IBM Software Group

Using the Dump Analyzer tool to diagnose JVM and WebSphere problems: new features and demo

Daniel Julin - IBM WebSphere Serviceability Team

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS

Outline

- Dump Analyzer tool - Overview
- Using the Dump Analyzer tool in IBM® Support Assistant
  - DEMO
- Typical usage
- Summary

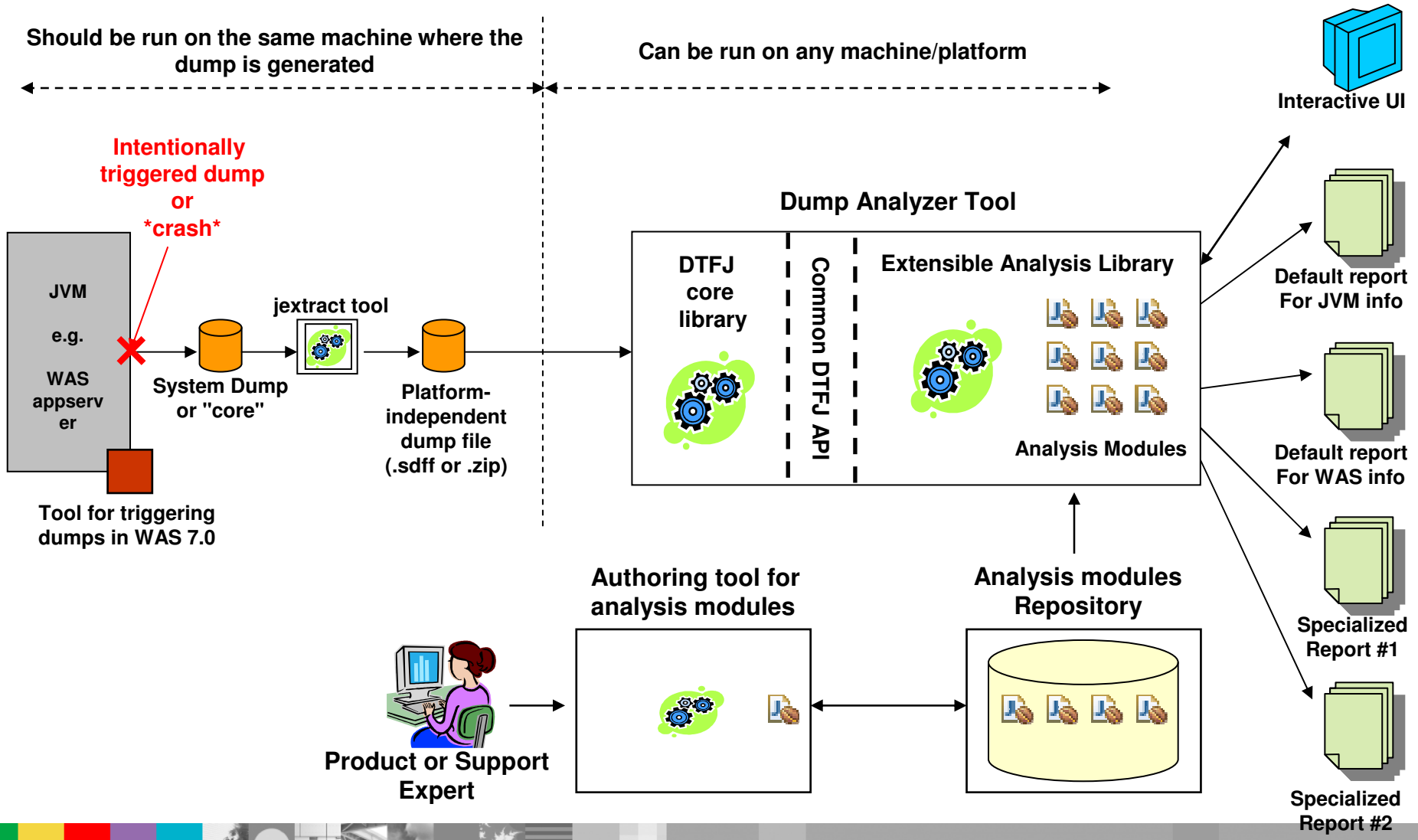
Outline

- Dump Analyzer tool - Overview
- Using the Dump Analyzer tool in IBM Support Assistant
  - DEMO
- Typical usage
- Summary

Dump Analyzer

- Full name:
  IBM Diagnostic and Monitoring Tools for Java™ - Dump Analyzer
- Based on DTFJ Technology
  - DTFJ = *Diagnostic Tooling Framework for Java*
  - A set of libraries and a common API for extracting information from a JVM™
  - dump
  - Initially System Dumps (core files), eventually other types of dumps
- A common tool / platform for examining dumps
  - Convergence of the earlier DTFJ-DumpReporter tool and new tooling efforts by the JTC and WebSphere Serviceability teams; developed in collaboration.
- Aim to automate the diagnostic as much as possible
  - Try to provide brief, simple and specific information about the particular problem found, and no extraneous information
  - Allow access to additional information and functions when the above is not possible
- Extensible, grow over time
  - JVM-specific problems, WebSphere® problems, … other products, …

Dump Analyzer Tool - Architecture



Why is it useful to analyze dumps

- Problem Determination has always relied on two complementary techniques
  - Log and Trace
  - State Dumps

- Log and Trace
  - Provides a timeline of events leading to a problem
  - If the right log/trace is available and enabled, may point to the problem directly, and sometimes flag it as soon as it happens
  - But requires instrumentation in the system being diagnosed, and that instrumentation must be enabled ahead of time
  - Performance overhead even before the problem occurs; sometimes high volume

