	808
	[20009] java.util.HashMap\$Entry @ 0x2d60	£ 24	808
	🕨 🗋 [29462] java.util.HashMap\$Entry @ 0x2d60	24	808
	[20010] java.util.HashMap\$Entry @ 0x2d60	24	1,616
	🕨 🗋 [6142] java.util.HashMap\$Entry @ 0x2d613	24	808
	🕨 🗋 [20011] java.util.HashMap\$Entry @ 0x2d61	£ 24	1,616
	🕨 🗋 [29464] java.util.HashMap\$Entry @ 0x2d61	24	808
	I [10750] java.util.HashMap\$Entry @ 0x2d61	24	808
	🕨 🗋 [20014] java.util.HashMap\$Entry @ 0x2d62	24	808
	🕨 🗋 [6143] java.util.HashMap\$Entry @ 0x2d626	24	808
	🕨 🗋 [15340] java.util.HashMap\$Entry @ 0x2d62	24	808
	🕨 🗋 [24646] java.util.HashMap\$Entry @ 0x2d62	24	808
	🕨 🗋 [24649] java.util.HashMap\$Entry @ 0x2d62	24	1,616
	∑. Total: 25 of 37,171 entries		

_____ Scroll down and display some further entries by right clicking the "Total" and selecting "Next 25". The "Expand All" option is likely to take some time so **avoid** clicking that.

🕨 🗋 [24646] java.util.HashMap\$Entry @ 0x2d62c			
🕨 🗋 [24649] java.util.HashMap\$Entry @ 0x2d62f			
Total: 25 of 37,171 entries			
🖫 Navigation History 🛛	+ Next 25		
i OverviewPane	+ Custom expand + Expand All		



Expand one of the HashMap entries. Notice that the HashMap entry refers to a ShoppingContainer class in the plants sample (you will need to adjust the column widths).

🔺 🗋 [20011] java.util.HashMap\$Entry @ 0x2d61640	24	1,616
> 起 <class> class java.util.HashMap\$Entry @ 0x6a1920 System Class</class>	80	80
> 🗋 com.ibm.websphere.samples.pbw.war.ShoppingBear <mark>\$ShoppingContainer @ 0x2d61658</mark>	16	784
> 🗋 java.util.HashSet @ 0x36bcba8	16	40,924,384
🕨 🗋 java.util.HashMap\$Entry @ 0x6720dd8	24	808
∑ Total: 4 entries		

Note:

The class name "ShoppingBean\$ShoppingContainer" means the "ShoppingContainer" class is an inner class of "ShoppingBean".

This ShoppingContainer object is responsible for keeping 784 bytes alive in the heap. The HashMapEntry actually keeps 1616 bytes alive in total, including the ShoppingContainer. All the other HashMapEntry objects in the list follow an identical pattern. **The exact statistics in your heap dump may vary.**

_____ Try expanding another HashMap entry. Their retained heap may vary, but they always contain a ShoppingContainer of the same size.

🔺 🗋 [1488] java.util.HashMap\$Entry @ 0x2d5f6b0	24	808
) 🔂 <class> class java.util.HashMap\$Entry @ 0x6a1920 System Class</class>	80	80
🕨 🗋 com.ibm.websphere.samples.pbw.war.ShoppingBean\$ShoppingContainer @ 0x2d5f6c8	16	784
> 🗋 java.util.HashSet @ 0x36bcba8	16	40,924,384
Σ Total: 3 entries		

Note:

You have established that objects from the plants sample seem to be involved with the memory leak - a ShoppingBean is referring to a large HashMap that contains many instances of ShoppingContainer. Let's look at the overall footprint of the entire plants sample.

____ Click on the 🛄 icon on the Memory Analyzer toolbar to open a histogram.



😫 heapdump.20130117.005630.5356.0001.phd 🙁						
i III 🖫 💀 ⊯ ▼ 🗞 ▼ 🔍 🍃 ▼ 📾 ▼ 🖾 ▼ 😤						
🚺 Overview 😂 default_report org.eclipse.mat.api:suspects 🗋 list_objects [context] 🔢 Histogram 🗙						
Class Name	Objects 🔻 9	Shallow Heap	etained Heap			
🛟 <regex></regex>	<numeric></numeric>	<numeric></numeric>	<numeric></numeric>			
🕑 java.lang.String	188,812	4,531,488	>= 19,011,68			
	166,359	17,639,112	>= 17,639,23			
🕑 java.util.HashMap\$Entry	117,958	2,830,992	>= 46,496,44			
O byte[]	54,042	42,619,192	>= 42,619,28			
🕒 com.ibm.websphere.samples.pb	50,000	800,000	>= 39,200,08			
o bytejj o com.ibm.websphere.samples.pbv	54,042	42,619,192 800,000	>= 42,619,28			

Type "**com.ibm.websphere.samples.pbw.***" in the "Regex" filter box and press the Enter key. The Regex filter box is the first line of the table, starting with the icon $\frac{24}{24} < \text{Regex} > 1$.

😫 heapdump.20130117.005630.5356.0001.phd 🖾					
i III №					
i Overview 器 default_report org.e	👔 Overview 竁 default_report org.eclipse.mat.api:suspects 🚺 list_objects [context] 🔢 Histogram 🗙				🛯 Histogram 🗙
Class Name	Objects	hallow Heap 🔻	etained Heap		
;m.ibm.websphere.samples.pbw.*	<numeric></numeric>	<numeric></numeric>	<numeric></numeric>		
🕒 byte[]	54,042	42,619,192	>= 42,619,28		

-

Heapdump.20130117.005630.5356.0001.phd X				
i III 🖫 👊 🔎 🕶 🍔 🕶 🔍 🎦 🕶 📼 🕶 🖾 🕶 😤				
i Overview 😂 default_report org.eclipse.mat.api:suspects	📘 🗋 list_object	s [context] 📗	Histogram 🗙	
Class Name	Objects 🔻	Shallow Heap	Retained Heap	
com.ibm.websphere.samples.pbw.*	<numeric></numeric>	<numeric></numeric>	<numeric></numeric>	
😉 com.ibm.websphere.samples.pbw.war.ShoppingBean\$Sh	50,000	800,000	>= 39,200,080	
⊙ com.ibm.websphere.samples.pbw.jpa.Inventory	13	1,040	>= 862,936	
😉 com.ibm.websphere.samples.pbw.war.ProductBean	12	192	>= 863,016	
⊖ com.ibm.websphere.samples.pbw.ejb.CatalogMgr	5	80	>= 160	
⊙ com.ibm.websphere.samples.pbw.war.ShoppingBean	1	32	>= 41,787,872	
	1	40	>= 848	
€ com.ibm.websphere.samples.pbw.ejb.EJSLocalNSLCatalo	1	24	>= 328	
	1	24	>= 504	
😉 com.ibm.websphere.samples.pbw.jpa.OrderItem	1	64	>= 176	
😉 com.ibm.websphere.samples.pbw.war.ImageServlet	1	16	>= 96	
⊖ com.ibm.websphere.samples.pbw.jpa.Order	1	120	>= 480	
⊖ com.ibm.websphere.samples.pbw.jpa.Supplier	1	48	>= 152	
⊖ com.ibm.websphere.samples.pbw.ejb.ShoppingCartBean	1	16	>= 216	
	1	24	>= 424	
G com.ibm.websphere.samples.pbw.jpa.OrderItem\$PK	1	24	>= 128	

Click the Retained Heap column heading to order the table by the largest retained heap.

