



What You Need to Know About CICS Java Performance

Ian Burnett

CICS TS for z/OS Performance Lead, IBM



SHARE is an independent volunteer-run information technology association
that provides **education, professional networking and industry influence.**

Copyright (c) 2015 by SHARE Inc. <http://creativecommons.org/licenses/by-nc-sa/3.0/>



SHARE in Orlando

Tuesday, 11th August

Ian Burnett

IBM CICS TS for z/OS Performance Lead

ian.burnett@uk.ibm.com

@IanBurnett

Legal Disclaimer



- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- IBM, the IBM logo, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Agenda

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Topics covered in this session:

- An overview of the CICS Explorer
- An overview of the IBM Health Center
- How to install the IBM Health Center into CICS Explorer
- A look at some useful views within the Health Center



CICS Explorer

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Overview

- Runs on workstation
- Graphical interface
 - Based on Eclipse
- View and manage multiple CICS TS regions easily
- Task-oriented views
- Context-sensitive resource editors and wizards
- Excellent integration with CICS tools
- Also includes IBM Explorer for z/OS



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Views



Regions

Region	Job Name	MVS System ID	Task Count	CICS Status	CICS TS Level	Release
IYCUZC27	IBTOR01	MV2A	3	✓ ACTIVE	050200	0690
IYCUZC28	IBTOR02	MV2A	3	✓ ACTIVE	050200	0690
IYCUZC29	IBTOR03	MV2A	3	✓ ACTIVE	050200	0690
IYCUZC30	IBTOR04	MV2A	3	✓ ACTIVE	050200	0690
IYCUZC31	IBAOR001	MV2A	3	✓ ACTIVE	050200	0690

Complete your session evaluations online at www.SHARE.org/Oriando-Eval



Systems view



The screenshot displays the IBM CICSpI Explorer interface. On the left, a tree view shows the hierarchy: Server: IB25, Workload Management, Systems, System Groups, ALLRGN5, AOR, TOR, IYCUZC27, IYCUZC28, IYCUZC29, IYCUZC30, IBPLEXW, Workload Management, Systems, IYCUZC26, System Groups. A red box highlights this tree view. In the center, a table lists transactions with columns for Region, Job Name, MVS System ID, Page, and Count. A red box highlights the table. On the right, a pane titled 'CICSpI Explorer' shows a tree view for 'Server: IB25' with sub-nodes for Workload Management, Systems, System Groups, ALLRGN5, AOR, TOR, IYCUZC27, IYCUZC28, IYCUZC29, IYCUZC30, IBPLEXW, Workload Management, Systems, IYCUZC26, and System Groups. A red box highlights this pane. At the bottom left, a 'Connections' pane shows the system tree for 'CICSpI Explorer'.

Region	Job Name	MVS System ID	Page	Count
IYCUZC27	BT0R01	MQA	0	11360
IYCUZC28	BT0R02	MQA	0	11360
IYCUZC28	BT0R03	MQA	0	11360
IYCUZC28	BT0R04	MQA	0	11360
IYCUZC28	BA0R01	MQA	0	970
IYCUZC29	BA0R02	MQA	0	970
IYCUZC29	BA0R03	MQA	0	970
IYCUZC29	BA0R04	MQA	0	970

Complete your session evaluations online at www.SHARE.org/Oriando-Eval



Perspectives



The screenshot displays the IBM OCS Explorer application. On the left, a tree view shows the hierarchy of OCS systems. The main area contains a table with the following columns: Region, Job Name, MMS System ID, Task Count, OCS Status, OCS % Level, Release, Total CPU, Page-In Count, Page-Out Count, and I/O Count. A vertical toolbar is highlighted with a red box, containing icons for different perspectives: a calendar, a person, a gear, a cloud, and a document. A red arrow points from the toolbar to the tree view.

Region	Job Name	MMS System ID	Task Count	OCS Status	OCS % Level	Release	Total CPU	Page-In Count	Page-Out Count	I/O Count
FCU0227	BT0001			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	11360
FCU0228	BT0002			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	11360
FCU0229	BT0003			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	11360
FCU0230	BT0004			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	11360
FCU0231	BA0001			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	970
FCU0232	BA0002			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	970
FCU0233	BA0003			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	970
FCU0234	BA0004			✓ ACTIVE	00.00	0000	0000.00.00.00.00	0	0	970