- State dump
  - Provides a “point in time” snapshot of the state of the system
  - Sometimes difficult to infer how we reached that state
  - Requires minimal instrumentation in the system being diagnosed – “post-mortem” analysis only
    - Important for dealing with third-party libraries, fast-evolving code base
  - Minimal or no performance overhead until a problem occurs

Dump Analyzer Functions

- An extensible collection of analyzer modules
  - Each module performs one particular analysis on the dump, checks for specific issues, and produces output in the form of a report
  - Small modules can be composed to construct more complex modules
  - New modules are being written continuously
  - Some modules are “officially supported”; others are provided as-is (to encourage fast development and release)
  - End-users (including Customers) can write their own modules and add them to the tool
- An interactive shell
  - Provides a full Jython interpreter, with special commands to access dump information
  - To manually examine objects for which no specific module is yet available

Dump Analyzer Availability

- In IBM Support Assistant V3
  - First release in April 2007; roughly quarterly updates since
  - Relatively “bare-bones” user interface
  - Mostly JVM-focused analysis modules; some WAS-focused modules
- In IBM Support Assistant V4
  - First release April 2008
  - Powerful new user interface
    - Integration with ISA V4 artifact management
    - Explicit selection of analysis modules
    - Dynamically add new analysis modules
    - Support for interactive shell (Jython)
  - Start a substantial collection of WAS-focused modules
  - Available in two “flavors”
    - JVM modules only
    - Add-on for WAS modules
    - … future: add-ons for other products

Background: IBM Support Assistant

- IBM Support Assistant (ISA) is a free tool that serves as a central place to organize the resources and tools that you need to resolve questions and problems with many IBM software products
  - Download from http://www.ibm.com/software/support/isa/ or find on your product CD
  - Over a hundred “plugins” available for various IBM products

- Key features:
  - Search and browse support-related information about each product
  - Collect and organize diagnostic data (logs, traces, etc.) – locally and remotely (ISAv4)
  - Find and use Problem Determination tools
  - Receive step-by-step guidance for troubleshooting (IBM Guided Activity Assistant)
  - Open and manage PMRs

- Currently available in two major versions
  - IBM Support Assistant V3 or V3.1 – since 2006
  - IBM Support Assistant V4 - new release in March 2008
    - Major update, including
      - New Eclipse (SWT) workbench with multiple views and multiple applications
      - Local workbench and remote agents for data collection

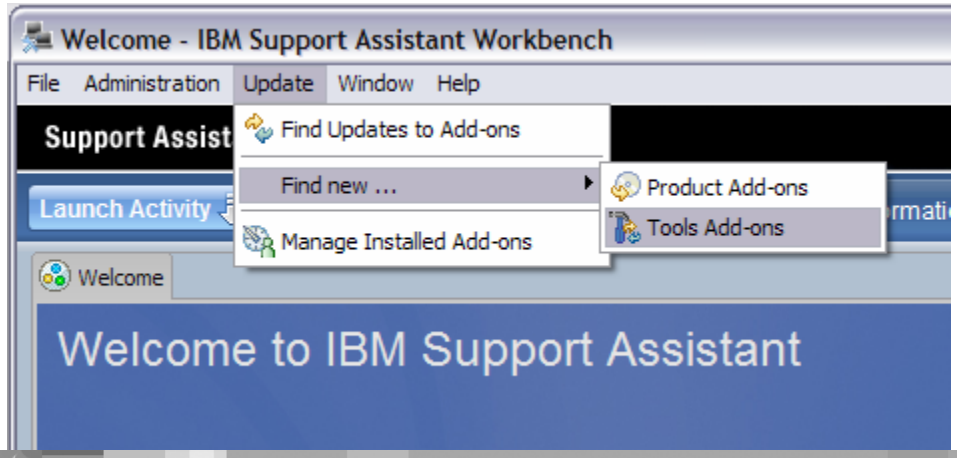
Outline

- Dump Analyzer tool - Overview
- Using the Dump Analyzer tool in IBM Support Assistant
  - DEMO
    - following charts / screenshots provided for reference only
- Typical usage
- Summary