😫 heapdump.20130201.085755.9312.0001.phd 🖾					
i III 🖫 💀 ⊯ 🖛 😪 ▾ I 🔍 🎦 ▾ 📾 ▾ 🛃 ▾ 😤					
🚺 Overview 🔀 default_report org.eclipse.mat.api:suspects 🗋 li	i Overview 🗞 default_report org.eclipse.mat.api:suspects 🗋 list_objects [context] III Histogram 🗙				
Class Name	Objects (Shallow Heap	tained Heap 🔻		
🔆 .*com.ibm.websphere.samples.pbw.*	<numeric></numeric>	<numeric></numeric>	<numeric></numeric>		
om.ibm.websphere.samples.pbw.war.ShoppingBean	1	32	>= 49,867,87		
	60,000	960,000	>= 47,040,08		
• com.ibm.websphere.samples.pbw.war.ProductBean	12	192	>= 863,016		
om.ibm.websphere.samples.pbw.jpa.Inventory	13	1,040	>= 862,936		
• com.ibm.websphere.samples.pbw.utils.Util	0	0	>= 4,016		
ocom.ibm.websphere.samples.pbw.war.ShoppingBean_\$\$_java:	1	40	>= 848		



Note:

In this heap dump there are 60,000 instances of ShoppingContainer with a shallow heap of 9Mb and a retained heap of 47Mb. The cause of this memory leak is becoming increasingly clear, and it seems only the ShoppingContainer and ShoppingBean classes are involved. **The exact statistics in your heap dump may vary.**

Optional Steps:

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



____ Click "Edit->Find" and search for "ShoppingContainer" to locate the definition of the inner class.

___ Click "Edit->Find" and search for method "deliberateMemoryLeak", click "Find Again".

Note:

You will find method "deliberateMemoryLeak" adds 10000 instances of the ShoppingContainer inner class to a **static** HashMap every time the wheelbarrow image is clicked. The constructor of the ShoppingContainer class fills a pointless array of size 750 bytes. This is the source of the memory leak.



```
private void deliberateMemoryLeak() {
   // -----
   // User clicked on the wheelbarrow, let's fill it with a
   // memory leak.
   // ------
   System.out.println("==> STARTING MEMORY LEAK");
   int LEAK_SIZE = 10000;
   for (int i = 0; i < LEAK_SIZE; i++) {</pre>
      leakObject.add(new ShoppingContainer());
   }
   System.out
   .println("==> Added "
         + LEAK SIZE
         + " objects to memory leak. The leak now contains "
         + leakObject.size() + " objects.");
   System.out.println("==> ENDING MEMORY LEAK");
}
   class ShoppingContainer {
   private byte[] array;
   ShoppingContainer() {
     array = new byte[750];
     Arrays.fill(array, (byte) 65);
     }
   }
```

Part 8: (Optional) Using the IBM Extensions to Memory Analyzer for Further Memory Analysis

This part of the lab as the title indicates is optional. However, this is available for you to do as time permits.

Note:

"IBM Extensions for Memory Analyzer" have been installed into ISA. These were initially released by IBM via the IBM Alphaworks site. They are now integrated with ISA 5 and have become a very useful extension to the Memory Analyzer tool. They provide IBM product specific analysis of heap dumps by applying knowledge of the internal data structures of the IBM products into useful reports. For example, there is a WebSphere Application Server report to view the size and other details of the WebSphere objects in the heap dump that hold the HTTP sessions. This enables the Memory Analyzer tool to be used for more than just diagnosing memory leaks. Most of the IBM extensions require the additional data only available in a full system dump, as opposed to a Java heap dump.

A heap dump is a dump of the state of the Java heap memory. This is very useful for analyzing the use of memory by a Java application, and therefore very useful for diagnosing out of memory issues. A system dump consists of all the memory that is being used by the JVM process; this includes the Java heap, along with all JVM and user libraries (native memory). System dumps also contain field values of Java objects. System dumps are often used in the diagnosis of other Java problem scenarios such as crashes, as they show detailed information about the state of the JVM. Because a system dump contains all of the memory allocated by the JVM process, system dump files can be very large.

In previous versions of the IBM SDK shipped with WebSphere Application Server, it was necessary to postprocess 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 **SDK 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.

To save time, for this section of the lab, a system dump is provided in "E:\Impact Lab Files\IBM Extensions". The dump was generated as a result of an **out of memory** condition.

_____ Close the browser tab showing the Memory Analyzer web edition, where you analyzed the IBM heap dump.

_____ In the ISA tab, click Tools, then select Memory Analyzer (Desktop)

IBM Software Accelerated Value Program

🕀 WebSphere Integrated Solutions Console 🛛 🗌 0003 - IBM Support Assistant 5.0 🛛 🗙 门 Plants By WebSphere Shopping 🛛 🗙					
localhost:10911/isa5/index.html#view=	tools&id=0003	&toolId=com.ibm.ja	va.diagnostics	.memory.analyzer.jws	
IBM Support Assistan	IBM Support Assistant				
Cases 🕨 [0003] PlantsByWebSphere OOM					
🗈 Files 🤷 Tools 🗽 Reports 🕻) Overview	📳 Symptoms	🗐 Globa	l Knowledge Base Matches	In Data Collector
Enter keyword <u>Filter</u> Res	<u>et</u>	Me	mory Ar	nalyzer [Desktop]	Version 1.2.0.2012
Sort By: 📮 Tag: All Tags	-	O L	aunch	(i) Tool Help	
Garbage Collection and Memory ✓ □ Visualizer (GCMV) [Desktop] ✓ □ Garbage Collection and Memory ✓ □		IBM	IBM Monitoring and Diagnostic Tools for Java(TM) - Memory		
 Visualizer (GCMV) [Report] Health Center HeapAnalyzer [Desktop] 	 ✓ □ ≪ □ 	Men help	Memory Analyzer is a feature-rich Java heap analyzer that helps you find memory leaks and reduce memory		
T Memory Analyzer [Desktop]	V D	con.	sumption.		
T Memory Analyzer [Report]	🗸 🛄	This	tool is prov	ided in three versions:	
T Memory Analyzer Web Edition [Web]	69 💿		 as a report generating version that reads the dump. 		
Thread and Monitor Dump Analyzer (TMDA) [Desktop]	69 🗊		 builds some index files and generates some zipped HTML reports. as an interactive GUI version running on the desktop an interac 		
Thread and Monitor Dump Analyzer (TMDA) [Report]	69 <u>III</u>				
WebSphere Application Server Configuration Visualizer	69 <u>III</u>	(1	(more)		

Click Launch, and then at the Run Memory Analyzer[Desktop] version pop window as shown below Click Submit.