Complete your session evaluations online at www.SHARE.org/Oriando-Eval





IBM Health Center

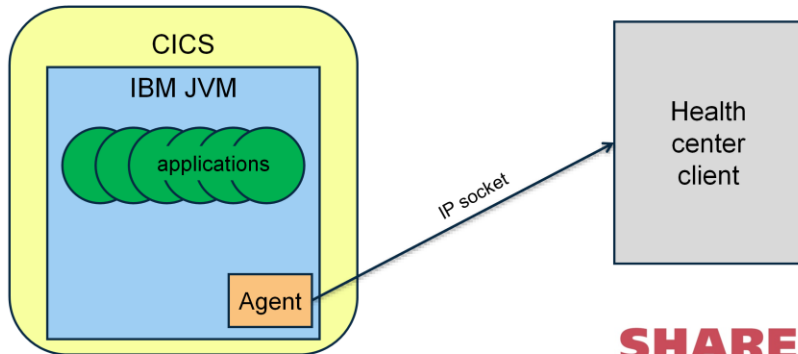
Complete your session evaluations online at www.SHARE.org/Orlando-Eval



What is the IBM Health Center?



- Diagnostic tool for IBM Java Virtual Machine (JVM)
- Two components
 - An agent installed into the JVM
 - A graphical client on your workstation



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



10

Health Center Client



- One of the tools supplied by the IBM Support Assistant
- Stand-alone Eclipse application
- Plug-ins for an Eclipse environment
 - CICS Explorer is an Eclipse environment

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



11



Installation

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Part 1: Installing workstation client



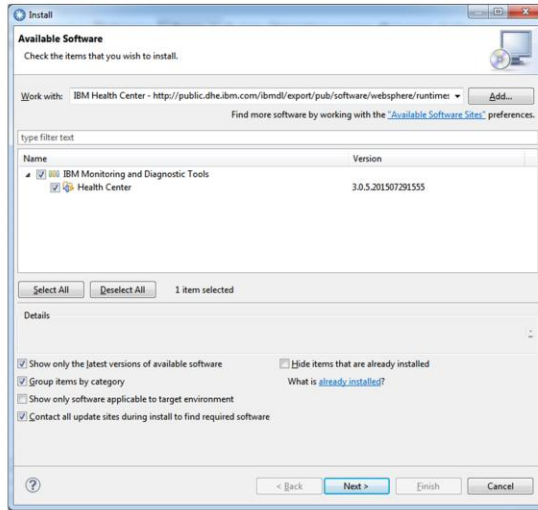
- Add a new *repository* to an existing CICS Explorer instance
 - **Help** → **Install new software ...**
 - **Add...**
 - public.dhe.ibm.com/ibmdl/export/pub/software/websphere/run-times/tools/healthcenter/
- Select the Health Center tools
- Click **Next** through the following screens
 - Includes acceptance of licence agreement
- Will require a restart of CICS Explorer

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



A repository is an Eclipse concept – a location (either on your local disk or on a remote server) which contains one or more features which may be installed into an Eclipse environment.

Installation selection screen



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Part 2: Installing Health Center agent into CICS



- Add parameters to CICS JVM profile

```
OVERTYPE TO MODIFY
CEDA DEFINE JVMserver( MYJVMSRV )
  JVMserver      : MYJVMSRV
  Group          : ISB
  DESCRIPTION    ==>
  Status         ==> Enabled
  JVMprofile     ==> MYJVMPRF
  LERUNOPTS     ==> DFHAXRO
  Threadlimit   ==> 015
DEFINITION SIGNATURE
```

This file on HFS
(see JVMPROFILEDIR)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



15

Agent parameters



- Simple case:
 - Start agent immediately and open a port
 - `-Xhealthcenter:port=1972`
- Advanced usage:
 - Start collecting when client connects
`-Dcom.ibm.java.diagnostics.healthcenter.data.collection.level=off`
 - Collect data on z/OS system (no transfer to agent)
`-Dcom.ibm.java.diagnostics.healthcenter.data.collection.level=headless`
- Options described in Help section of CICS Explorer

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



16

Agent starting



- The agent will initialize when the JVM server is started
- Will issue messages like the following to stderr

```
[Thu Aug 6 13:25:51 2015] com.ibm.diagnostics.healthcenter.java INFO: Health
Center 3.0.0.20141209
Aug 06, 2015 1:25:51 PM
com.ibm.java.diagnostics.healthcenter.agent.mbean.HCLaunchMBean <init>
INFO: Agent version "3.0.0.20141209"
Aug 06, 2015 1:25:51 PM
com.ibm.java.diagnostics.healthcenter.agent.mbean.HCLaunchMBean createJMXConnector
INFO: IIOP will be listening on the next available system assigned port. Use
com.ibm.java.diagnostics.healthcenter.agent.iiop.port to specify a port
Aug 06, 2015 1:25:51 PM
com.ibm.java.diagnostics.healthcenter.agent.mbean.HCLaunchMBean startAgent
INFO: Health Center agent started on port 32105.
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



17

Verification



- You can check the Health Center agent is listening using the netstat command