IBM Support Assistant V4

Installing Dump Analyzer in ISA V4

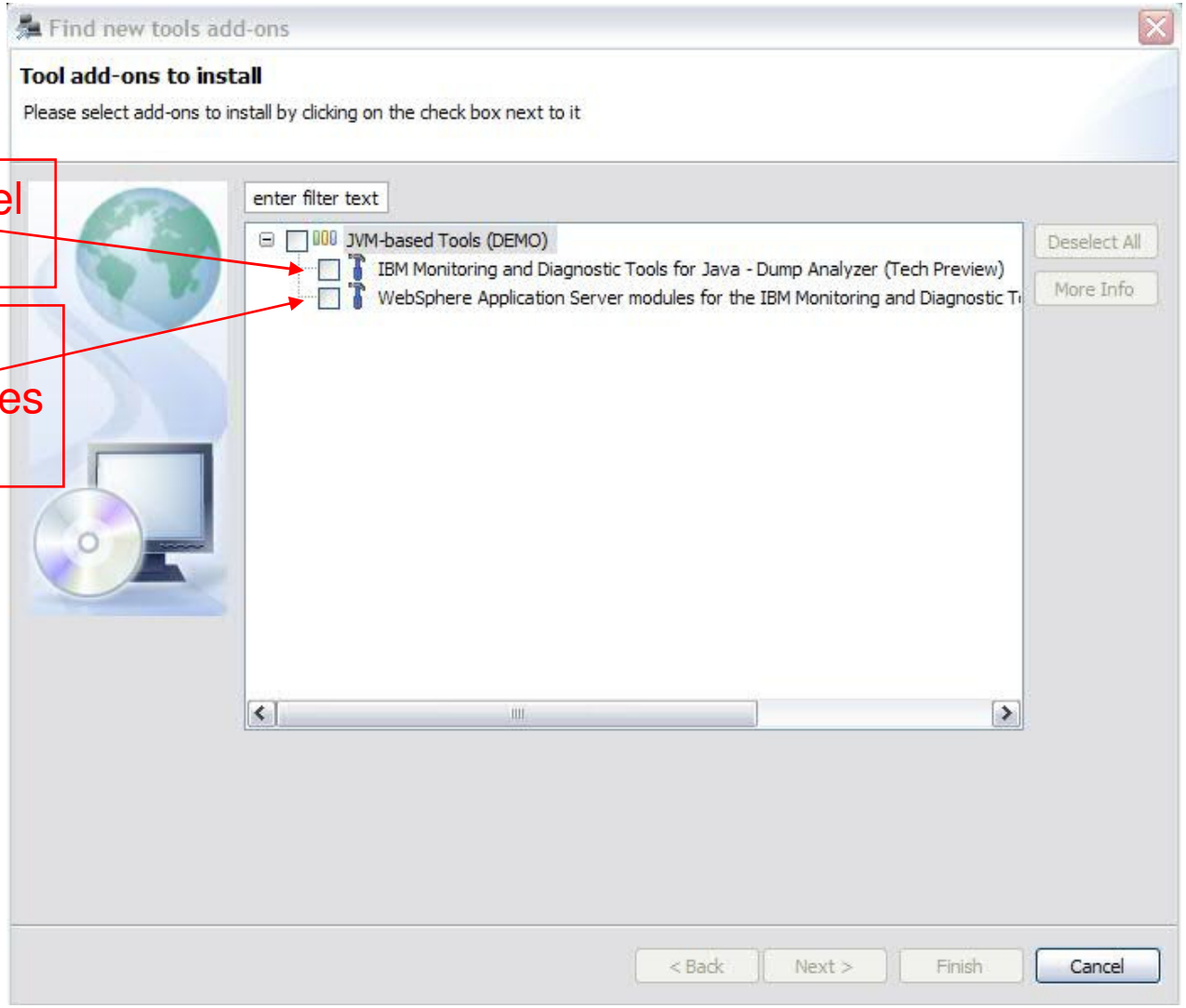
- Install ISA V4
  - Download from http://www.ibm.com/software/support/isa/
- Go to “Update -> Find new… -> Tools Add-ons”
- Select the Add-ons for Dump Analyzer
  - IBM Monitoring and Diagnostic Tools for Java - Dump Analyzer
  - WebSphere Application Server modules for the IBM Monitoring and Diagnostic Tools for Java - Dump Analyzer
- Click “Next”; review and accept licenses; click “Finish”