Impact2013

Business. In Motion.



Res	et 👻	Memory Analyzer [Desktop] Version 1.2.0.201208221220 FP1 Launch I Tool Help	
	V I	Problem Analysis ×	
	✓ □ ≪ □	Run Memory Analyzer [Desktop] (Version 1.2.0.201208221220 FP1)	
 This tool is a desktop application. It will be launched using Java Web Start workstation. Using the tool with files associated with ticket will require that from the workstation. If a file is located on a remote server, you can down system location or access the file through a shared storage area. Any exis accessed by the tool. 		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.	
)IT	*3 🗖	In some cases, analysis of files on your workstation can noticeably degrade performance of other applications running on your workstation.	No Sc
	°\$ 📊	Click 'Submit' below to begin.	3 Probler
		a jextracted Java 1.4.2 dump file (*.sdff): (e.g. core.20080514.143839.5904.txt.sdff)	as input 1

Wait a few minutes for the Memory Analyzer Desktop tool to launch. Once launched you will see the following window. (If the window is not visible, select the **IBM Monitoring and Diagnostic Tool** on the taskbar at the bottom of the windows desktop to bring up the tool.)

IBM Monitoring and Diag		
🕑 IBM Monitoring and Diagnostic Tools for Jay	ra - Memory Analyzer	×
<u>File Help Views</u>		
📑 🅒 Memory Analyzer		
🖸 Inspector 🛛 🔄 🗖		
Statics Attributes Class Hierarchy Value Type Name Value		
	🖉 Notes 🛛 💱 Navigation History	
	39M of 53M 🛅	



Click File->Open heap dump.

=ł	🕒 IBM Monitoring and Diag	nostic Tools for Java - Memory Analy:
s	<u>File</u> <u>H</u> elp <u>V</u> iews	
s	E <u>x</u> it <u>P</u> references	
5	膋 Open Heap Dump	

Navigate to the system dump in E:\Impact Lab Files\IBM Extensions\.

Open Snapshot			? ×
Look jn:	🗁 IBM Extensions	- 😮 🕫 📂 🖽 -	
My Recent Documents Desktop My Documents My Computer	My Recent Documents Desktop My Documents My Computer 3/5 Floppy (A:) CD Drive (D:) Impact 2013 Lab Files (E:) Impact lab files My Network Places	ion_Anal.zip	
My Network	File <u>n</u> ame:	▼ <u>O</u> pen	٦
Places	Files of type: All Known Formats	Cance	

_____ Select the dump file labeled *core.20130303.050212.7744.0001.dmp* and click **Open**. Please be patient, parsing a system dump can take some time. Approximately 3-5 minutes may pass with little nothing reported by the progress bar.



IBM Software Accelerated Value Program

🕒 IBM Monitoring and Diagnostic Tools for Jav	a - Memory Analyzer	_ 🗆 X
Eile Help Views		
Memory Analyzer	(@	
🕰 Inspector 🗙 🦉 🗖	Score.20130303.050212.7744.0001.dmp	
	Opening	
	Parsing heap dump from 'E:\Impact lab files\IBM Extensions\core.20130303.050212.7744.0001.dmp'	
Statics Attributes Class Hierarchy Value	🔮 Parsing heap dump from 'E:\Impact lab files\IBM Extensions\co 📃 🗖 🗙	
Type Name Value	Processing DTFJ Image from file E:core.20130303.050212.7744.0001.dmp	
	Finding monitor objects missing from bean	
	Always run in background	
	Run in Background Cancel Details >>	
	Protes 23 G- Navigation History	
	ar an	_
	220M of 306M 👔 Parsing heap dump f1.dmp': (1%)	

ST Start				
My Network Places	File <u>n</u> ame:	core.20130303.050212.7744.0001.dmp	•	<u>O</u> pen
	Files of <u>type</u> :	All Known Formats	-	Cancel

On the "Getting Started Wizard" box, click **Cancel** to avoid generating the standard reports.



Next, Click the reports icon and select IBM Extensions->WebSphere Application Server->Class Loaders->Path to Leaking Application Class loader as shown below.

🥞 core.20130303.050212.7744.0001.dmp 🛛	
i 💵 🖳 🚾 🖌 🚝 🕶 🖓 🗸	
I Overview ☆ List objects Show objects by class Details Size: 133.7 MB Classes: % Merce Shortest Paths to GC Roots	achable Objects Histogram
Eclipse	
Biggest Objects by I IBM Extensions Java Basics Java Collections	CICS Transaction Gateway Java SE Runtime Utilities
1.7 MB Leak Identification	WebSphere Application Server Class Loaders Application ClassLoader Leaks (deprecated) Database (JDBC) Path to Leaking Application Class Loaders Dynacache
Search Queries	HA Manager HTTP Sessions HUng Threads HarmManager

On the "Path to leaking Application Class Loaders" window, expand the classloader below by clicking on the "+" as shown below.

😫 core.20130303.050212.7744.0001.dmp 🕺			
i III 🖫 💀 📈 + 🗞 + 🔍 🖾 + 👌			
👔 Overview 🔀 Path to Leaking Application Class Loaders 🛛			
Class Name	Shallow Heap	Retained 💌	
🕂 <regex></regex>	<numeric></numeric>	<numeric></numeric>	
E Com.ibm.ws.classloader.CompoundClassLoader @ 0x2b9cc38	160	4,024	

Left click any of the class loader identified as leaking and select List Objects->with outgoing references



					ttributes
😂 core.20130303.050212.7744.0001.dmp 🖾					
i III 🖫 💀 🔚 • 🍪 • 🔍 🖾 • 🖾 • 😃					
👔 Overview 😂 Path to Leaking Application Class Loaders 🛛					
Class Name		Shallow Heap	Retained 🔻	▲	
♣ <regex></regex>		<numeric></numeric>	<numeric></numeric>		
🖃 🔞 com.ibm.ws.classloader.CompoundClassLoader @ 0x2b9cc38		160	4,024		
🛨 🚺 (gm. ibm. wc. classicalder. CompoundClassicalder. 🖲 0v2b70		160	27 704		
🛨 🔂 😋 List objects	•	🄰 with outg	oing references		
\pm $\overline{\mathbf{a}}$ \mathbf{c} Show objects by class	•	📄 with incon	ning references		
🕀 🚹 🛛 🔁 Path To GC Roots	•	160	27,704		
🕀 🚺 🕫 🏪 Merge Shortest Paths to GC Roots	►	160	27,704		
	►	160	27,704		
🕀 🚺 🕬 Java Basics	•	160	27,704		I
🗄 🔂 😋 Java Collections	•	160	27,704		

Expand the class loader references by clicking on the "+" as shown below.

🥞 core.20130303.050212.7744.0001.dmp 🔀			
i III 🗄 💀 🕅 + 🍪 + 🔍 🛛 🕶 🖓 🖬			
👔 Overview 😂 Path to Leaking Application Class Loaders 🍃 with outgoing r	references 🖾 📃		
Class Name	Shallow Heap	Retained Heap	
🔁 <regex></regex>	<numeric></numeric>	<numeric></numeric>	
王 🙆 com.ibm.ws.classloader.CompoundClassLoader @ 0x2b79780	160	27,704	

_____ The resulting objects loaded by this class loader relates to the plants sample class (com.ibm.websphere.samples.*) which gives a clear indication of the application that is leaking memory as shown below.