```
netstat ( port 32105
```

```
MVS TCP/IP NETSTAT CS V2R1
```

```
User Id Conn State
```

```
-----
```

```
IBAOR001 09290784 Listen
```

```
Local Socket: :...32105
```

```
Foreign Socket: :...0
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



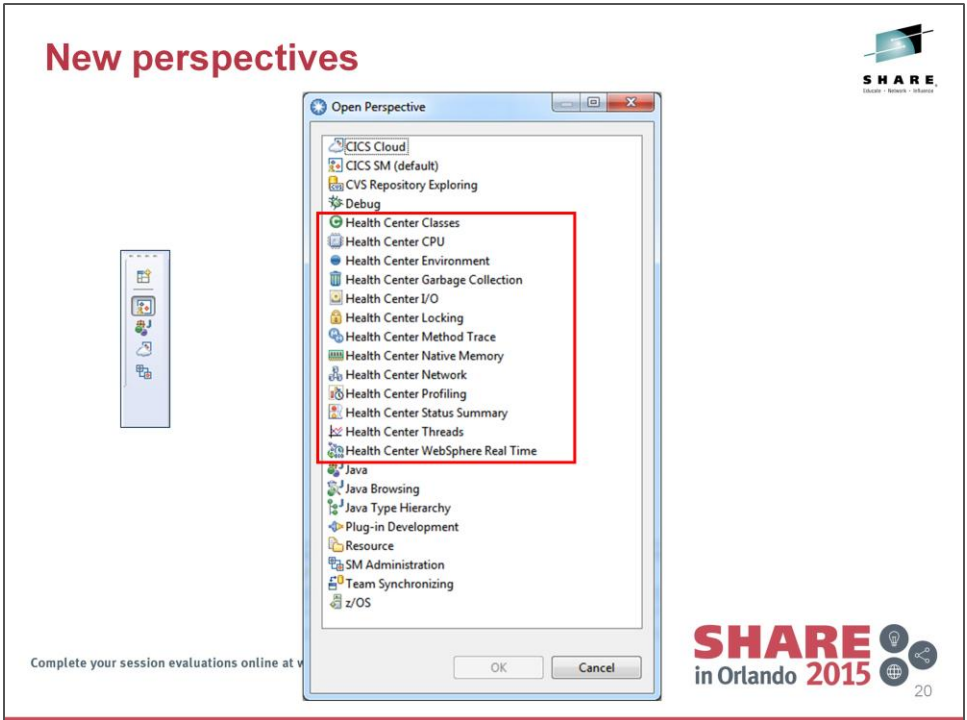
18



Connecting

Complete your session evaluations online at www.SHARE.org/Orlando-Eval





Installation of IBM Health Center adds a collection of new perspectives which allow you to customise your screen based on your area of interest.

Create a new connection



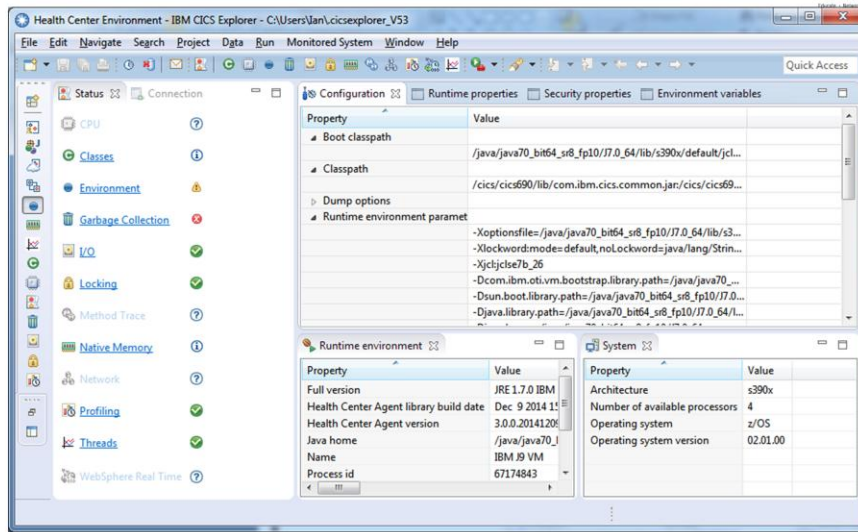
- Open the Health Center Environment perspective
 - **Window** → **Open Perspective** → **Other ...**
 - **Health Center Environment**
- Create a new connection
 - **File** → **New Connection ...**
 - Enter hostname, port and authentication details
 - Click **Finish** to connect

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



21

Environment perspective



Complete your session evaluations online at www.SHARE.org/Orlando-Eval




The environment perspective is a launchpad for other perspectives which provide more detailed information about the connected JVM.