Installing Dump Analyzer in ISA V4

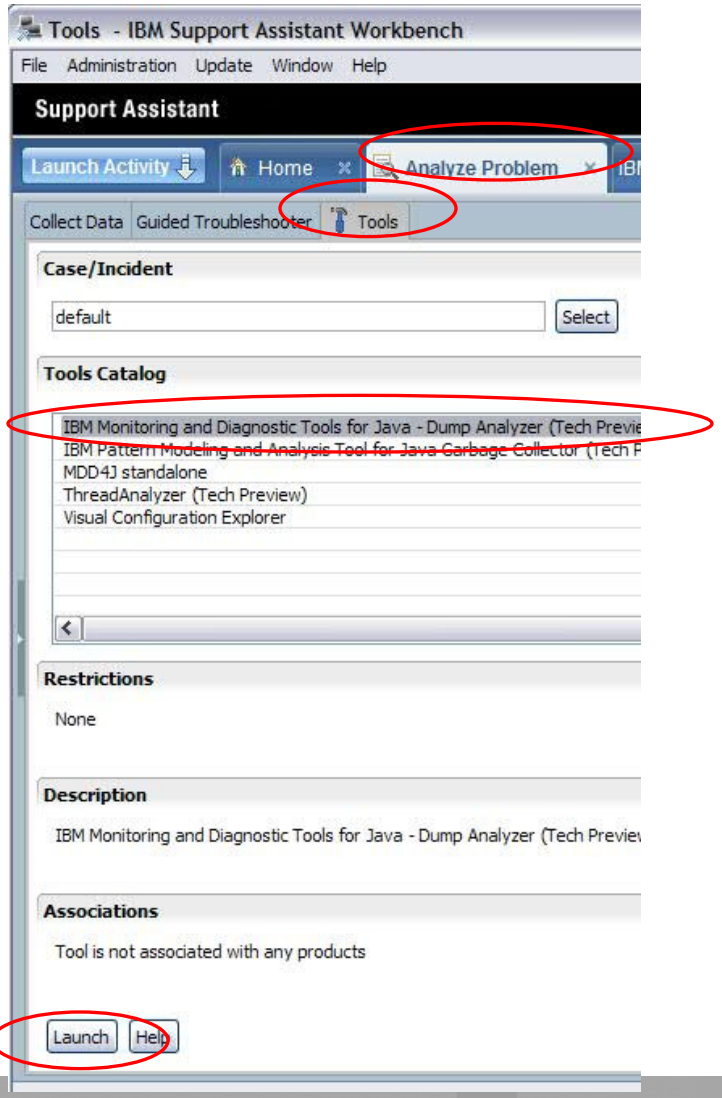
Base tool + JVM-level analysis modules

General-purpose WAS analysis modules + Jython shell



Launching Dump Analyzer in ISA V4

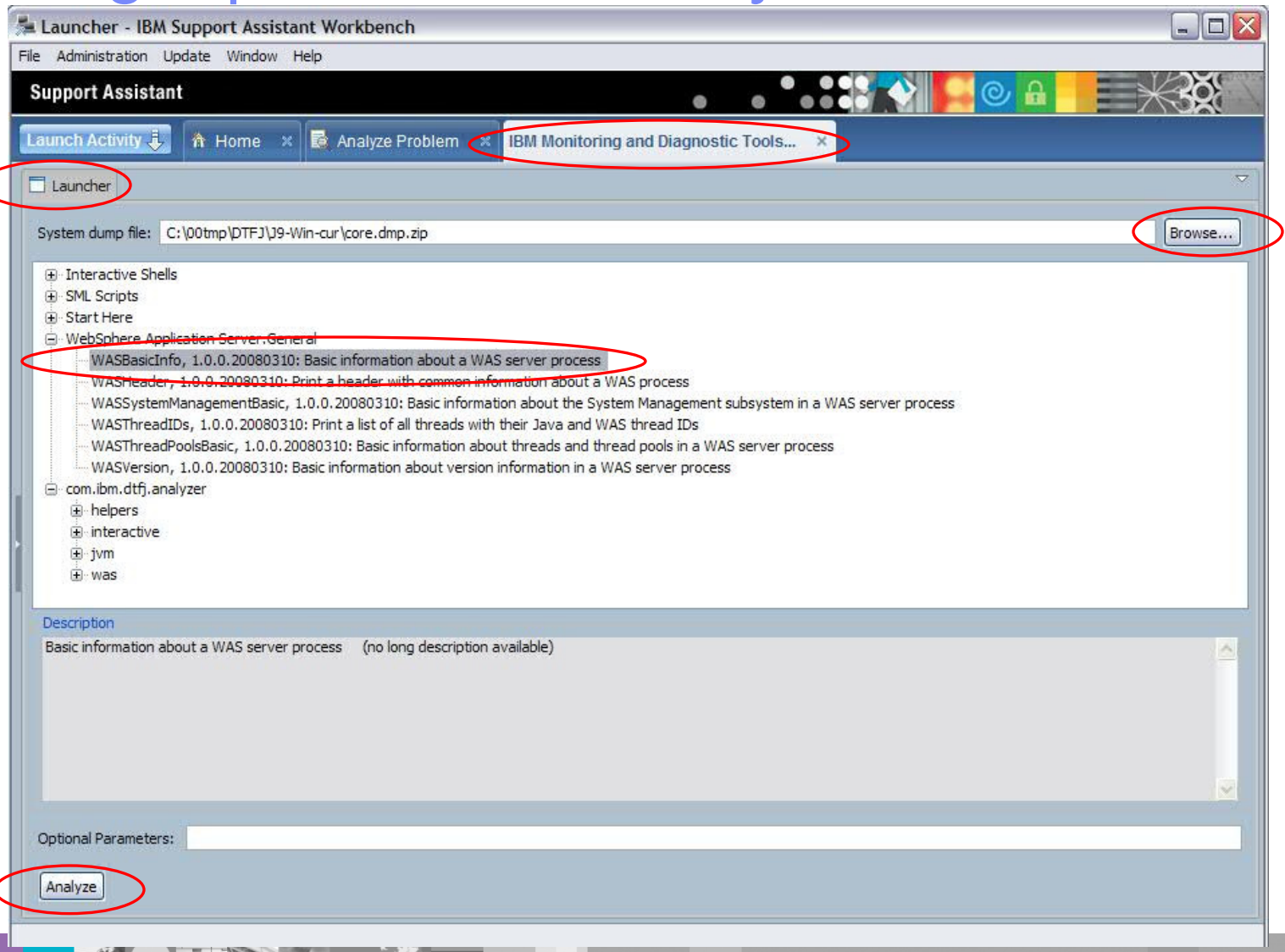
- Go to the ISA V4 “Analyze Problem -> Tools” panel
- Select the “Dump Analyzer” entry in the Tools Catalog
- Click “Launch”
- Note: in ISA V3, you needed to first select a product before selecting a tool. This is no longer necessary in ISA V4



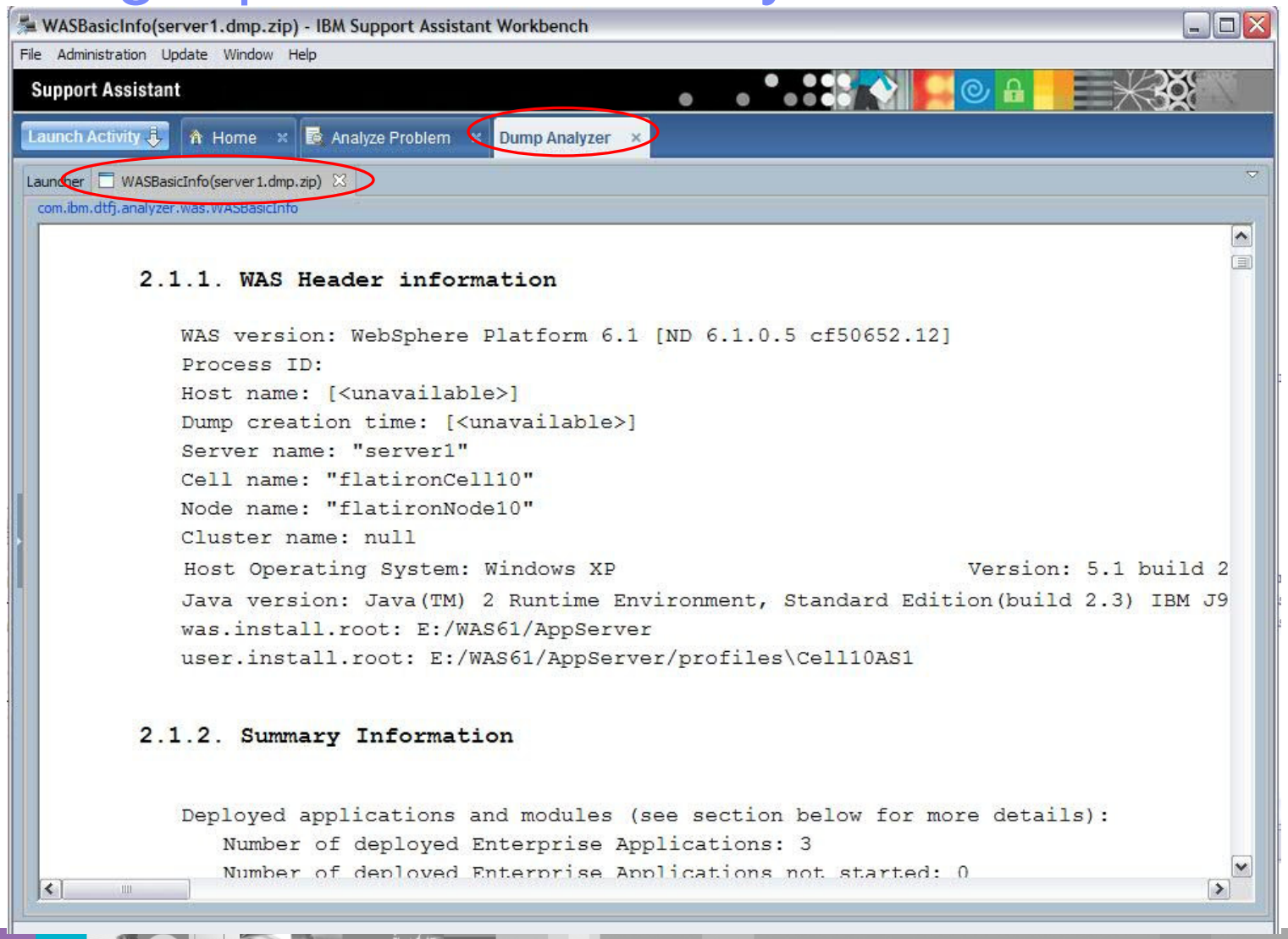
Running a pre-defined analyzer in ISA V4

