



IBM Software Group

# IBM® WebSphere® Application Server for z/OS® V6.0.2

## *Serviceability features*



@business on demand.

© 2005, 2006 IBM Corporation  
Updated November 3, 2006

This presentation will explore the serviceability features of WebSphere Application Server for z/OS, version 6.0.2.

## Agenda

- Class Loader Viewer
- Memory Dump Diagnostic for Java
  - ▶ Examples
- FFDC
- APAR PK05477 rollup
- Miscellaneous Debug Information
  - ▶ Dump Types



After looking at the Class loader viewer, this presentation will examine the memory dump diagnostics for Java, First Failure Data Capture, APAR PK05477 roll-up, and other miscellaneous debugging information.

## Section

# ***Class loader viewer***

The first section covers the class loader viewer.

## Class Loader Viewer

- Class loaders can seem complex and confusing
  - ▶ relationships between class loaders - not obvious
- Class Load Viewer can be used to visualize class loader relationships
  - ▶ Useful for diagnosing problems related to class loaders



Diagnosing problems with class loaders can be complicated and time-consuming. It is often difficult to realize how configuration choices affect the way that classes are loaded, so the wrong class can inadvertently get loaded for an application. What the class loader viewer does is help you better understand class loader relationships by visualizing those relationships based on your configuration settings, thus helping you to diagnose and fix problems more quickly. The class loader viewer was previously available as a download and it had to be installed in order to be utilized. It is now integrated into the administrative console which makes it much handier to use. Keep in mind that it will not fix any problems, but it can help diagnose a Classloader problem. Classloader information can also be exported and sent in to the support team when needed.

IBM Software Group IBM

## Integrate Class Loader Viewer in Administrative Console

- A plug-in for administrative console which will display various class loaders and their hierarchy
- Base Application Server on z/OS shown here

Serviceability features 5

© 2005, 2006 IBM Corporation

As shown here, the Class Loader Viewer is a plug-in for the administrative console. It shows you the class loaders visible to a Web module or enterprise bean in an installed enterprise application. Once you select a particular Web module or enterprise bean, you can see which Classes are loaded by a particular class loader by clicking on the 'Classes' link. You can also look at a Table View, which provides information about class loaders visible to the module in an HTML table format for each class loader. Information found in the Table View includes delegation, Classpath and classes loaded. If you are looking for a particular class, you can search for it. The resulting page will tell you what class loader loaded the class, the source of the class, and the code source. You will note that this slide is showing you the class loader viewer from a Base Application Server install on z/OS. The next slide shows why that is an important distinction.

## Integrate class loader viewer in administrative console

- Network Deployment Configuration on z/OS

Really  
warning  
messages

The first thing you will notice when going into the viewer on a Network Deployment Configuration on z/OS is the 'error' message. This should really be a 'warning'. The class loader viewer is not truly supported in a multi-servant environment. If the code detects that it is an ND environment under z/OS, it assumes that it may be a multi-servant environment and generates this message. It is a 'warning' in the sense that no functions are disabled and if you are working with a single-servant server in an ND environment, this can be ignored. The limitation with the multi-servant environment is that you don't know which servant the information is coming from. Each time you click on one of the options, you may go to a different servant and while one servant may have the class loaded, another may not have it loaded yet so your results will be different. So, be forewarned!

## Section

# ***Memory dump diagnostic for java***

The next serviceability enhancement is a tech preview, the Memory Dump Diagnostic for Java.

## Memory dump diagnostics for java

- Currently there are a number of tools available to analyze heap dumps
  - ▶ HeapAnalyzer, HeapRoots, HeapWizard, LeakBot
  - ▶ Abilities vary, no one complete tool
  - ▶ Not part of WebSphere Application Server
- stand-alone tool for analyzing heap dumps generated by an Application Server
  - ▶ Provided as a technical preview in V6.0.2
  - ▶ Available for download at:  
[http://www.ibm.com/developerworks/websphere/downloads/memory\\_dump.html](http://www.ibm.com/developerworks/websphere/downloads/memory_dump.html)



Currently, there are a number of different tools available to analyze memory leaks, including HeapAnalyzer, HeapRoots, HeapWizard and LeakBot. They all have nice features and the Memory Dump Diagnostics tool for Java attempts to combine them to form a best of breed tool. For instance, LeakBot has a nice GUI interface while HeapAnalyzer allows you to compare 2 heapdumps. The Memory Dump Diagnostics tool has combined these into a single tool, allowing you to compare 2 heapdumps using a nice GUI. The tool is available for download on the developerworks website as shown on the slide.