Usage

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

We now move on to looking at some of the data which the Health Center client can display.

Classes perspective



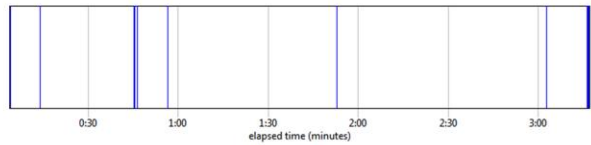
Status | Connection

- CPU
- Classes**
- Environment
- Garbage Collection
- I/O
- Locking
- Method Trace
- Native Memory
- Network
- Profiling
- Threads
- WebSphere Real Time

Analysis and Recommendations

- Class [C] has the most instances: 47,375 occupying 4577 KB of the heap
- Class [B] is occupying the most heap: 5048 KB by 15,867 instances
- Your application has loaded 1,140 classes. Make sure you have class sharing enabled with the -Xshareclasses option to save memory and reduce JVM startup time.

Class loading timeline




elapsed time (minutes)

Classes loaded | **Class histogram**

Filter classes: Apply Clear

Time loaded	Shared cache	Classname
3:16 minutes	No	com/ibm/cicsts/transform/channel/PipelineContainerBase
3:16 minutes	No	com/ibm/ras/RASMaskChangeEvent
3:16 minutes	No	com/ibm/cicsts/wlp/channel/ChannelStateChannelType
3:16 minutes	No	com/ibm/cicsts/transform/channel/PipelineContainerBase\$ContainerMode
3:16 minutes	No	com/ibm/cicsts/transform/dataTransform/DFHPIL009Exception
3:16 minutes	No	com/ibm/cicsts/wlp/channel/ContainerUnchangedException
3:16 minutes	No	com/ibm/cicsts/transform/dataTransform/DFHPIL008Exception
3:16 minutes	No	com/ibm/cicsts/transform/channel/PipelineContainerBase\$ContainerDataType
3:16 minutes	No	com/ibm/cicsts/gen/api/Platform\$EndianType
3:16 minutes	No	com/ibm/cicsts/transform/dataTransform/ConversionLib\$IntegerValues
3:16 minutes	No	com/ibm/jzos/fields/BinaryAsLongField
3:16 minutes	No	com/ibm/cicsts/transform/dataTransform/DFHPIL010Exception

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Density of class loading over time

Which classes were loaded at which time

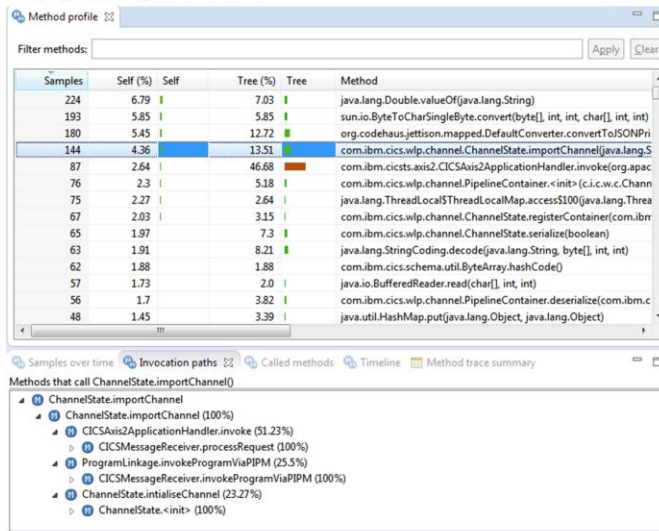
Whether a class was loaded from the class sharing cache

Also available is class histogram data

A snapshot of the classes that are in the heap

The amount of heap space that the instances are occupying

Profiling perspective



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



The Just-In-Time (JIT) compiler within the JVM uses a sampling approach to decide which Java methods should be more aggressively compiled. This sampling data is exposed in Health Center in the profiling perspective.