- Pick a dump file from the artifact browser
- Pick an analyzer module
  - Analyzers are organized by categories
  - Main category reflects the package name of the analyzer class
- Click “Analyze”
- Pick another analyzer module, run a new analysis
  - Each analysis report appears in a different view

Running a pre-defined analyzer in ISA V4



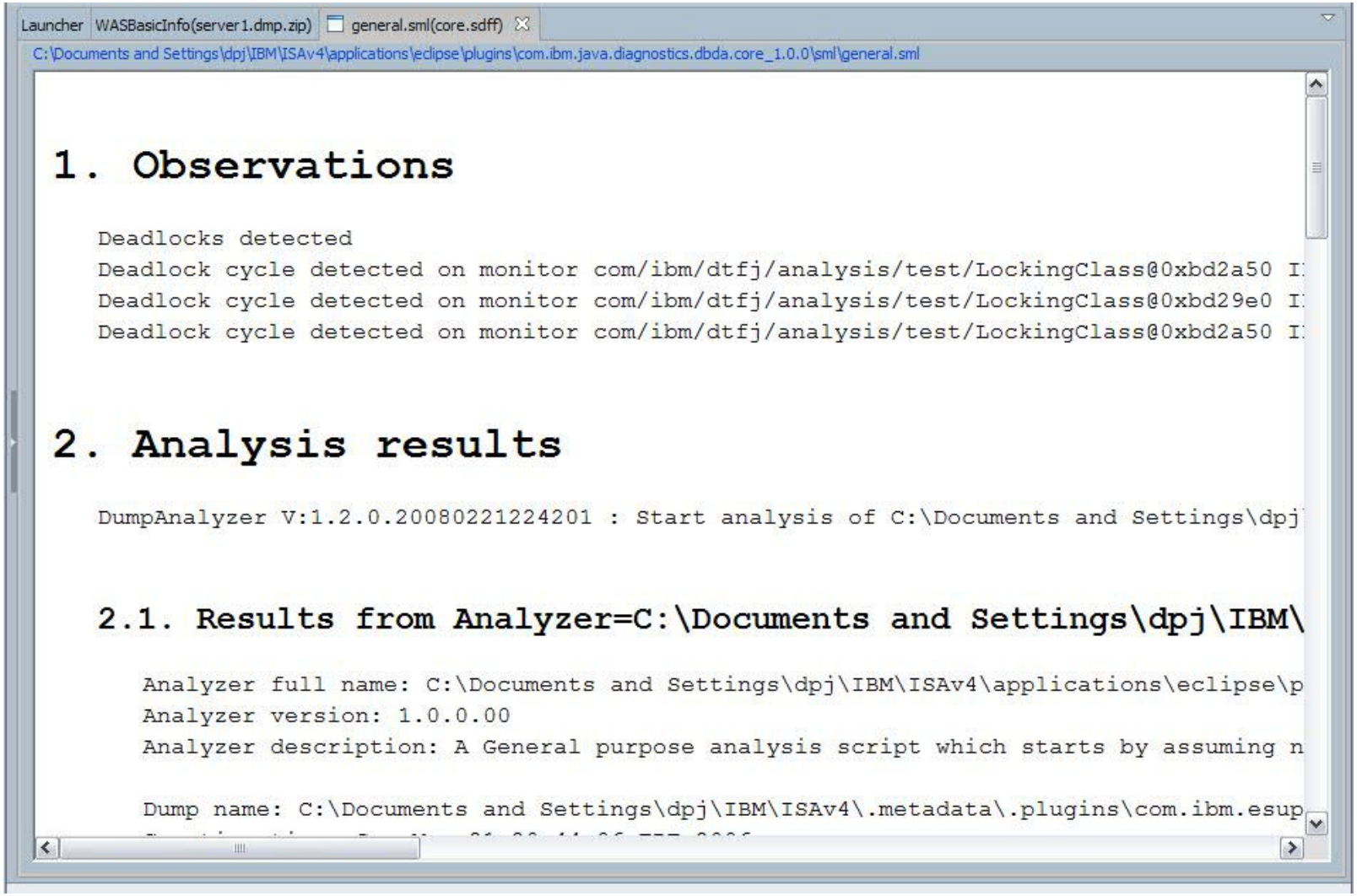
Running a pre-defined analyzer in ISA V4