Gere.20130303.050212.7744.0001.dmp ☎			- 0
i III 🗄 💀 🔎 - 😪 - 🔍 🖾 - 🕼			
👔 Overview 🔀 Path to Leaking Application Class Loaders 🚺 with outgoing refer	rences 🔀		
Class Name	Shallow Heap	Retained Heap	▲
😑 👩 com.ibm.ws.classloader.CompoundClassLoader @ 0x2b79780	160	27,704	
🗄 💽 < class > class com.ibm.ws.classloader.CompoundClassLoader @ 0x12ec	e 19,097	19,793	
🛨 💼 class com.ibm.websphere.samples.pbw.jpa.OrderItem @ 0x26251e0	414	430	
🛨 👩 class com.ibm.websphere.samples.pbw.jpa.OrderItem\$PK @ 0x2629a30	299	315	
🕀 💽 class com.ibm.websphere.samples.pbw.jpa.Inventory @ 0x2629b80	531	547	
王 💽 class com.ibm.websphere.samples.pbw.jpa.Order @ 0x2629be8	689	705	
표 👩 class com.ibm.websphere.samples.pbw.jpa.BackOrder @ 0x262c128	332	348	
王 👩 class com.ibm.websphere.samples.pbw.jpa.Customer @ 0x262ce50	312	328	
王 🖻 class com.ibm.websphere.samples.pbw.jpa.Supplier @ 0x2630dd0	249	265	
🛨 🗋 annotationCache java.util.Hashtable @ 0x2998538	48	6,208	
🛨 🗋 packages java.util.Hashtable @ 0x2998568	48	128	
표 🗋 packageSigners java.util.Hashtable @ 0x2998598	48	152	
🗉 🗋 methodCache java.util.Hashtable @ 0x29985c8	48	104	
🛨 🗋 fieldCache java.util.Hashtable @ 0x29985f8	48	8,264	
🗉 🗋 constructorCache java.util.Hashtable @ 0x2998628	48	104	
Image: Image: Section End of the section of the	16	16	
표 🗋 lazyInitLock java.lang.ClassLoader\$LazyInitLock @ 0x2b83798	16	16	
표 🗋 pds java.util.HashMap @ 0x2b837a8	48	144	
표 🗋 localClassPath java.lang.String @ 0x2b837d8 C:\IBM\WebSphere\App	oS 24	496	
🖭 🔟 nativelibpaths java.lang.String[0] @ 0x2b837f0	16	16	
IbraryClassLoaders com.ibm.ws.classloader.CompoundClassLoader[0]] 16	16	
Description: De	38 16	1,768	
🗉 🗋 reloadableParents java.util.Vector @ 0x2b83820	24	72	
🗉 🗉 🗋 resourceRequestCache java.util.Collections\$SynchronizedMap @ 0x2	b 16	3,360	
🛨 🗋 preDefinePlugins java.util.ArrayList @ 0x2b83848	24	72	-
	·		

Next, Click the reports icon Server->WAS Cache Analysis.

Gore.20130214.125316.9560.0001.dmp ☎								
i III 🖫 💀 🔎 = 🎭 = 🔍 🖾 = 🖾 = 🔚								
i Overview 🚱 Path to	going re	ferences 🔀	<u> </u>					
Class Name	5hallow	Heap Ret	ained Heap					
Regex> Co Path To GC Roots	<nur< td=""><td>neric></td><td><numeric></numeric></td><td></td><td></td><td></td><td></td><td></td></nur<>	neric>	<numeric></numeric>					
Com.ibm.ws.classlo		160	27,672					
	• 2	21,834	22,514					
IBM Extensions		5 Transaction	Gateway					
packages jav Java Basics	 Java 	SE Runtime		•				
packageSign Java Collections	 Utiliti 	ies		<u> </u>		_		
ImethodCache Leak Identification	Web	Sphere Applic	ation Server	•	Channel Framework (TCP)	•		
Immediate Dominators	Web	Sphere eXtre	me Scale	•	Class Loaders	•		
ConstructorC Show Retained Set		48	104		Database (JDBC)	•		
assertionLoci		8	8		Dynacache	•		
IazyInitLock Search Queries		8	8		HA Manager	•		
Des java.util.H		48	144		HTTP Sessions	•		
IocalClassPat History	•	24	496		Hung Threads	•		
Imativelibpaths java.lang.String[0] @ 0x25b6e00		16	16		XMC	•		
IbraryClassLoaders com.ibm.ws.classloader.Compoun	c	16	16		AlarmManager			
providers com.ibm.ws.classloader.SinglePathClassProvid	1	16	1,768		EJB Container			
reloadableParents java.util.Vector @ 0x25b76e0		24	72		PMI Data			
resourceRequestCache java.util.Collections\$Synchron	1	16	3,360		Statement Caches			
Image: PreDefinePlugins java.util.ArrayList @ 0x25b77c0		24	72		Thread Pool Analysis			
Image: Image: The second se	¢	24	120		WAS Cache Analysis			
Electric class com.ibm.websphere.samples.pbw.jpa.OrderItem @	1	398	414		WAS Overview			
Electric class com.ibm.websphere.samples.pbw.jpa.OrderItem\$P	ĸ	283	299		WAS Security			
E class com.ibm.websphere.samples.pbw.jpa.Inventory @	1	515	531		Web Container Analysis			
E Class com.ibm.websphere.samples.pbw.jpa.Order @ 0x2	8	673	689					
Electric class com.ibm.websphere.samples.pbw.jpa.BackOrder @		316	332					
class com.ibm.websphere.samples.pbw.jpa.Customer @	C	296	312					
Description: De	1	160	4,008					
≥. Total: 25 of 26 entries								

again and select IBM Extensions->WebSphere Application



🛢 core.20130303.050212.7744.0001.dmp 🛛

i 📊 🖫 💀 🔎 📲 🕶 🕵 🕶 🔍

👔 Overview 😂 Path to Leaking Application Class Loaders 🗋 with outgoing references 😂 was_cache_analysis 🛛

WebSphere Cache Analysis

WebSphere Cache Analysis

WAS Caches Found

Unknown Column: address	Unknown Column: cacheName	Unknown Column: currentCacheSize	Unknown Column: cacheSizeLimit	Unknown Column: diskOffload	Unknown Column: shallowHeap	Unknown Column: retainedHeap
0x2158c60	com.ibm.ws.wssecurity.sctClientCacheMap	0	2000	disabled	328	15584
0x1d2e738	com.ibm.workplace/ExtensionRegistryCache	12	5000	disabled	328	43808
0x1d28828	com.ibm.ws.wssecurity.sctServiceCacheMap	0	2000	disabled	328	15584
∑ Total: 3 entries						

▼ EJS Caches Found

Unknown Column: address	Unknown Column: cacheName	Unknown Column: entries	Unknown Column: softLimit	Unknown Column: hardLimit	Unknown Column: evictionStrategy	Unknown Column: shallowHeap	Unknown Column: retainedHeap
0x21a5508	EJB Cache	31	2053	2258	com.ibm.ejs.util.cache.BackgroundLruEvictionStrategy	64	170064
<u>0x21856e8</u>	Wrapper Cache	33	4106	4516	com.ibm.ejs.util.cache.SweepLruEvictionStrategy	64	114792
∑ Total: 2 entries							
Table Of Content	5					Cr	eated by <u>Memory Analy</u>

This illustrates the contents of the WebSphere caches such as dynacache. These types of caches reside in memory. If the WAS cache report shows a high memory footprint, the size of the caches can be limited with WebSphere administration, or the cache contents can be automatically off loaded to disk. For this lab, there is no action required.