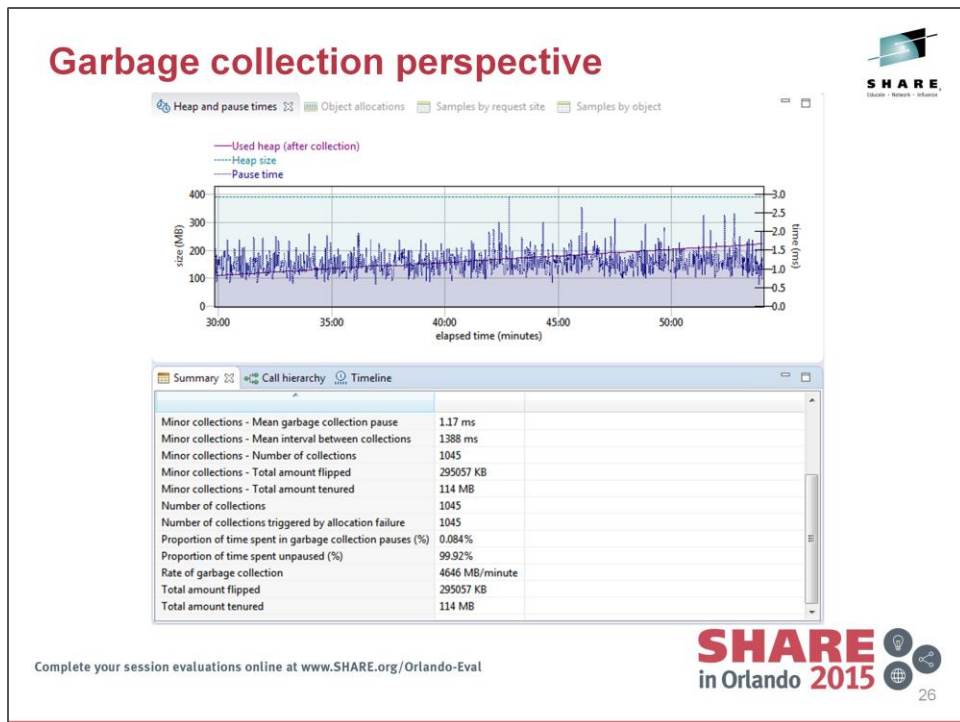
This data provides method-level profiling of the applications running in the JVM.

Methods can be filtered by class or package name.

The Method profile view shows sample counts for specific methods.

Self is when the method is at the top of a call stack and tree is when a method appears in a call stack.

Additionally the Invocation and Called method views allows you to analyze the call path of each profiled method to ascertain how it was invoked, and what further methods it calls.



The Garbage Collection perspective provides a set of views to assist in analyzing the garbage collection (GC) process used by the JVM to manage memory in the JVM heap.

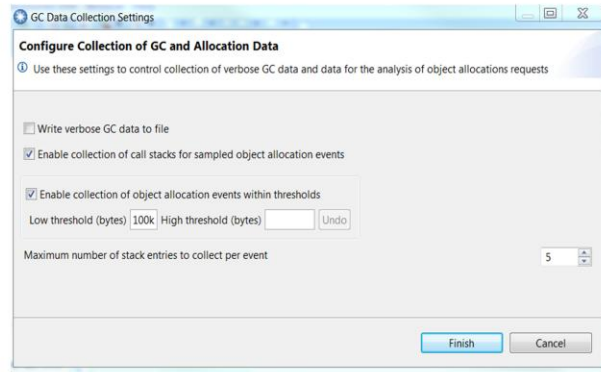
Using the default gencon GC policy splits the Java heap into two areas, the new or nursery area, and the old or tenured area. CICS JVM server statistics call new or nursery activity minor GC and call old or tenured activity major GC.

The Summary view provides detailed information on the GC process. Much of this information is also displayed in the CICS Explorer JVM Servers view, Object allocations view can be enabled by using Monitored JVM -> Garbage Collection and allocated data collection and then select Enable collection of object allocation events within and choosing low and high thresholds

Object allocations



- **Monitored System** → **Garbage Collection and allocation data collection ...**



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Use this view to identify code that is allocating large objects

Set low and high thresholds using Monitored JVM->Garbage Collection and allocation data collection

Object allocations view



Heap and pause times | **Object allocations** | Samples by request site | Samples by object

Filter by request site: Apply Clear

Count	%	Allocations	Average size (KB)	Request site
730	83.0	<div style="width: 83%;"></div>	104	com.ibm.java.diagnostics.healthcenter.agent.dataproviders.TraceSubscriberDataProvider...
54	6.14	<div style="width: 6.14%;"></div>	195	java.nio.HeapByteBuffer.<init> (HeapByteBuffer.java:57)
52	5.92	<div style="width: 5.92%;"></div>	195	com.ibm.java.diagnostics.healthcenter.agent.dataproviders.methoddictionary.MethodDict...
26	2.96	<div style="width: 2.96%;"></div>	136	java.lang.StringBuilder.ensureCapacityImpl (StringBuilder.java:342)
9	1.02	<div style="width: 1.02%;"></div>	193	com.ibm.java.diagnostics.healthcenter.agent.dataproviders.JLADDataProvider.reportJLA (N...
5	0.57	<div style="width: 0.57%;"></div>	171	java.lang.StringCoding.encode (StringCoding.java:597)
3	0.34	<div style="width: 0.34%;"></div>	136	java.lang.StringBuffer.ensureCapacityImpl (StringBuffer.java:338)