Structure of an analysis report

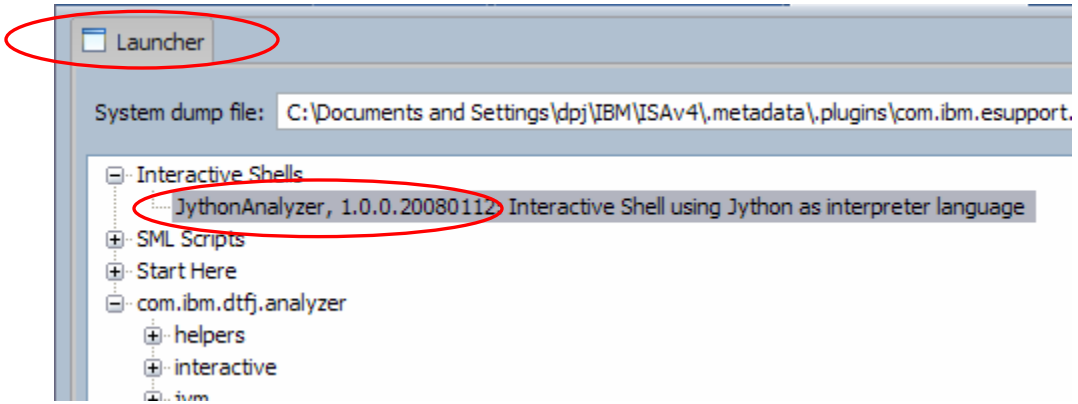
- Observations
  - Major findings from the analyzer module, that indicate potential problems detected in the dump
  - E.g. "*deadlock detected*"; "*the authentication cache is overflowing*"; …
  - Hard to reliably and automatically detect problems, while avoiding false alarms – this section is often empty
- Analysis Results
  - All the information produced by the analyzer, for examination by an expert
  - Contents depend on each analyzer; may be very short or very detailed
  - Most analyzers subdivide their output between a Summary section and a Details section
- Errors
  - Errors that prevented the analyzer from doing a perfect job
  - E.g. partial corruptions or in-flight inconsistencies in the dump; mismatches between the dump version and the analyzer; bugs in the analyzer
  - Every analysis typically has a few errors; evaluate them to assess if the analysis results are reliable