__ Click the **reports** icon **Container Analysis**.

and select IBM Extensions->WebSphere Application Server->Web



🔒 core.20130303.050212.	7744.0001.dmp 🛛								
i III 🖫 💀 🕨 🕶	🗞 🕇 🔍								
i Overview 🌄 Path to	List objects Show objects by class	+ +	going refere	ences 🜏 was_cad	:he_analy	/sis 🛛			
WebSphere Cache An	😸 Path To GC Roots 🎥 Merge Shortest Paths t	o GC Roots 🔹 🕨							
WebSp	Eclipse IBM Extensions Java Basics) 	CICS Tra Java SE	nsaction Gateway Runtime	+				
▼ WAS Caches F	Java Collections Leak Identification	+ +_	Utilities WebSph	ere Application Ser	► ver ►	Class Loaders Database (IDBC)	+		
Unknown Column: address	Pail Immediate Dominators			Unknown Co currentCach	lumn: eSize	Dynacache HA Manager	•	lumn: nit	Unknown Colu diskOffload
<u>0x2158c60</u>	😪 Search Queries		heMap	0		HTTP Sessions	۲		disabled
0x1d2e738	History	•	ryCache	12		Hung Threads AlarmManager	•		disabled
∑ Total: 3 entries			вспемар	U		EJB Container PMI Data			disabled
▼ EJS Caches Fo	und					Thread Pool Analysis WAS Cache Analysis WAS Overview			
Unknown Column: address	Unknown Column: cacheName	Unknown Column: entries	Unk Cole soft	cnown umn: tLimit	Unk Coli har	WAS Security WAS Versions Web Container Analysis		ı Column: evi	ictionStrategy
0x21a5508	EJB Cache	31	205	3	225-	Contrainer Andrysis		ejs.util.cache.E	ackgroundLruEvic
III avatarica	w		44.0	/	ALC: N	:-			

This shows details of all the configured web applications. Observe that in this case only the **Plants by WebSphere application** has any active sessions. In the next step, we will check the memory size of these sessions.

Gore, 20130303.050212.7744.0001.dmp	
i III 🖫 😡 🔎 + 🚱 + 🔍	
i Overview 🗞 Path to Leaking Application Class Loaders 🗋 with outgoing references 民 was_cache_analysis 🕵 web_container_analysis 🕱	
Web Annitration Analysis	

Web Application Analysis

▼ Web Application Details

Unknown Column: address	Unknown Column: virtualHostName	Unknown Column: webGroupName	Unknown Column: webAppName	Unknown Column: Ioader	Unknown Column: destroyed	Unknown Column: currentSessions	Unknown Column: overflowSessions
<u>0x2667688</u>	default_host	/IBM_WS_SYS_RESPONSESERVLET/*	ibmasyncrsp		false	0	0
<u>0x1c370a8</u>	default_host	/PlantsByWebSphere/*	pbw-ear		false	31	0
∑ Total: 2 entries							

Table Of Contents

Created by <u>Memory Analyzer</u>



Click the **reports** icon

, select IBM Extensions->WebSphere Application Server->HTTP

Sessions->HTTP Sessions List. This shows details of all HTTP sessions including the size, session attributes, timeout, user ID and session ID.



Ensure the table of sessions is ordered by "Retained Heap". (**Click on the Retained Heap Column to order it as shown below**) Notice how some or most sessions for PlantsByWebSphere are around 23k. while others are ranging from 9K to 24k.

👔 Overview 😂 Path to Leaking Application Class Loaders 🗋 with outgoing references 😂 w	as_cache_analysis	: 😂 web_container_	analysis 🔒 HTTP Sessions List 🔅	
Class Name	Shallow Heap	Retained 🔺	AppName	SessionID
😫 <regex></regex>	<numeric></numeric>	<numeric></numeric>	<regex></regex>	<regex></regex>
E Com.ibm.ws.session.store.memory.MemorySession @ 0x3580a80 for memory store pbw- bw-ibm.ws.session.store.memory.MemorySession @ 0x3580a80 for memory.	112	9,912	default_host/PlantsByWebSphere	oTrm4G35OpTwY7-26
🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x72bce88 for memory store pbw-6	112	16,448	default_host/PlantsByWebSphere	QQxCH3F0185xFZqJ
en 🗅 et el compañíaco (el compañíaco (e			1.8 6.1 (64) (67) (7)	

Impact2013
Business. In Motion.