Summary | Call hierarchy | Timeline

- com.ibm.java.diagnostics.healthcenter.agent.dataproviders.TraceSubscriberDataProvider.getTraceBuffers (Native Method)
 - com.ibm.java.diagnostics.healthcenter.agent.dataproviders.TraceSubscriberDataProvider.getMXData (TraceSubscriberDataProvider.java:130) (100%)
 - com.ibm.java.diagnostics.healthcenter.agent.mbean.HealthCenter.getMXData (HealthCenter.java:111) (100%)
 - sun.reflect.GeneratedMethodAccessor23.invoke (Bytecode PC: 198) (100%)
 - sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethodAccessorImpl.java:55) (100%)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Samples by object view



Heap and pause times | Object allocations | Samples by request site | **Samples by object**

Filter class names: Apply Clear

Count	%	Total size (KB)	Allocated Object
4	23.5	0.13	java/lang/StringBuffer
3	17.6	0.12	com/ibm/rmi/util/buffer/ByteBuffer\$Position
3	17.6	0.12	java/util/HashMap\$Entry
1	5.88	0.1	com/ibm/ws/genericbnf/internal/HeaderElement
1	5.88	0.023	com/ibm/rmi/iiop/CDRReader\$PendingReadMarker
1	5.88	0.07	java/util/HashMap
1	5.88	0.031	java/lang/StringBuilder
1	5.88	0.023	java/util/HashMap\$KeySet
1	5.88	0.055	java/lang/StackTraceElement
1	5.88	0.19	com/ibm/rmi/iiop/RequestMessage

Summary | Call hierarchy | Timeline

- Calls to allocation of class java/lang/StringBuffer
 - com.ibm.ws.http.channel.internal.HttpRequestMessageImpl.getRequestURL (HttpRequestMessageImpl.java:686) (100%)
 - com.ibm.ws.http.dispatcher.internal.channel.HttpRequestImpl.getURL (HttpRequestImpl.java:143) (100%)
 - com.ibm.cics.wip.impl.CICSHttpRunnable.getURL (CICSHttpRunnable.java:419) (100%)
 - com.ibm.cics.server.internal.CICSThreadExecutor\$DefaultThreadFactory\$3.run (CICSThreadExecutor.java:571) (100%)
 - com.ibm.cics.server.internal.CICSThreadExecutor\$DefaultThreadFactory\$3.run (CICSThreadExecutor.java:534) (100%)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Use these views to identify code that is allocating large numbers of objects outside of the thread local heap.

Enable collection of call stacks to show call hierarchy Monitored JVM -> Garbage Collection and allocation data collection

Threads perspective



The screenshot displays the 'Current threads' perspective with the following components:

- Thread name filter:** A text input field with 'Apply' and 'Clear' buttons.
- Thread list:** A table with columns 'Thread name' and 'Thread state'.

Thread name	Thread state
Health Center (memory)	RUNNABLE
Health Center (threads)	RUNNABLE
Health Center (memory counters)	RUNNABLE
Health Center (locking)	RUNNABLE
MemoryPoolMXBean notification di...	RUNNABLE
Health Center (methoddictionary)	RUNNABLE
RT=0-P=371862-O=0:TCPTransport...	RUNNABLE
Finalizer thread	RUNNABLE
DFHSJILTASK76227.GENH	RUNNABLE
DFHSJILTASK76228.GENH	RUNNABLE
DFHSJILTASK76234.GENH	RUNNABLE
JMX server connection timeout 37	TIMED_WAITING
Timer-0	TIMED_WAITING
JMX server connection timeout 4804...	TIMED_WAITING
RMI Scheduler(0)	WAITING
Thread-4	WAITING
stop JMX Server on shutdown	WAITING
WT=171	WAITING
- Number of threads:** A line graph showing the number of threads (y-axis, 0.0 to 40.0) over elapsed time (x-axis, 40:00 to 55:00 minutes). The thread count fluctuates around 30.
- Thread details:** A panel for 'Owned monitor name' with a scrollable list and fields for 'Contended monitor' and 'Contended monitor owner'.
- Thread stack:** A stack view for thread 'DFHSJILTASK76227.GENH' showing the call stack:
 - com.ibm.cics.domains.DomainCall.dfhcdjni(Native Method)
 - com.ibm.cics.domains.Dfhpipmj.invoke(Dfhpipmj.java:1631)
 - com.ibm.cics.wlp.channel.program.ProgramLinkage.invokeProgramViaPIPM(ProgramLinkage.java:160)

