

---

# WebSphere Application Server Version 8.0

## Java SDK enhancements



This presentation covers WebSphere® Application Server Version 8.0 Java™ SDK enhancements.

---

## Table of contents

- Install and service changes
- Java SDK configuration changes
- Java SDK garbage collector policy changes
- Java SDK heap dump changes

This module covers installation and service changes, configuration changes, garbage collector policy changes, and heap dump changes for the Java SDK.

## Installation and service changes (1 of 2)

- Installation and service of the Java SDK is now handled by IBM Installation Manager
  - Except for IBM i

WebSphere Application Server Version 8.0 uses a new installer called IBM Installation Manager. This installer is common across an increasing set of IBM products and it is used not only for installation, but for service as well.

All platforms will use this to service the application server and the Java SDK except for IBM i, which uses the operating system installed Java SDKs.

## Installation and service changes (2 of 2)

- Service of Java SDK levels is synchronized with application server service levels
  - Except for IBM i

Service of the Java SDK will occur as part of service of the WebSphere Application Server product. It is no longer handled separately, as has been the case in the past.

See the WebSphere Application Server Version 8.0 installation education module for further details.

## Java 6 R2.6 SDK configuration changes (1 of 2)

- The default garbage collection policy for the stand-alone Java SDK is now the Generational Garbage Collector
- Stand-alone Java SDK now defaults to using a persistent shared class cache for all platforms except z/OS®
  - AIX® 6.1 TL6 (AIX 6100-06) and later have an OS API to enable this.
  - AIX 6100-06 is the minimum supported platform for Java 6 R2.6
- Shared class caches for Java 6 R2.6 are unique from previous versions of Java 6
  - Will use more memory when running both on the same system

There are several notable changes for configuring the Java SDK in WebSphere Application Server Version 8.0.

The first is that the default garbage collection policy is now the generational garbage collector. This can affect both server configurations and stand-alone thin clients.

In addition, the stand-alone Java SDKs now default to using a persistent shared class cache, except on the z/OS platform. These shared class caches are unique from other versions of Java 6 and will therefore consume additional memory when running multiple versions simultaneously.

## Java 6 R2.6 SDK configuration changes (2 of 2)

- managesdk command added to configure environments with multiple Java SDKs available
  - Typically used on z/OS and IBM i for 31/32 bit and 64 bit SDK selection
  - Manages SDK used by profiles and default SDKs
  - Replaces enablejvm on IBM i. enablejvm is deprecated.
- Handling Endorsed Standards Override in stand-alone environments
  - `-Djava.endorsed.dirs=WAS_HOME/runtimes/endorsed`

There is a new command line tool available to help manage and configure environments with multiple Java SDKs. Typically, this will be limited to z/OS and IBM i selection of 31 or 32 bit Java SDK versus 64 bit SDK selection. The tool can configure profiles and command line tools to use specific SDKs and can change the defaults for new profile creation. This tool will replace the enablejvm common line tool on IBM i, which is being deprecated.

When running stand-alone SDKs, there are APIs that should be overridden with current versions. Use the `java.endorsed.dirs` system property to reference the appropriate runtime jar file. This will be handled automatically for server configuration.

## Java 6 R2.6 SDK garbage collection policy changes (1 of 3)

- Garbage collection policy subpool is now an alias for garbage collection policy optthruput
- New verbose garbage collection format
  - Handle more concurrent garbage collection policies
  - Infer start / stop from event stream
  - Better format for tools

Some of the garbage collection policies have also changed. The subpool policy is no longer needed and is now an alias for the optthruput policy.

In addition, a new verbose garbage collection log format has been defined to support additional garbage collection policies. The format is less hierarchical and provides better support for analysis by tools. Start and stop of various garbage collection phases may be inferred by examining the event stream.



## Java 6 R2.6 SDK garbage collection policy changes (2 of 3)

- New balanced garbage collection policy for large heaps (> 4 GB, 64 bit only)
  - Use `-Xgcpolicy:balanced -Xcompressedrefs`
  - Supported platforms are Windows® x86, Linux® x86, AIX, Linux POWER®, z/OS, z/Linux
  - Partial garbage collections should be the norm, but global garbage collection phases can be triggered by a system garbage collection and startup

A new garbage collection policy has been added for large heaps, generally of 4 GB or more, but it is only supported on 64 bit systems with compressed references turned on and only on some platforms. The balanced garbage collection policy tries to reduce pause times for garbage collection by performing partial garbage collection phases on subsets of the heap. Partial garbage collection phases should be the norm in the event log, but a global garbage collection phase may be triggered by a system garbage collection or during startup as the heap is growing.



## Java 6 R2.6 SDK garbage collection policy changes (3 of 3)

- New balanced garbage collection policy for large heaps (> 4 GB, 64 bit only)
  - Typically reports less free memory than `gencon` policy; fragmented free memory is not reported
  - May require approximately 7% more heap than other policies to manage space
  - Use global garbage collection frequency and global mark phase frequency versus `gencon` concurrent garbage collection frequency to identify appropriate heap size
  - Processor saturation can affect partial garbage collections

It should be noted that the balanced garbage collection policy will typically report less memory available than other garbage collection policies since fragmented free memory in the heap is not counted. Also, this policy may use about 7% more heap memory due to memory management structures that are kept. When tuning this policy, look at the global garbage collection phase frequency and global mark phase frequency versus the `gencon` concurrent garbage collection phase frequency to help identify appropriate heap sizes. Also, processor saturation can affect partial garbage collection phase completion and cause greater global garbage collection phase frequency.

## Java 6 R2.6 SDK heap dump changes

- Format for heap dump has changed
- Requires updates to analysis tools like IBM Support Assistant

Finally, the heap dump format has been changed to include more information. This will affect tools that are used to analyze heap dumps. In particular, the latest level of IBM Support Assistant should be used to analyze the heap dump format used by the Java 6 R2.6 SDK.

## Summary

- Java SDK continues to improve in performance and scalability
- Memory management is improved to support increasingly large 64-bit heap sizes
- Serviceability of the Java SDK continues to improve

This presentation has covered changes to the Java SDK in the areas of performance, memory management, and serviceability.

## 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 email feedback:
- [mailto:iea@us.ibm.com?subject=Feedback\\_about\\_WASv8JDK\\_Changes.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WASv8JDK_Changes.ppt)
  
- This module is also available in PDF format at: [../WASv8JDK\\_Changes.pdf](#)

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



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, AIX, POWER, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, 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 PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION 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 ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.