Structure of an analysis report - example

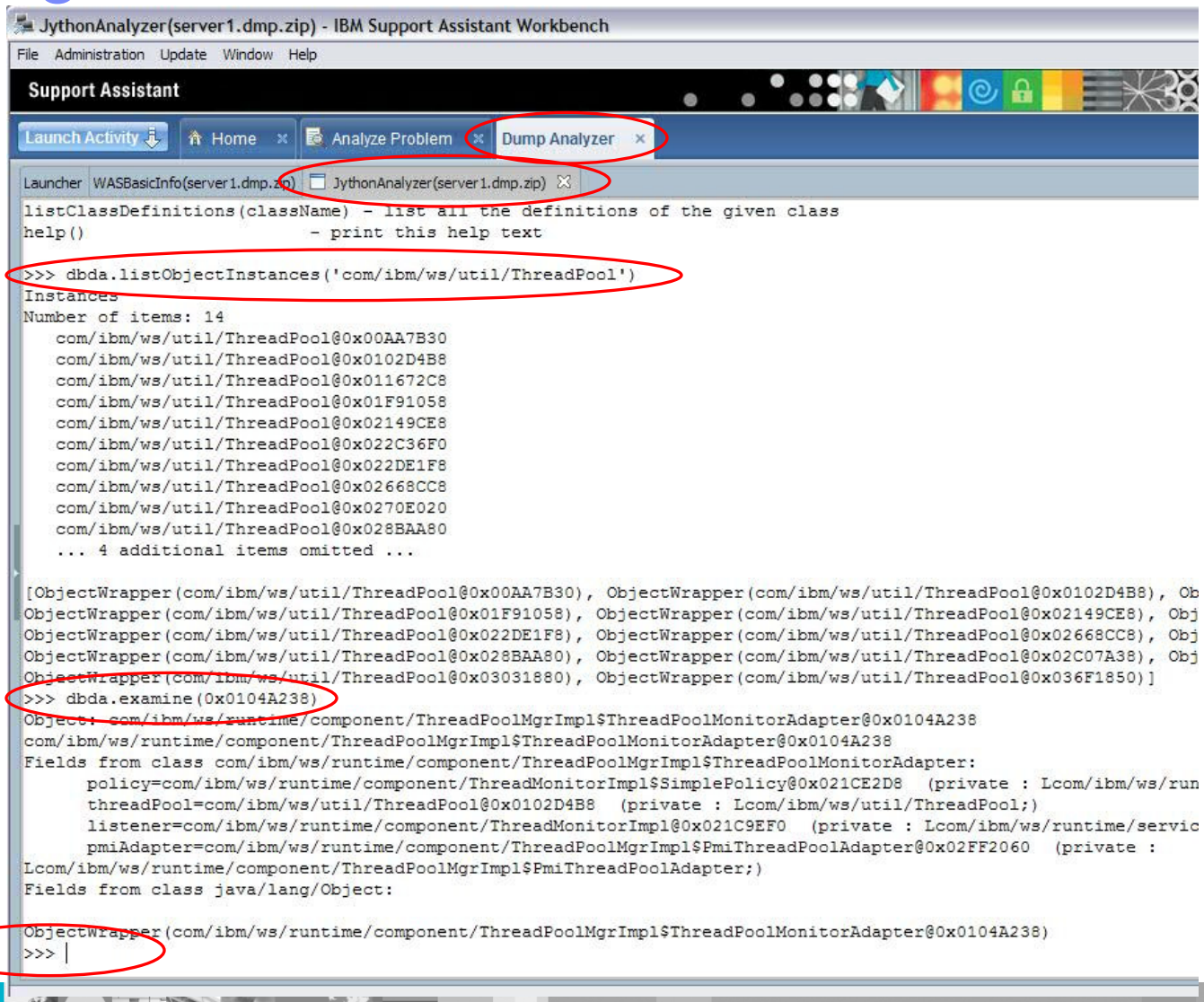


Running the interactive shell

- Pick a dump file from the artifact browser
- Pick the "JythonAnalyzer" analyzer module
- Click "Analyze"
- The interactive shell appears in a new view
- Type "*DumpAnalyzer.help()*" for a list of commands
  - Can be abbreviated to "*dbda.help()*"
- Typical commands
  - *dbda.listObjectInstances()* – list all instances of a given class
  - *dbda.examine()* – print the contents of the Java object at a given address
  - *dbda.runAnalyzer()* – execute a given analyzer module on the current dump

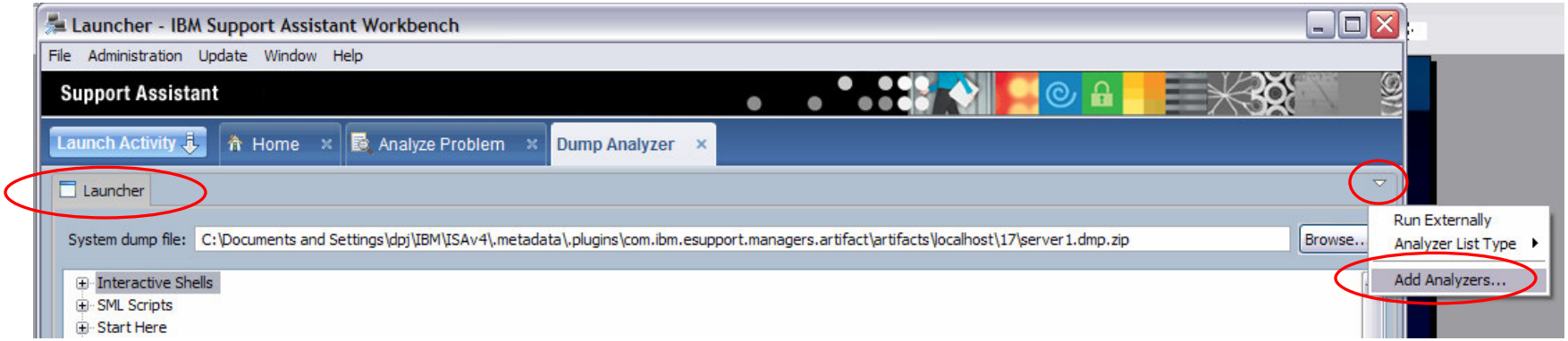


Running the interactive shell



Writing and using a new analyzer module

- New analyzer modules can be written as
  - A SML script
  - A Jython script
  - A Java class
- A tutorial and javadoc are provided with the tool documentation
- To invoke a new analyzer module
  - Package it in a jar file
  - Use the “Add Analyzers” menu entry in the main Dump Analyzer launch screen
  - The new analyzers appear in the menu of available analyzers, alongside all other (pre-defined) analyzers



Outline

- Dump Analyzer tool - Overview
- Using the Dump Analyzer tool in IBM Support Assistant
  - DEMO
- Typical usage
- Summary

Typical Usage

- Start with an initial high-level look through the dump, using one of several high-level analyzer modules
  - general.sml – attempt to automatically detect JVM-level problems (e.g. deadlock, out-of-memory, crash, …)
  - WASBasicInfo – attempt to automatically detect WAS-level problems; provide overview of WAS state
- Use one of the specialized, “drill-down” analyzer modules to get in-depth information about one particular area
  - JVM-level: DefaultDumpReport, NativeMemAnalysis, CountSockets, …
  - WAS-level: WASRuntimeBasic, WASVersion, WASThreadIDs, …
- Use the interactive shell to manually examine objects for which no pre-defined analyzer is available
  - Provides a full Jython interpreter, with special commands to access dump information
- Write your own custom analyzer module
  - As a Jython script, a SML script, or a Java class

Typical Usage – Who does the work?

- Three alternatives:
  - Perform the analysis yourself (at Customer site) and examine the results
  - Send the raw dump to IBM Support, let them do the analysis
    - Note: the raw dump may contain sensitive information
  - Run some pre-defined analyzer modules at IBM Support's request, send the report output to IBM
    - The report does not contain sensitive information, and can be easily reviewed to confirm it
    - May need multiple rounds of sending specialized reports, as we get closer to the problem in each round

What analyzers are available?