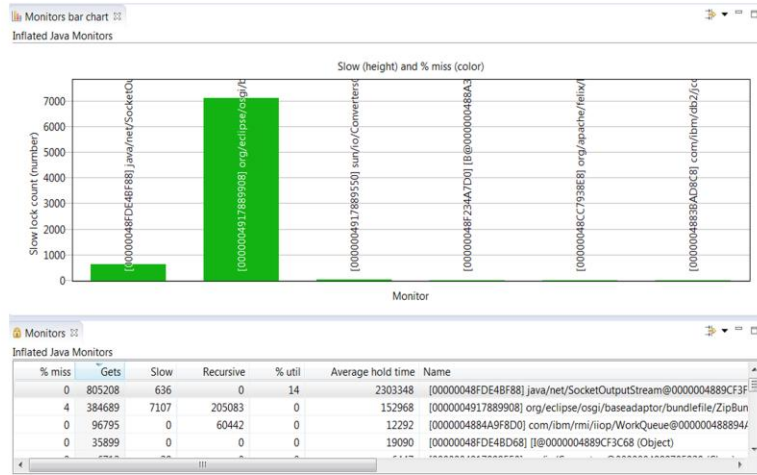
Complete your session evaluations online at www.SHARE.org/Oriando-Eval



Current threads view can be filtered

Thread stack can be used to show call stack

Locking perspective



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



The Locking perspective profiles Java lock (aka monitors in Java) usage and helps identify points of contention in the application or Java runtime environment that prevent the application from scaling.

Useful metrics are:-

- % miss:- percentage of non-recursive requests that had to wait for the lock
- Slow:- number of times a requests had to wait
- % util:- percentage of time this lock was held during the measurement interval

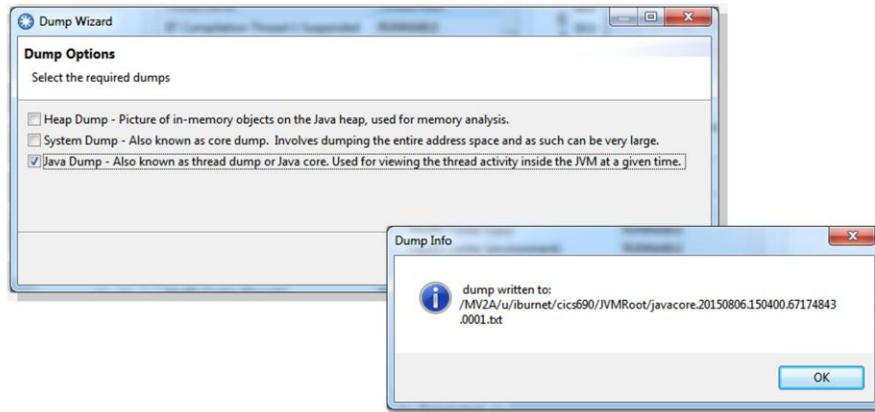
Controlling diagnostics

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Requesting a Java dump



- **Monitored System** → Request a dump ...



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Various JVM diagnostics actions can be driven from the Health Center client by using Monitored JVM

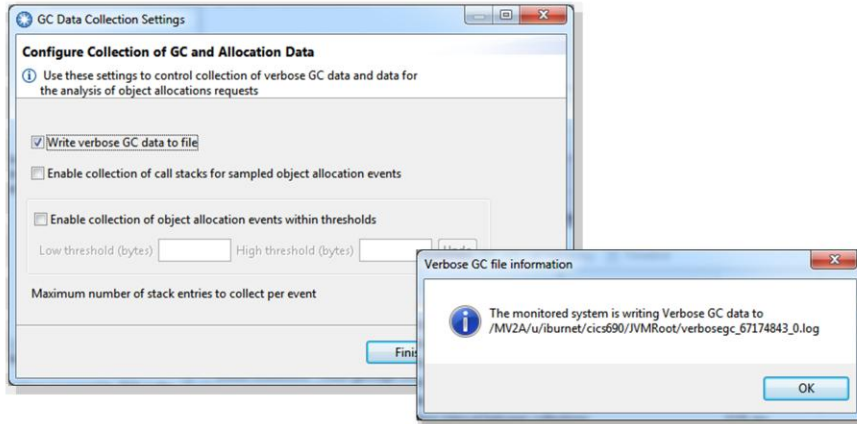
- > Request a dump to produce either Heap, System or Javacore dumps to a file
- > Garbage Collection to select verbosegc data be written to a file
- > Trace settings to enable and disable Java method tracing

Dumps and verbosegc data can be analysed with tools provided by the IBM Support Assistant

Verbose garbage collection



- **Monitored System** → **Garbage Collection and allocation data collection ...**



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



34

Bonus tips

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Saving data for later analysis

