



IBM Software Group

# IBM® WebSphere® Application Server V7

## *Java™ serviceability components*



@business on demand.

© 2008 IBM Corporation  
Updated September 25, 2008

WebSphere Application Server V7 runs on top of the IBM SDK for Java Version 6. The SDK is instrumented with many serviceability components to help with problem determination and performance tuning.

## Agenda

- Runtime serviceability components
- Problem determination tools



First, this presentation will focus on serviceability components that are embedded in the Java runtime, like dumps and traces and APIs for processing them. Second, the presentation will describe several problem determination tools for JVMs that are available through the IBM Support Assistant Workbench.

## Section

# *Runtime serviceability components*



This section discusses serviceability components that are packaged with the Java runtime.

## Native stack traces

- Console dumps and javacores include native stack traces

```

0SECTION      Stack backtrace for failing native thread
NULL
NULL
BTTHREADID    Native thread id: 0x00256045
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libj9vm24.so:0xd538ddb4
               [0xd5363000 +0x0002adb4]
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libfontmanager.so:0xd5e6df14
               [0xd5e29000 +0x00044f14]
1BTSTACKENT   0x3076ba18
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/j9vm/libjvm.so:0xd530edf8
               [0xd530a000 +0x00004df8]
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libjava.so:0xd53e074c
               [0xd53b1000 +0x0000f74c]
1BTSTACKENT   0x3076ab84
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/j9vm/libjvm.so:0xd530edf8
               [0xd530a000 +0x00004df8]
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libjava.so:0xd53e074c
               [0xd53b1000 +0x0000f74c]
1BTSTACKENT   0x304546c4
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libj9vm24.so:0xd537c934
               [0xd5363000 +0x00019934]
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libj9prt24.so:0xd533f8d0
               [0xd533c000 +0x000038d0]
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libj9vm24.so:0xd537dAA4
               [0xd5363000 +0x0001AAA4]
1BTSTACKENT   /data/AIX/testjava/java6-32/sdk/jre/lib/ppc/libj9thr24.so:0xd5331444
               [0xd532f000 +0x00002444]
1BTSTACKENT   /usr/lib/libpthread.s.a:0xd0111444 [0xd010e000 +0x00003444]
1BTSTACKENT   0x00000000
NULL
-----

```

New in Java 6, some dumps contain stack backtraces that provide a full stack trace of the failing native thread. You can use the stack backtrace to determine if a failure is caused by an error in the JVM or the native application. A stack backtrace is present in a console dump whenever one is generated and in Java dumps for SIGSEV, SIGILL and SIGFPE signals. The stack backtrace is platform dependent, and the information available varies by platform. Stack backtraces are not available on all platforms. The example on this slide shows a stack backtrace from a console dump on AIX. The trace provides information about the failing native thread, showing the native libraries that are being called and the sequence of function offsets in those libraries.