## Benefits and uses

- System administrators can analyze heap dumps to find applications responsible for memory leaks
  - ▶ Can be traced to specific components
  - ▶ Applicable to production environment
- Application developers can use the tool to better test their code, before deploying to production
  - ▶ Visualize a Java memory dump



One benefit to using the tool is the ability to trace possible memory leaks to a specific component. Once a component is identified, the application developers will most likely need to get involved. In-depth knowledge of the code is often necessary to determine whether what the tool found is expected behavior. It is important to understand that the tool can just point out what **appears** to be strange behavior; for instance, it might pinpoint a large number of objects as a possible memory leak and while it may be a memory leak, a developer might look at the code and determine that the large number is expected.

## Supported platforms

Platform	WebSphere version	JDK Version	Java Memory Dump Format
AIX	6.0 - 6.02	IBM JDK 1.4.2	Portable Heap Dump (PHD)
	5.1	IBM JDK 1.4.1	IBM Text Heap Dump
	5.0 - 5.02	IBM JDK 1.3	IBM Text Heap Dump
	4.0	IBM JDK 1.3	IBM Text Heap Dump
Windows	6.02	IBM JDK 1.4.2	Portable Heap Dump (PHD)
	5.1	IBM JDK 1.4.1	IBM Text Heap Dump
	5.0 - 5.02	IBM JDK 1.3	IBM Text Heap Dump
	4.0	IBM JDK 1.3	IBM Text Heap Dump
Solaris	6.0 - 6.02	SUN JDK 1.4	HPROF
	5.1	SUN JDK 1.4	HPROF
	5.0 - 5.02	SUN JDK 1.3	HPROF
	4.0	SUN JDK 1.3	HPROF
z/OS	6.01 - 6.02	IBM JDK 1.4.2	SVC/PHD
	5.1	IBM JDK 1.4.1	SVC
	5.0-5.02	IBM JDK 1.3	SVC

As you can see on this slide, the Memory Dump Diagnostic Tool for Java is supported on a number of platforms. On z/OS, the format of the dump that can be analyzed includes the new Portable Heap Dump format, or PHD, and the SVC dump that most system programmers are more familiar with.

## Obtaining a memory dump

- Automatic IBM heap dump triggering due to OutOfMemoryError
- Manual triggering of heap dumps
  - ▶ Use verbose gc information to observe Java heap usage
  - ▶ Use wsadmin to trigger two dumps while memory usage is increasing
- Low memory detection in WebSphere Application Server runtime (single servant only)
- Manually transfer large memory dump files out of production environment



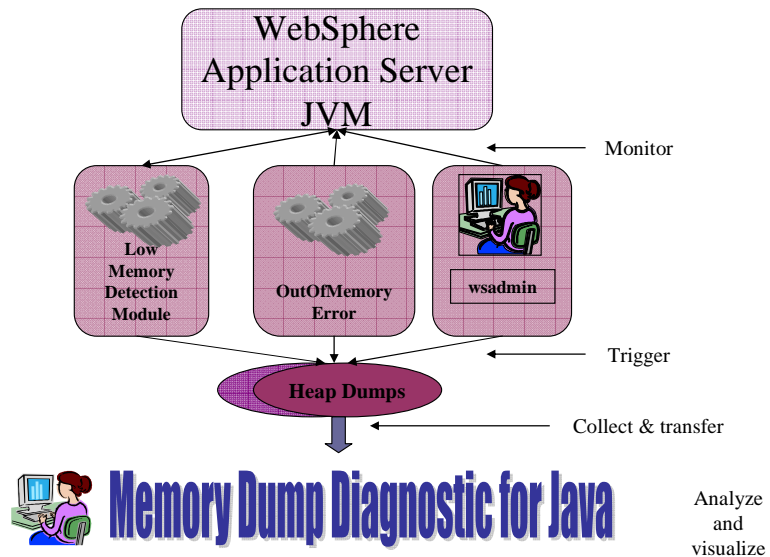
There are a number of ways to generate a Heap Dump. A heap dump will, by default, be generated due to an OutOfMemoryError or you can use wsadmin to trigger one manually. If you suspect you have a memory leak, you can use “verbose gc” to verify your suspicion and then manually take two dumps while the memory is increasing. With two dumps to compare, the tool can better pinpoint what component might not be cleaning up.

Using the Runtime Performance Advisor, you can also enable the memory leak detection advice to help in the early detection of memory problems.

This is a lightweight memory leak detection mechanism that runs within the WebSphere Runtime Performance Advisor framework. It is designed to provide early detection of memory problems in test and production environments but does not provide analysis of the source of the problem. It can be used to provide notification and help generate the information that is required to use analysis tools however.

Once the memory dump is obtained, it needs to be manually transferred out of the Application Server environment to somewhere it can be analyzed using the new Memory Dump Diagnostic tool.

## Tool usage



This graphic shows the stages of memory leak analysis. JVM memory size should be monitored and if necessary you should have some HeapDumps taken by one of the methods shown. Once you have collected any HeapDumps, you can use the Memory Dump Diagnostic tool to analyze the memory leak and determine its cause.