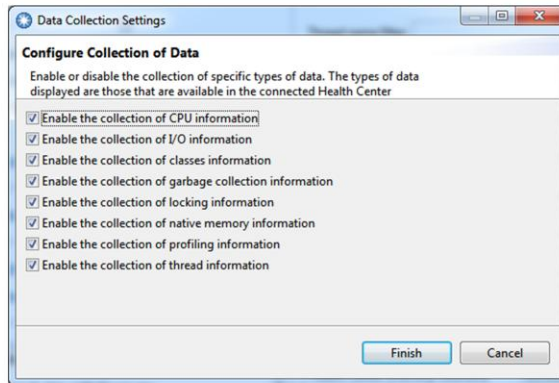


- Health Center uses temporary backing files
 - Rolling data storage onto your workstation
 - Each file maximum 2.5 GB
 - Default of maximum 1 file
 - **Window** → **Preferences** → **Health Center** → **Data Storage Settings**
 - Files are discarded when Health Center is closed
- Files can be saved
 - **File** → **Save Data ...**
 - Can be opened at a later date without a live connection

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Reducing the amount of data collected



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



If an application generates more data than Health Center can process, it is possible that Health Center might lose some data. If data loss occurs, you see a message about dropped data points in the agent connection view.

You can reduce the likelihood of losing data by turning off the collection of data from areas that you are not interested in.

To access these options, use Monitored JVM > Data Collection Settings.

Additional information sources

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

CICS Explorer and the IBM Health Center are not the only sources of Java performance information when running Java applications in CICS. In this section we look at some other sources of information available to you when investigating Java performance within CICS.

CICS statistics



JVMSERVER STATISTICS

```
JVMSERVER Name . . . . . : JSONJVM
JVMSERVER JVM profile name . . . . : DFHJVMAX
JVMSERVER LE runtime options . . . : DFHAXRO
JVMSERVER use count. . . . . : 219327
JVMSERVER thread limit . . . . . : 250
JVMSERVER current threads. . . . . : 3
JVMSERVER peak threads . . . . . : 26
JVMSERVER thread limit waits . . . : 0
JVMSERVER thread limit wait time . : 00:00:00.0000
JVMSERVER current thread waits . . : 0
JVMSERVER peak thread limit waits. : 0
JVMSERVER system thread use count. : 1
JVMSERVER system thread waits. . . : 0
JVMSERVER system thread wait time. : 00:00:00.0000
JVMSERVER current sys-thread waits.: 0
JVMSERVER peak system thread waits : 0
JVMSERVER state. . . . . : Enabled
JVMSERVER JVM creation time. . . . : 08/06/2015 14:56:11.9339
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



CICS statistics contains several important metrics which are produced for each JVMSERVER in the CICS region.

CICS statistics (contd.)



```
...
JVMSEVER current heap size . . . . : 98820888
JVMSEVER initial heap size . . . . :    400M
JVMSEVER maximum heap size . . . . :    400M
JVMSEVER peak heap size . . . . . :    385M
JVMSEVER heap occupancy . . . . . : 48924704
JVMSEVER Garbage Collection policy: -Xgcpolicy:gencon
JVMSEVER major GC collections. . . :      1
JVMSEVER elapsed time in major GC.:     25
JVMSEVER major GC heap freed . . . :   303M
JVMSEVER minor GC collections. . . :   2521
JVMSEVER elapsed time in minor GC.:    2772
JVMSEVER minor GC heap freed . . . : 300983M
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



40

... a continuation of the previous slide.

CICS monitoring data



- Many fields available in CICS monitoring data for JVMSERVER
 - All of the "regular" monitoring data
 - JVMSUSP – JVM suspend time
 - JVMTHDWT – JVM server thread wait time
 - T8CPUT – Time spent on a CICS T8 TCB
 - CPUTONCP, OFFLCPUT – Calculations of offload to specialty engine

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



All of the monitoring data which is available for non-Java CICS tasks is also available for tasks containing Java programs.

Also available are several fields which help identify where tasks are waiting, as well as understanding at a task level which tasks are benefitting / would benefit from offload to a specialty engine.

Questions?

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Session 17676

Ian Burnett
ian.burnett@uk.ibm.com

Thank you

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Reference material

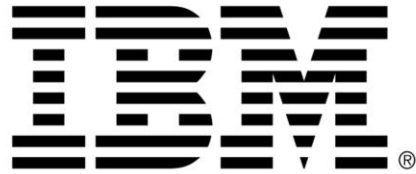


- IBM Health Center
 - ibm.com/developerworks/java/jdk/tools/healthcenter/
- IBM Support Assistant
 - ibm.com/software/support/isa/
- CICSdev blog post
 - ibm.com/developerworks/community/blogs/cicsdev/entry/hcandjvmservers
- CICS Explorer
 - ibm.com/software/products/en/cics-explorer

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



44



Complete your session evaluations online at www.SHARE.org/Orlando-Eval