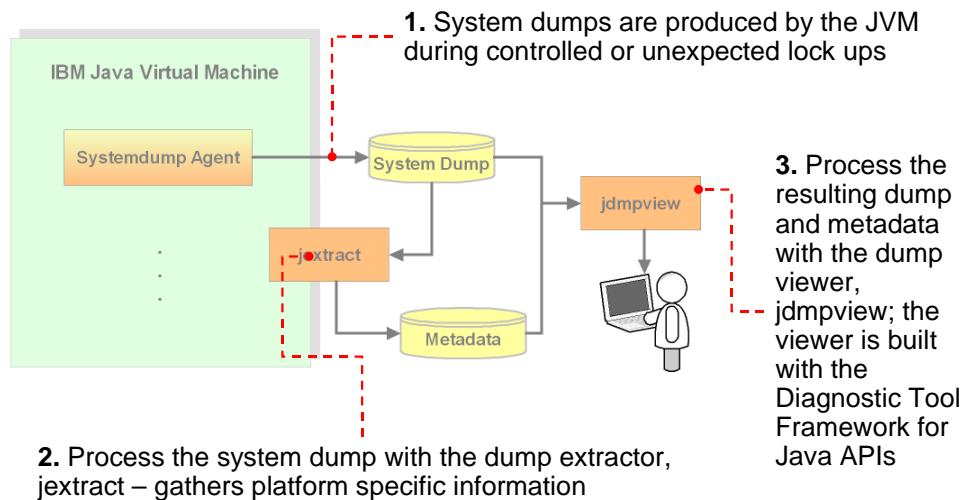
## Heap dump improvements

- Binary heap dump compression
  - ▶ On 64-bit platforms, dumps can be up to 10x larger than similar dumps in the sovereign VM
- Improved heap dump performance
  - ▶ Use compression and caching to boost performance
  - ▶ Can be up to ten times faster than in Java 6
- Heap dumps produced by default for OutOfMemory events produced in the VM
  - ▶ New systhrow dump event



In Version 5, Heapdumps were produced by default for any uncaught OutOfMemoryError event, including OutOfMemoryErrors produced in user code. The default behavior has changed in Version 6, so that Heapdumps are produced for OutOfMemoryErrors triggered by the virtual machine. The new systhrow event has been added in this release to support this new Heapdump behavior. Also new in Java 6, the Heapdump generator takes advantage of compression capabilities in the binary dump file format and uses a file writing cache scheme to produce smaller Heapdump files and write them out to disk faster than in the previous release.

## System dump processing



This diagram illustrates the system dump processing flow, using built-in components and commands in the IBM SDK. First, a system dump agent that is configured to monitor for certain events – like when a JVM has a general protection fault – will trigger a dump when those events occur. Second, before you try to analyze your system dump with any other tools, the dump file should be fed into the dump extractor, for pre-processing. The dump extractor will pull out useful metadata about your Java operating environment and create an archive file that contains the metadata and your system dump file, which you can then use in a system dump processing tool. Whenever possible, you should run the dump extractor on the system where the dump was produced. If that is not an option, you can try running the dump extractor on a machine that is running the same operating system and JVM level. Third, take the output from the dump extract and use it as input to the dump viewer, jdmpview. The dump viewer is a command-line debugger that allows you to explore the contents of your system dump. The dump viewer in Java 6 is built on the Diagnostic Tool Framework for Java APIs.