## Types of analysis

- Single memory dump analysis
  - ▶ Detects memory leaking region which has grown far beyond normal operating parameters
  - ▶ Used most often when an OutOfMemory condition is detected
- Comparative memory dump analysis
  - ▶ Compares regions in two memory dumps for growth
  - ▶ Used on slow leaks and when two dumps are available
  - ▶ More accurate but also more time consuming



Two main types of analysis function are provided by this tool: single memory dump analysis and comparative analysis. **Single memory dump** analysis is most commonly used with memory dumps that are triggered automatically by the IBM Developer Kit, Java Technology Edition with OutOfMemoryExceptions. The single memory dump analysis looks to find regions which seem to have grown beyond what the tool would consider 'normal'. **Comparative analysis** is used between two memory dumps taken during the run of a memory leaking application where the total free memory is dropping. For analysis purposes, the memory dump that is triggered early in the run of the leaking application is referred to as the baseline memory dump. The memory dump that is triggered after the leaking application has run for some time to allow for the growth of the leak is referred to as the primary memory dump. In memory leak situations, it is expected that the primary memory dump contains a much larger number of objects occupying a much larger Java heap space than the baseline memory dump.

## Single memory dump analysis

- Calculate aggregated size for all objects through a depth first search from all roots
- Largest drop analysis
  - ▶ Quick analysis based on largest drop in total size between parent and child objects
- Display leak suspect in a tree view of all objects
  - ▶ User can browse and expand each leak suspect and verify the actual cause of the leak
  - ▶ Leak suspect may not be the actual leak cause
- List all objects and types for further information
  - ▶ Sort on criteria such as total reach size, size, name, address
  - ▶ Search for specific objects in the heap by name or address



Single Memory Dump analysis first calculates the aggregated size for all objects. It then seeks to find the largest drop in the Total reach size from a container object (for example, a HashMap) to its largest Total reach sized child. This type of analysis provides a relatively fast analysis result for leak suspects and displays them in a tree view of all objects. You can then browse and expand each suspect and determine if it might be the cause of the leak.

Note that leak suspects are not guaranteed to be the actual root cause. Memory leaks and caches are sometimes indistinguishable - the only difference being in size and application logic. Therefore it is important to browse the heap.

## Comparative memory dump analysis

- More complicated form of analysis involving a number of steps
  - ▶ Conversion to common internal format
  - ▶ Depth first search
  - ▶ Dominator analysis
  - ▶ Heap dump differencing
  - ▶ Region inference
  - ▶ Region matching
  - ▶ Region comparison and report results



The Comparative Memory Dump Analysis is a more complicated form of analysis involving a number of steps. After conversion to a common internal format, a "depth first" search is done on the object reference graph and all the objects are numbered and the references are sorted and stored in intermediate files for faster and more scalable post processing. This step is done for each of the two dumps.

In the dominator analysis, *dominating* relationships and *unique ownership* for each object is computed. This step again is run twice, once for each snapshot.

In the Heap Dump differencing step, the tool determines classes whose objects have increased in numbers from one heap to the next, objects that are present only in the latest snapshot and not in the first snapshot and objects that do not change in numbers between the two snapshots. The output of this step is used in the next steps to identify the leaking region. This step is run once for comparing the two snapshots.

In region inference, the latest snapshot is analyzed to identify the regions corresponding to the classes that have grown. For each region, the leaking unit and the owner chain are identified. The data types or class IDs of the leaking unit and the owner chain are used as a key pattern for the region in subsequent matching operations. This step is run once with the latest snapshots.

In region matching, all the regions that are identified in the latest snapshot are matched against each of the snapshots to identify the region in that snapshot. The nesting relationships between regions that are owned by other regions are computed. This step is part of the comparative analysis and is run twice, once for each snapshot.

Finally, all of the matched regions for each snapshot are compared using the criteria,  $\langle \text{size growth} \rangle * \langle \text{final size} \rangle$  and the leak analysis results are presented. All the nested regions within an owning region are shown with an indentation. Thresholds are computed at 25%, 5%, 0.5% and 0.05% levels with respect to the largest region in the heap.

## 'How-to' whitepaper

- MemDumpDiagFAQS.html
  - ▶ Found with the download  
[http://www.ibm.com/developerworks/websphere/downloads/memory\\_dump.html](http://www.ibm.com/developerworks/websphere/downloads/memory_dump.html)
- Look here for screen captures and information on using the tool to analyze your dumps.



For more information on using the Memory Dump Diagnostic tool, there is a white paper available from the link shown here. This is a good place to see what the tool looks like as it is being used, and for information on how to use it effectively.