😫 core.20130303.050212.7744.0001.dmp 🕅							
i III 🖳 👊 🛌 - 🚱 - 🔍 🖾 - 🛃 - 🤚							
👔 Overview 🔀 Path to Leaking Application Class Loaders 🍙 with outgoing references 😫	was_cache_analysis	👸 web_container_	analysis 😪 HTTP Sessions List 🛛				
Class Name	Shallow Heap	Retained 🔺	AppName	SessionID			
🔆 <regex></regex>	<numeric></numeric>	<numeric></numeric>	<regex></regex>	<regex></regex>			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x3580a80 for memory store pl	ow-i 112	9,912	default_host/PlantsByWebSphere	oTrm4G35Op			
🛛 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x72bce88 for memory store pt	ow-e 112	16,448	default_host/PlantsByWebSphere	QQxCH3F01			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x72bb850 for memory store pl	DW-1 112	17,128	default_host/PlantsByWebSphere	6Coa3Tqkah			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x403f320 for memory store pt	оw-е 112	23,920	default_host/PlantsByWebSphere	ulDfYgngJxd			
🗄 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x3d2c1e8 for memory store pt	ow-e 112	23,928	default_host/PlantsByWebSphere	TyeQA0jJv0			
🗄 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x4aa09b8 for memory store pl	DW-1 112	23,936	default_host/PlantsByWebSphere	OcbLxydKlUB			
🛛 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x35828e8 for memory store pl	ow-i 112	23,936	default_host/PlantsByWebSphere	sR1l6Ald-Iun			
🛛 🕀 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x4aa0930 for memory store pl	bw-i 112	23,944	default_host/PlantsByWebSphere	H3_r_LBIIqs			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x324b958 for memory store pl	ow-i 112	23,944	default_host/PlantsByWebSphere	8XBJJhn7cw			
🛛 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x30e7890 for memory store pl	ow-i 112	23,944	default_host/PlantsByWebSphere	zmMvtCpSSn			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x2d910f0 for memory store pb	оw-е 112	23,944	default_host/PlantsByWebSphere	iHFQu0PMMc			
🛛 🕀 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x4fd82c8 for memory store pb	w-e 112	23,952	default_host/PlantsByWebSphere	oBD87-YM_E			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x3582958 for memory store pl	DW-1 112	23,952	default_host/PlantsByWebSphere	eWNf7gD1d			
🗄 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x39f52b8 for memory store pt	оw-е 112	23,960	default_host/PlantsByWebSphere	XjD4623-Dxh			
🛛 🕀 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x324b9c8 for memory store pt	ow-e 112	23,960	default_host/PlantsByWebSphere	CYzg8gppWł			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x2d91080 for memory store pl	DW-1 112	23,960	default_host/PlantsByWebSphere	uyWl9wRRal			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x60f1170 for memory store pb	ow-ε 112	23,968	default_host/PlantsByWebSphere	5GRB8YO06			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x5a62d10 for memory store pl	DW-1 112	23,968	default_host/PlantsByWebSphere	Cm2dHesirY2			
🛛 🕀 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x347caa8 for memory store pt	ow-e 112	23,968	default_host/PlantsByWebSphere	xB_blLfUQQi			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x3250078 for memory store pl	bw-i 112	23,968	default_host/PlantsByWebSphere	KMUDH5sil_c			
🛛 🕀 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x30e7918 for memory store pl	bw-i 112	23,968	default_host/PlantsByWebSphere	6_npVsglsDK			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x60f10e8 for memory store pt	оw-е 112	23,976	default_host/PlantsByWebSphere	4V8uii_vinVV			
🛛 🕀 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x5a62d80 for memory store pl	DW-1 112	23,976	default_host/PlantsByWebSphere	hAKk71U_Tz			
🛛 🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x4fd8ec8 for memory store pb	w-е 112	23,976	default_host/PlantsByWebSphere	-lxqJ8lbGmk7			
🛛 🕀 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x392d720 for memory store pl	DW-1 112	23,976	default_host/PlantsByWebSphere	bydytilspr 7			
📗 🕀 🗋 com.ihm.ws.session.store.memory.MemorySession @ 0x44r10d0 for memory store of	112 nw-f	23 992	default_bost/PlantsBvWebSobere	wWFK2D2Vof			

____ Let's Highlight and then expand one of the **23k** session objects. (For this example, the first 23k in the ordered list was picked as shown below)

Gere.20130303.050212.7744.0001.dmp ⊠										
i III 🖫 💀 🖳 + 🚱 + Q, 🖾 + 🛃 + 🦺										
👔 Overview 😤 Path to Leaking Application Class Loaders 📓 with outgoing references 😫 was_cache_analysis 😤 web_container_analysis 😫 HTTP Sessions List 🕱										
Class Name	Shallow Heap	Retained 🔺	AppName	SessionID						
<regex></regex>	<numeric></numeric>	<numeric></numeric>	<regex></regex>	<regex></regex>						
🗉 📋 com.ibm.ws.session.store.memory.MemorySession @ 0x3580a80 for memory store pbw-i	112	9,912	default_host/PlantsByWebSphere	oTrm4G35OpTwY7-26XtF						
🗉 🗈 com.ibm.ws.session.store.memory.MemorySession @ 0x72bce88 for memory store pbw-6	112	16,448	default_host/PlantsByWebSphere	QQxCH3F0185xFZqJxk5						
🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x72bb850 for memory store pbw-i	112	17,128	default_host/PlantsByWebSphere	6Coa3TqkahZImXlkCNvsE						
🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x403f320 for memory store pbw-e	112	23,920	default_host/PlantsByWebSphere	ulDfYgngJxdzHLnmXpw8l						
🗉 📄 com.ibm.ws.session.store.memory.MemorySession @ 0x3d2c1e8 for memory store pbw-6	112	23,928	default_host/PlantsByWebSphere	TyeQA0jJv0cW6_vljKY10						
🗉 🗈 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x4aa09b8 for memory store pbw-i	112	23,936	default_host/PlantsByWebSphere	OcbLxydKlUEupmQFct6D						
🗉 🗈 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x35828e8 for memory store pbw-(112	23,936	default_host/PlantsByWebSphere	sR1l6Ald-IunE1azRnntiSc						
🗉 🕀 🗋 com.ibm.ws.session.store.memorv.MemorvSession @ 0x4aa0930 for memorv store bbw-i	112	23.944	default_bost/PlantsRvWebSobere	H3 r LBIInsDM vk5350L						

The session attributes have been automatically extracted from the WebSphere objects and presented by the IBM Memory Analyzer extensions. Notice there is a key called "deliberateLargeSession" which contains a very long String of letter "C"s. This looks like another deliberate mistake in the Plants application.

🕀 📄 All Outgoing References of the Session Object	0	0
Key=javax.faces.request.charset,Value=ISO-8859-1	24	24
Key=WebBeansConfigurationListener, Value=com.ibm.ws.webbeans.services.JCDIWebListener @ 0x2b34c88	24	24
Key=jsf_sequence, Value=java.lang.Integer @ 0x844b48 17	24	248
Key=deliberateLargeSession,Value=byte[10240] @ 0x4040aa8 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC	24	10,272
Key=org.apache.myfaces.view.facelets.DefaultFaceletsStateManagementHelper.SERIALIZED_VIEW,Value=org.apache.myface	: 24	12,816
∑ Total: 6 entries		

_____ Make a note of the "_sessionId" for the object. This can be seen on the "Attributes" tab as shown below on the left under the Inspector tab.

🍉 IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer	
<u>File Help Views</u>	
🔛 🅞 Memory Analyzer	
🖸 Inspector 🛛 🤤 🗖 🗖	eore.20130303.050212.7744.0001.dmp ⋈
@ 0x403f320	i III 🗣 💀 🔎 + 🗞 + 🔍 🖾 + 🛃 - 🔚
MemorySession	
🖶 com.ibm.ws.session.store.memory	i Overview 🔀 was_cache_analysis 🛃 HTTP Sessions List 🛛
class com.ibm.ws.session.store.memory.MemorySession @ 0x197bfc8	Class Name
🔍 java.lang.Object	🔆 <regex></regex>
🙆 org.eclipse.osgi.internal.baseadaptor.DefaultClassLoader @ 0x13fbe38	🗉 📋 com.ibm.ws.session.store.memory.MemorySession @ 0x3580a80 for memory store pbv
112 (shallow size)	🗄 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x72bce88 for memory store pbv
1 23,920 (retained size)	🗄 🗄 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x72bb850 for memory store pbv
o no GC root	😑 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x403f320 for memory store pbw
Statics Attributes Class Hierarchy Value	All Outgoing References of the Session Object
Type Name Value	Key=WebBeansConfigurationListener, Value=com.ibm.ws.webbeans.services.JCDIWebLi
ref storeCallback com.ibm.ws.session.StoreCallback@0x1c4fec0	
refsessionIdulDfYgngJxdzHLnmXpw8Pw6	Key=deliberateLargeSession,Value=byte[10240] @ 0x4040aa8 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
boolean _isValid true	Key=org.apache.myfaces.view.facelets.DefaultFaceletsStateManagementHelper.SERIAI
boolean _isNew false	∑ Total: 6 entries
long _currentAccessTime 1362315605093	🗄 🕒 com.ibm.ws.session.store.memory.MemorySession @ 0x3d2c1e8 for memory store pbv
long creationTime 1362315585706	🗉 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x4aa09b8 for memory store pbv
int version 0	E 🗋 com.ibm.ws.session.store.memory.MemorySession @ 0x35828e8 for memory store pbv
ref _userName anonymous	Com.ibm.ws.session.store.memory.MemorySession @ 0x4aa0930 for memory store pby
int _maxInactiveInterval 600	T com.jbm.ws.session.store.memory.MemorySession @ 0x324b958 for memory store bb
int refCount 0	

Right click it and choose **Copy->Value**, as shown below.