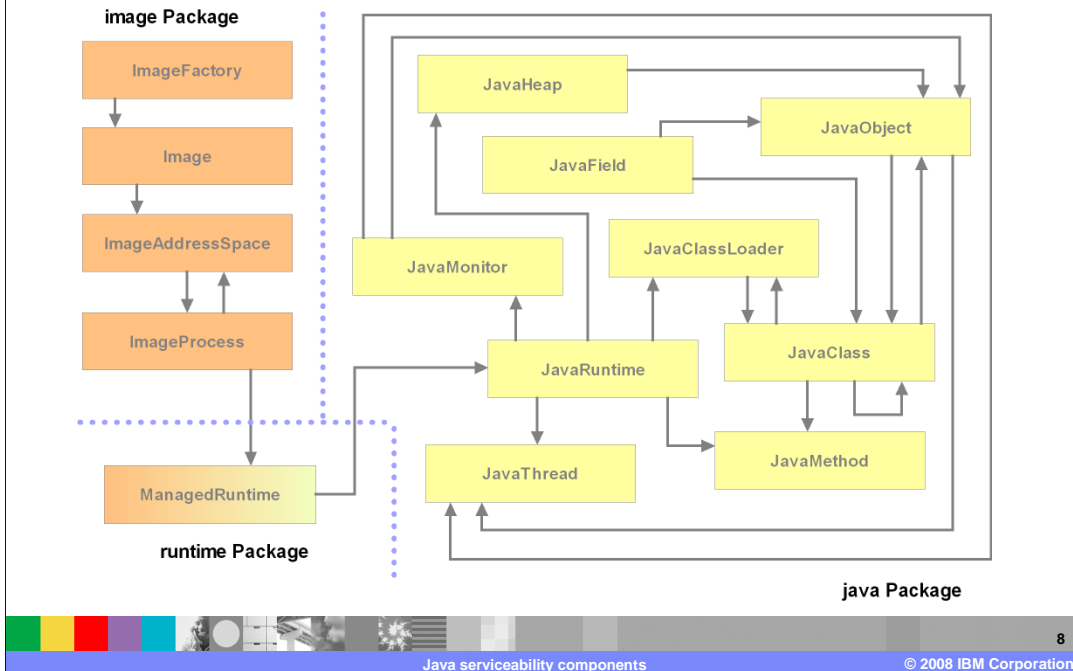
## Diagnostic Tool Framework for Java

- The Diagnostic Tool Framework for Java (DTFJ) is a Java API used to build diagnostic tools for Java programs
  - ▶ Processes system dumps and Java dumps
  - ▶ Enables writing tools without knowing exact dump structure
- The DTFJ API helps diagnostic tools access this information, and more:
  - ▶ Java classes and objects that were present in the heap
  - ▶ Relationships between memory locations and Java internals
  - ▶ Details of the machine on which the dump was produced
  - ▶ Details of the Java version that was being used
  - ▶ The command line that launched the JVM
  - ▶ Java threads and native threads
- Examples can be found in the Diagnostics Guide



The Diagnostic Tool Framework for Java acts as a layer of abstraction between a tool developer and the underlying structure of diagnostic data in the virtual machine. The DTFJ APIs allow Java tool developers to access data in a dump, like the Java version, threads, and heap data, without needing to understand the exact structure of the dump itself. DTFJ is implemented in pure Java and tools written using DTFJ can be cross-platform. Therefore, it is possible to analyze a dump taken from one machine on another (remote and more convenient) machine. To work with a system dump, the dump must first be processed by the dump extractor, jextract. The jextract tool produces metadata from the dump, which allows the internal structure of the JVM to be analyzed. If possible, you should run jextract on the system that produced the dump. If that is not possible, you can use a system that is running the same operating system and virtual machine level as the system that produced the dump. In Version 6, DTFJ support has been added for Java dumps. To work with a Java dump, no additional processing is required.

## DTFJ class diagram

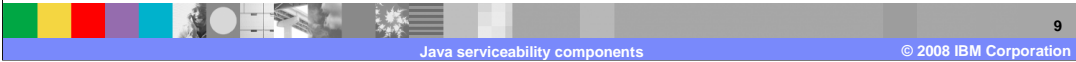


A typical DTFJ program starts on the left side of the diagram, using the `ImageFactory` to create an `Image` based on the dump being processed, and continuing to work down through the class relationships to pull out the `ManagedRuntime` associated with the `Image`. From there, you can access all of the classes in the Java package, which contain the detailed information that was a part of the dump – including information on the Java heap and the related objects, classes, methods, and fields. This is a partial class diagram of some of the components available in the Diagnostic Tool Framework for Java. A full class diagram and code examples are available in the Diagnostics Guide.



## Section

# *Problem determination tools*



This section discusses problem determination tools that are available in the IBM Support Assistant Workbench.

## IBM Support Assistant

- IBM Support Assistant is a self help and support application that helps you resolve questions and problems with IBM software products
- With IBM Support Assistant, you can
  - ▶ Personalize your workbench with product and tool add-ons
    - WebSphere Application Server V7.0 add-on is available
    - Examples of other product add-ons: WebSphere Process Server, Rational® Application Developer, Lotus® Notes®
    - Examples of tool add-ons: Dump Analyzer, Memory Dump Diagnostic for Java, Garbage Collection and Memory Visualizer
  - ▶ Access product specific documentation
  - ▶ Search for product information
  - ▶ Use problem determination tools
  - ▶ Gather and organize diagnostic data
  - ▶ Work with service requests



The IBM Support Assistant Workbench is the client-facing portion of IBM Support Assistant that you install on your workstation. The workbench can be customized with a variety of product and tool add-ons. For example, you can download product add-ons for WebSphere Application Server, Rational Application Developer, WebSphere Process Server, and many other products. Product add-ons contain references to product documentation, articles, skill-building activities, information on important fixes, and related problem determination information for the Guided Troubleshooter. You can also download tool add-ons for your workbench. The sample tools listed above are for Java problem determination activities. The Dump Analyzer, Memory Dump Diagnostic for Java, and Garbage Collection and Memory Visualizer process Java core files, memory dumps, and verbose garbage collection logs. These tools process their input files and then provide recommendations to help solve your problem or improve your application's performance. The workbench also includes an integrated Guided Troubleshooter that walks you through the problem determination process. In addition to product add-ons and problem determination tools, you can use the workbench to organize diagnostic data and work with your service requests.