- JVM-level problems
  - General-purpose
    - general.sml
    - DefaultDumpReport
    - HighLevelSummary
  - Specialized analyzers
    - NativeMemAnalysis
    - SystemProperties
    - ListZipJars
    - AnalyzeFinalizers
    - AnalyzeReferences
    - …
- WAS-level problems
  - General-purpose
    - WASBasicInfo
  - Specialized analyzers
    - WASThreadIDs
    - WASVersion
    - WASThreadPoolsBasic
    - WASRuntimeBasic
    - WASSystemManagementBasic
    - …
- Note: analyzers are “best-effort” on older WAS versions

Example: DefaultDumpReport

- Basic information about the JVM process
  - Processor type, process ID, command line, JVM version, etc.
- JVM initialization arguments
  - System classpath, heap tuning parameters, etc.
- Environment variables
- Native libraries loaded in this process
- Memory areas in this process (optional)
- Threads (both Java threads and native threads)
  - Java thread ID, WAS thread ID, java.lang.Thread object, priority, etc.
  - Java stack, native stack
  - Registers (optional)
- Monitors (both Java and native) (optional)
- Heaps
  - Heap memory layout
  - List of objects on the heap (optional)
- Compiled Java methods (optional)
- Classloaders (optional)
  - List of loaded and cached classes

Example: WASBasicInfo

- Observations
  - A few check for known anomalies
  - E.g. hung thread detection, large caches
- Common WAS “header” information common to all WAS-level reports
  - WAS version, server name, node name, install root, etc.
  - Similar to the information provided at the start of a WAS trace file
- Summary information
  - A few statistics for a “at a glance” view of what’s in this JVM, from a WAS perspective
  - Mostly counters, intended for comparison with other dumps, and to flag large counts
  - Deployed applications, threads and thread pools, security subsystem, web container, EJB container, transactions, JMS objects, connections, system management, …
- Detailed information on a few selected summary sections
  - List of deployed applications, modules
  - List of known servers in the cell
  - List of all thread pools and threads
- Notes:
  - The WASBasicInfo analyzer is intentionally very generic, to work as best as possible with any WAS version
  - We will continue to refine this analyzer to select the most relevant information, based on usage experience

Which types of dumps can we analyze

- Currently, DTFJ / Dump Analyzer can only work with *System Dumps*
  - e.g. core file (UNIX/Linux), process dump (Windows), svcdump (z/OS)
- May extend to other java artifacts over time, using DTFJ technology
  - e.g. thread dump / javacore, heap dump
- Consider extending to other sources of state data in the future
  - e.g. WebSphere Diagnostic Providers, First Failure Data Capture (FFDC) dump, Performance Monitoring Infrastructure (PMI), etc.
- System Dumps are the most expensive, but provide the most complete information

Where is DTFJ supported?

- jextract + the main DTFJ runtime library for reading system dumps are now shipped and supported with the standard IBM JDK
  - IBM JDK 1.4.2 SR4 and beyond ("Sovereign") -> WAS 5.1, 6.0
  - IBM JDK 1.4.2 SR4 for 64-bit platforms ("J9") -> WAS 6.0.2
  - IBM JDK 5.0 SR1 and beyond ("J9") -> WAS 6.1
  - IBM JDK 6.0 and beyond (“J9”) -> … WAS 7.0 (not yet GA)
  - On all IBM JDK platforms: AIX, Linux, Windows, z/OS, iSeries
    - including 32-bit and 64-bit
  - Fully supported, available to Customers
- End-user tools must be obtained separately
  - E.g. Dump Analyzer on IBM Support Assistant
  - Tools do not need to run on the same platform as where the dump was generated
- No support for non-IBM JDKs (e.g. Sun, HP) … YET?
  - Formalization as Java standard under consideration

Typical problems encountered when using Dump Analyzer

- Truncated or corrupted dump file
  - see Technotes with keywords like “truncated core” for instructions
  - We are working to make DTFJ better able to deal with partially-corrupted dumps
- Attempting to analyze an unsupported file format, or not processed by jextract
  - currently, Dump Analyzer can *only* work on system dumps (core, *.dmp)
  - these dumps *must* be processed by jextract (yielding a .sdff or .zip)
- The tool may take a long time to run
  - run time depends on the type of analysis and the size of the dump
  - tens of minutes not unusual for large WAS dumps and complex analysis
  - JDK 1.4.2 dumps sometimes take longer than JDK 5.0 or 6.0 dumps
  - tip: use “HighLevelSummary” or “DefaultDumpReport_Fast” for a quick scan
- Some information is missing from the report (*[<null>], [<unavailable>],* …)
  - depending on how the dump was generated, some information may simply not be present in the dump at all
- Need a module that performs a different analysis than those provided
  - let us know
  - volunteer to write some new analyzers

Outline

- Dump Analyzer tool - Overview
- Using the Dump Analyzer tool in IBM Support Assistant
  - DEMO
- Typical usage
- Summary

Summary

- The DTFJ technology provides a platform to write new tools for problem determination in Java processes
- The Dump Analyzer is available today in IBM Support Assistant
- Expect DTFJ-based tools to expand rapidly, with new modules to cover new problem types and new problem domains

For more information

- For information about the DTFJ API:
  - IBM JDK Diagnostics Guide: http://www.ibm.com/developerworks/java/jdk/diagnosis/
  - Javadoc included with the IBM JDK and with the Dump Analyzer tool documentation in IBM Support Assistant
- For information about how to enable system dumps in the IBM JVM and running jextract
  - IBM JDK Diagnostics Guide: http://www.ibm.com/developerworks/java/jdk/diagnosis/
- For information about IBM Support Assistant and tools inside IBM Support Assistant
  - Download ISA: http://www.ibm.com/software/support/isa/
  - See documentation for individual tools inside the IBM Support Assistant itself
- For information about Dump Analyzer
  - See documentation inside IBM Support Assistant
  - “Java diagnostics, IBM style, Part 1: Introducing the IBM Diagnostic and Monitoring Tools for Java - Dump Analyzer”: http://www.ibm.com/developerworks/java/library/j-ibmtools1/
  - “Java diagnostics, IBM style, Part 4: Extending the IBM Diagnostic and Monitoring Tools for Java - Dump Analyzer with analysis modules”: http://www.ibm.com/developerworks/java/library/j-ibmtools4/

Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Learn about other upcoming webcasts, conferences and events: http://www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: http://www.websphere.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a Flash replay with step-by-step instructions for using the Electronic Service Request (ESR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My support emails: http://www.ibm.com/software/support/einfo.html

Questions and Answers