Statics	Attributes Cla	ss Hierarchy	Value					+	<u> D</u> /	All Outgoing Refe	erences of t	:he
T		so r nor ar cri,	10.00					H 🗉	Key=	javax.faces.reqi	uest.charsel	st,
Туре			Value					+	Key=	WebBeansConfig	gurationListe	ег
rer	_store	ck	pow-e	ar at /Pi	ants	By WebSphere currently v B StoreCallback @ 0v1c4	Altr fec0	+	Key=	isf_sequence,Va	lue=java.la	ani
ref	storecaliba		UDEV	ina lydzi	:5510	II.Storecaliback@OxIC4	ieco	1 - T	Kev=	deliberateLarge9	Session.Valu	ie:
boolear	isValid		true	Ingoxozi		Go Into			Kev-	ora anache myf:	erec view f:	ar
boolear	i _isNew		false			List objects		•	Σι	otal: 6 entries	;	
long	_currentAcc	ess lime	13623	156050		Show objects by class		- + I	com.	ibm.ws.session.	store.memo	۶rs
long		en me	13623	156045	1	Dath To CC Doote			com.	ibm.ws.session.	store.memo	ors.
int	_creation		0	155057	60	Path to GC Roots			com.	ibm.ws.session.	store.memo	ors.
ref	userName		anony	mous	1 222	Merge Shortest Paths to	GC Roots	•		ibm we session :	ctore memo	<u>~ 7</u>
int	maxInactiv	eInterval	600			IBM Extensions		F		ibm.ws.session	store.memo	<u>~</u>
int	refCount		0			Java Basics		F	Com.	iom.ws.session.:	score.memo	жγ
ref	_appName		defau	lt_host/F		Java Collections		•	com.	IDM.WS.SESSION.	store.memo	yry
ref	_attributes		java.u	ıtil.Hash] com.	ibm.ws.session.	store.memo	Jr
ref	_attributeNa	mes	java.u	ıtil.Hash	_	Leak Identification) com.	ibm.ws.session.	store.memo	۶r
ref	_adaptation		com.it	m.ws.se	1	Immediate Dominators) com.	ibm.ws.session.	store.memo	brγ
ret boolear	_appAdapta i isOverflow	tion	null false		1	Show Retained Set) com.	ibm.ws.session.	store.memo	γrγ
boolear	invalInProgr	ess	false			Сору		•	📄 Ado	lress	iemo	»r
ref	_gcCount		null		8	Search Queries			Clas	s Name	iemo	γŊ
ret	_smc		com.it	0m.ws.s6		mooooloninahagoi coning			🕒 Vali	10	iemo	Jr
rer	аррічатедні	atastring						E .			iemo	»r
								E I	Sav	e Value To File	iemo	»rγ
								±	📄 Sele	ection	iemo	brγ
								E C] com.	ibm.ws.session.	store.memo	۶r

A possible way to relate this unusually large session to the application code is to search the log files for the session ID. If the application uses this in its logging, it may be possible to determine what the user did to cause the large session.

Note:

This lab provides a SystemOut.log file that correlates with the time the system dump was generated. This will be used to search for more details about the large session size.

____Launch Windows Explorer. Navigate to and right click file: E:\Impact Lab Files\IBM Extensions\SystemOut.log, then select Edit with Notepad++ as shown below.

🔲 core.zo	130214,123310,9300,000	n_web_Appicac	0 ND	Compressed (zipp
🚺 🚺 core.20	130214.125316.9560.000	8 KB	Compressed (zipp	
📕 🖻 core.20	130303.050212.7744.000)1.dmp	381,708 KB	DMP File
📕 🗐 System	Outloa		532 KB	Text Document
🔲 🖬 core.20	<u>O</u> pen	hreads	114 KB	THREADS File
🗖 🖬 core.20	Print	x.index	7,208 KB	INDEX File
🛛 🗟 core.20	7-Zip	2c.index	2,618 KB	INDEX File
🛛 🖬 core.20	Edit with Notepad++	2s.index	2,008 KB	INDEX File
🗖 🖬 core.20	Open Wit <u>h</u>	bound.index	11,121 KB	INDEX File
🗖 🖬 core.20	Sond To	Lutbound.index	11,121 KB	INDEX File
🗖 core.20	Sella to	dex	22,926 KB	INDEX File
🛛 🗖 core.20	Cu <u>t</u>	pmIn.index	2,749 KB	INDEX File
🛛 🖬 core.20	⊆opy	pmOut.index	8,255 KB	INDEX File
🗖 core.20	Croate Shortaut	2ret.index	7,217 KB	INDEX File
Core.20	Create <u>p</u> hortcut	Veb_Applicati	8 KB	Compressed (zipp
Core.20	<u>D</u> elece Rena <u>m</u> e	VebSphere_C	8 KB	Compressed (zipp
	Properties			



After Notepad++ opens the systemOut.log file. Select "**Ctrl + F**" or Click **Search >Find** and search the file for the Session ID you identified or copied earlier from the core dump. You should be able to copy the id from your clipboard into the search window. It should then reveal a log statement that gives a clue about the application's actions as shown below.

```
      [3/3/13 4:59:58:718 PST]
      0000009e SystemOut
      0 performProductDetail : itemID=F0018

      [3/3/13 4:59:58:734 PST]
      0000009e SystemOut
      0 ==> STARTING DELIBERATE LARGE SESSION for ID=ulDfYgngJxdzHLnmXpw8Pw6

      [3/3/13 4:59:58:734 PST]
      0000009e SystemOut
      0 ==> ENDING DELIBERATE LARGE SESSION

      [3/3/13 4:59:59:171 PST]
      00000098 SystemOut
      0 performProductDetail : itemID=F0015

      [3/3/13 4:59:59:718 PST]
      0000009e SystemOut
      0 performProductDetail : itemID=F0015
```

Optional Steps:

__ Double click the desktop shortcut as shown below for ShoppingBean.java



Click Edit->Find and and search for "STARTING DELIBERATE LARGE SESSION".