## IBM Support Assistant V4 interface

The screenshot displays the IBM Support Assistant V4 interface within a Windows-style window titled "Welcome - IBM Support Assistant Workbench". The interface features a "Launch Activity" dropdown menu in the top left. The main content area is divided into three primary activity groups on the left: "Find Information" (with a magnifying glass icon), "Analyze Problem" (with a document icon), and "Manage Service Request" (with a globe icon). On the right side, there are sections for "First Steps" (Make first steps), "Tutorials" (Go through tutorials), and "Latest News" (listing various technical updates and webcasts). Red dashed boxes and arrows highlight the "Analyze Problem" section, the "Tutorials" section, and the "Latest News" section, with corresponding text boxes on the right explaining their functions.

Activity-based user interface helps you find the information you need to analyze your problems and get support

Configuration tools and task-based tutorials to help you get started

Support Assistant news feed for latest feature updates, tools and announcements

Java serviceability components © 2008 IBM Corporation 11

This is the home page of the new workbench interface, which is designed around an activity-based workflow. In the main panel on the left, the three major activity groups are listed – Find Information, Analyze Problem, and Manage Service Request. The Find Information activity is where you want to go to access the information contained in your product add-ons and to do searches. Under the Analyze Problem activity, you can directly access the new Guided Troubleshooter and load any problem determination tools that you have installed. The Manage Service Request activity allows you to interact with your service requests, including submitting new problem reports to IBM and collecting and sending diagnostic data to IBM support. You can also access these main activity areas through the Launch Activity dropdown menu that is highlighted in the upper left of the display. In the upper right, you will find a link to configuration tools under First Steps and several instructional recordings in the Tutorials section. The home page also contains a news feed that is regularly updated with feature and tool updates, announcements, and information about important fixes.

## IBM Support Assistant data collection

- Support for connecting to remote systems with an IBM Support Assistant Agent
- Data collection scripts for key JVM-related problem determination tasks
  - ▶ Trigger and collect a heap dump
  - ▶ Trigger and collect a system dump
  - ▶ Trigger and collect a javacore
  - ▶ Enable and collect verbose GC



The workbench contains a set of data collection tools that you can use to automatically collect data on local and remote systems. You can access these tools from the Analyze Problem activity in the workbench, under the Collect Data tab. First, you need to choose where to store the data that gets collected. Select the case that you want to use to organize your data. Next, choose the system that you are going to run the collector on. This can be the local system where you are running your workstation or any system that you are connected to through the agent manager. All available systems will be visible in the system dropdown menu. Next you need to choose which collector to run. You will have different collectors available depending on which add-ons you have installed. If, for example, you have the WebSphere Application Server V7 add-on installed and are experiencing a general problem, you can choose the General Problem collector to capture standard application server log files. Once you have selected a collector to run, click the Add button to add it to your collector queue. When you have added all of the collectors to your queue that you want to run, click the Collect All button to start the collectors. A status window will pop up that will keep you updated with the progress of the collectors. There are several new data collection scripts that are packaged with the WebSphere Application Server V7 add-on that are related to the JVM. For example, you can use the workbench to connect to a remote system, collect a heap dump from the application server's JVM, and pull that heap dump back down to the system where you are running the support assistant workbench.

## IBM Support Assistant tools

- Dump Analyzer (using DTFJ)
  - ▶ Analysis engine for common Java problems like crashes and OutOfMemoryErrors
- Memory Dump Diagnostic for Java
  - ▶ Analyzes heap dumps to identify leaks
- Garbage Collection and Memory Visualizer
  - ▶ Verbose GC visualization and analysis



There are three major types of diagnostic data associated with a Java Virtual Machine: system dumps – sometimes called core dumps or core files, heap dumps, and verbose garbage collection logs. There are graphical tools available as add-ons to the IBM Support Assistant Workbench that are designed to process this diagnostic data. The Dump Analyzer is a tool that processes system dumps and does automatic analysis to try to determine the cause of common Java problems, like crashes and hangs. There are special Dump Analyzer modules available to help with processing dumps from WebSphere Application Server. The Memory Dump Diagnostic for Java analyzes heap dumps and helps identify likely causes of memory leaks. The Garbage Collection and Memory Visualizer is a tool that processes verbose garbage collection logs, providing line plot visualization of GC characteristics and application tuning recommendations.

## GC and Memory Visualizer overview

- The Garbage Collection and Memory Visualizer is a graphical tool for displaying garbage collection data
  - ▶ Available as an add-on for IBM Support Assistant
  - ▶ Processes verbose garbage collection logs (optthruput, optavgpause, and gencon GC policies) and `-xtgc:freelist` trace output
  - ▶ Works with all IBM runtime environments for Java
- The visualizer provides
  - ▶ Line plots to visualize a variety of GC data characteristics
  - ▶ Reports with GC tuning recommendations
  - ▶ View of multiple datasets on a single set of axes
  - ▶ Ability to save data as an image (jpeg) or comma separated file (csv)



The GC and Memory Visualizer allows you to visualize your garbage collection data, as raw datasets, line plots, reports, and images. The “Line plot” portion of the tool displays graphs of over forty different garbage collection data characteristics – including used heap, pause times, and the reason for garbage collection being triggered. You can read in multiple sets of garbage collection logs and display them together on a single set of axes, which allows you to easily compare garbage collection behavior across multiple test runs of an application. The “Report” section of the tool contains a summary of the line plot data, along with information on general garbage collection behavior and heap sizing recommendations based on overall heap occupancy.

## Plotting data with the visualizer

Use **File > Open** to open a new input file

Use **File > Compare** to add multiple input files to a single data set for comparison and aggregated display

Right-click on the plot and use the pop-up menu to export data

The **VGC Data** menu allows you to choose what data to display

The **Axes** panel supports customized units and pan-and-zoom

The **Line plot** tab contains the data visualization

Java serviceability components

© 2008 IBM Corporation

15

This is an example of what the GC and Memory Visualizer looks like in action. This graph shows the GC cycle pause times for three test runs of the same application, all using the generational concurrent garbage collection policy. All of the plots for your dataset are displayed in the Line plot tab. To read in a new file, use the File > Open menu in the upper right of the workbench. If you want to load in multiple input files to compare on a single set of axes, use the File > Compare File... option to add more files to your data set. There are over forty data characteristics that you can display in your line plot; these can be controlled in the VGC Data menu. If you want to save a copy of the line plot, right-click anywhere on the plot, and you will see a pop-up menu that allows you to save the plot as an image file.

## Section

# *Summary and references*



This section contains a summary and references.



## Summary

- Runtime components support Java serviceability
  - ▶ Native stack traces
  - ▶ Heap dump improvements
  - ▶ Diagnostic Tool Framework for Java
- Serviceability tools are available through IBM Support Assistant



The IBM SDK for Java includes a broad range of built-in diagnostic and serviceability components, many of which have been enhanced in the Version 6 release. New native stack backtraces are available in console dumps and Java dumps. These stack backtraces provide useful information for debugging failures in native threads. The default conditions for producing Heapdumps has been changed to trigger when memory exhaustion is reported by the virtual machine. Heap dump files are also more compressed and can be written out to file faster than in the previous release. The Diagnostic Tool Framework for Java has been enhanced to provide support for Java dumps, in addition to the support that was previously in place for processing system dumps. The system dump viewer that is packaged with the SDK is built on these DTFJ APIs. Serviceability tools for processing common JVM diagnostic data and log files – like heap dumps and garbage collection logs – are available as add-ons to the IBM Support Assistant Workbench.

## References

- **Diagnostics Guide**  
<http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/index.jsp>
- **IBM Support Assistant**  
<http://www.ibm.com/software/support/isa/>

## Feedback

### Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_WASv7\\_JavaRAS.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WASv7_JavaRAS.ppt)

This module is also available in PDF format at: [..\\WASv7\\_JavaRAS.pdf](..\\WASv7_JavaRAS.pdf)



You can help improve the quality of IBM Education Assistant content by providing feedback.

## Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM                      WebSphere

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon 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 throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.