The deliberate mistake is clear – for any user that clicks on the white poinsettia image, their session is loaded with an attribute containing a 10k string of "C"s (ASCII code 67). Congratulations, you have successfully located the final deliberate mistake in the plants sample.

private void deliberateLargeSession(ExternalContext externalContext) {

// ----// User clicked on the White Poinsettia, let's make the session for this user
// blooming big.
// ------

System.out.println("==> STARTING DELIBERATE LARGE SESSION for ID="+((HttpSession)externalContext.getSession(true)).getId());

byte[] sessionAttr = new byte[10240]; Arrays.fill(sessionAttr, (byte) 67); ((HttpSession)externalContext.getSession(true)).setAttribute("deliberateLargeSession", sessionAttr);

```
System.out.println("==> ENDING DELIBERATE LARGE SESSION");
```



Part 9: (Optional) Using the WebSphere Application Server Configuration Visualizer

This part of the lab as the title indicates is optional. However, this is available for you to do as time permits.

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

To demonstrate one of the report generator tools, first switch to the "**PlantsByWebSphere**" case that has been prepared for this lab, and click the **Files** tab.

	IBM Support Assistant	
Cases 🕨	[0003] PlantsByWebSphere	~
	[0000] Example Case	
D Files	[[0002] lab case 2	L.
	[0003] PlantsByWebSphere	K

Note:

The case contains file "**Collector tool output impact2013Cell.jar**" as shown below. This is the 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.

	BM Supp	oort Assistar	nt					~			(iot I
Cases 🕨	[0003] Pla	ntsByWebSphere				•				s	can this Case) (
Files	💼 Tools	🚺 Reports 🕻) c)verview	Symptoms	🗐 Global Ki	nowledge Basi	e Matches 📲	• Data Collec	tor		
E Tree	View 🔻 🕻	CASE:0003/*									Searc	h Fili
Add file	s	Or drag files into browser to add		Name Filt	er 🕌			<u>Filter</u> B	leset			
Navigat	or			Name			Symptoms	Knowledge Ba	First Timestam	Last Timestam	Size	Тур
00	03	Ŷ		Collector to	ol output impact201:	3Cell.jar					6 MB	jar



To demonstrate one of the report generator tools, return to the Tools tab and select **WebSphere Application Server Configuration Visualize**r as shown below. Then press the launch button.

🥹 0003 - IBM Support Assistant 5.0 - Mozilla F	Firefox			_ 🗆 ×
<u> E</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools	<u>H</u> elp			
\bigoplus WebSphere Integrated Solutions Console 🛛 🗙	0003 - IBM Support Assi	stant 5.0 × +		
Contempt Not the second	ols&id=0003&toolId=com.it	bm, wsvisualizer	☆ マ C Scogle	۶
IBM Support Assistant			Got Feedback? Help	IBM.
Cases [0003] PlantsByWebSphere		•	Scan this Case 📀 🔻 Global Filter	- Off 🔻
🗈 Files 🗀 Tools 📙 Reports 📋 C	Overview 🏭 Sympto	oms Global Knowledge Base Matches 🗄 Data Collector		
			Search Tool Help	Q
Enter keyword Filter Reset		WebSphere Application Server Configuratio	n Visualizer Version 1.0.0	
Sort By:	•	Launch (1) Tool Help		
Garbage Collection and Memory Visualizer (GCMV) [Desktop]	<			-
Garbage Collection and Memory Visualizer (GCMV) [Report]	✓ <u>Iu</u>	WebSphere Application Server Configuration Visualizer		
THealth Center	< □	Generates an interactive HTML visualization of a WebSphere Application Server configuration, including Service Integration		
🖥 HeapAnalyzer [Desktop]	⁶ 9 🗊	Buses and databases. Accepts any combination of zips or jars		
🖥 Memory Analyzer [Desktop]	√ □	containing configuration directories. Configurations from multiple nodes in a cell will be merged into a single		
T Memory Analyzer [Report]	√ <u>I</u>	visualization, and multiple cells can be displayed in the		
TMemory Analyzer Web Edition [Web]	69 💿	output.		
Thread and Monitor Dump Analyzer (TMDA) [Desktop]	°5 🗖		No Screenshot Available	
Thread and Monitor Dump Analyzer (TMDA) [Report]	°S 📊			
WebSphere Application Server Configuration Visualizer	°S 🔟	Tags: 📊 Report Generator Tool 69 As-is Problem Area: Web:	Sphere Problem Area: Configuration	
		Restrictions:		-
Build ID: 5.0.0.0_Beta2_20121016-1409			© Copyright IBM Corp. 2011, 2012. All rights :	reserved

_ Press the Browse button from the above window to select the input files.

Problem Analysis			X				
Run WebSphere Application Server Configuration Visualizer (Version 1.0.0)							
Input Files and Fold	ers *						
			Browse				
Parameters							
Parameter	Description	Value					
ReportName	Report Name	topology.html					
Run as background task:							
			Submit Cancel				



х

Select a File	х
- Control -	
Collector tool output impact2013Cell.jar	
SystemOut.log	_
OK Cano	cel 🛛

Select the file Collector tool output impact2013Cell.jar as shown above and press OK.

Problem Analysis

Run WebSphere Application Server Configuration Visualizer (Version 1.0.0)

/ISA5Beta2/ISA5/isa/cases/0003/Collector tool output impact2013Cell.jar B	rowse
---	-------

Parameters							
Parameter	Description	Value					
ReportName	Report Name	topology.html					
Run as background task: 🜌							
			Submit Cancel				

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	녪 Reports	📋 Overview	🏭 Symp			
Enter keyword <u>Filter</u> Reset							
Sort By: \downarrow_Z^a \downarrow_Q^o							
WebSphere Application Server Configuration Visualizer			1/3/13 06:41:56				
Collector tool output impact2013Cell.jar							



Select the completed report.

Files	💼 Tools	🔝 Reports	📋 Overview	🏭 Sym		
Enter keyword <u>Filter</u> Reset						
Sort By:	↓ <mark>a</mark> ↓o			Ŷ		
WebSphere Application Server Configuration Visualizer 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.





Reference Links

- IBM Support Assistant Information and Downloads:
 - o <u>http://www-01.ibm.com/software/support/isa/</u>
 - o ISA 5: http://www-01.ibm.com/software/support/isa/teamserver.html
- Health Center
 - o http://www.ibm.com/developerworks/java/jdk/tools/healthcenter/
- Configuring a controller to start on a specific node or deployment manager
 - o http://www-01.ibm.com/support/docview.wss?uid=swg21425281
- Verify Java SDK version shipped with IBM WebSphere Application Server fix packs:
 - o <u>http://www-01.ibm.com/support/docview.wss?uid=swg27005002</u>
- WAS 8.5 Infocenter:
 - <u>http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-dist&topic=welcome_nd</u>
- What's new in WebSphere Application Server 8.0 & 8.5
 - o http://www.ibm.com/developerworks/websphere/techjournal/1106 alcott/1106 alcott.html
 - o http://www.ibm.com/developerworks/websphere/techjournal/1206 alcott/1206 alcott.html