## Tips for memory leak analysis

- Monitor verbose gc output to identify the period, duration and rate of memory leak
- Obtain multiple memory dumps throughout the execution of the leaking application to ensure that they encompass the leak duration
- Object caches may appear as leak suspects, examine region details and number of objects to distinguish
- Avoid taking memory dumps immediately after starting
  - ▶ WebSphere caches will be filling during this time
- The larger the leak size between memory dumps, the easier it will be to diagnose



Finally, some tips for memory leak analysis. Monitor “verbose gc” output to help identify how big of a problem the leak is. Is it leaking memory at a fast rate over a few hours or does it appear to be leaking over a period of days or even weeks? Looking at the “verbose gc” information will help you obtain memory dumps that should encompass the leak duration so that the comparative leak analysis will help.

Keep in mind that the tool is showing you leak suspects. Someone familiar with the application will most likely need to look at the leak suspects to rule them out. Object caches often appear as leak suspects but can usually be ruled out once discovered. To avoid having too many caches appear as leak suspects, you should avoid taking memory dumps immediately after starting the server. Lastly, the larger the difference in the two memory dumps, the easier it will be to diagnose.

## Section

# ***FFDC***



The next serviceability item is First-failure data capture enablement on z/OS.

## FFDC

- FFDC now enabled by default on z/OS
  - ▶ Collects and analyzes information on errors and events occurring in the runtime
  - ▶ In addition to the "normal" logging
  - ▶ Data is uniquely identified for the servant region that produced the exception



FFDC, or First Failure Data Capture, has been available and enabled on distributed WebSphere since V5.0. It was never enabled on z/OS because it was not set up to be multi-servant capable. This shortcoming has been addressed in V6.0.2 and the data is now uniquely identified for each servant region.

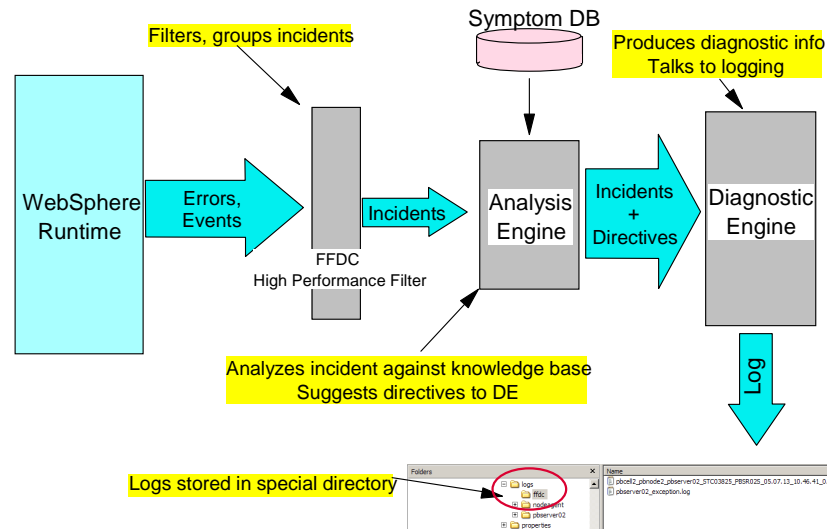
## What is First Failure Data Capture (FFDC)?

- A mechanism to collect, analyze, and diagnose events and errors occurring in the WebSphere runtime
  - ▶ Captures information right when the incident occurs
    - Reduces need for reproducing the problem
  - ▶ In version 5, try-catch blocks were instrumented for FFDC
- Always active, users are not required or supposed to configure FFDC
  - ▶ Totally transparent
  - ▶ Minimal, negligible performance impact
- FFDC produces special logs
  - ▶ Include analysis and diagnostic information
  - ▶ Incidents are logically grouped together
    - Multiple occurrences of the same incident
  - ▶ Data collected greatly varies depending on the incident



FFDC is a mechanism that collects and diagnoses exceptions, error conditions, failure, and potential problem situations at the time they actually occur. Much of the WebSphere runtime code was instrumented to interact with FFDC in V5 - in particular most of the try/catch blocks include FFDC-related code. From your standpoint, there is nothing to configure, turn on, or modify to be covered by FFDC support. It is 'on' by default and it will produce special log files that are meant to be looked at by support personnel when a problem occurs. It has minimal, negligible performance impact and will collect different data depending on the incident being reported.

## How does FFDC work?



This slide shows the architecture of the FFDC engine. The instrumented runtime produces errors and events - which are then filtered by the "high performance filter" (HPF). The HPF creates incidents based on the event data that is produced by the runtime. The HPF is smart enough to recognize whether a certain event is a duplicate and increment a counter of occurrences in that case, rather than creating a new incident.

The Analysis Engine has the primary function of directing the activities of the diagnostic engine. The Analysis Engine uses the Log Analyzer symptom database to influence the diagnostic engine, by suggesting which information needs to be collected and logged and from which components, based on the type of incident. For example, a database related incident may require that the Diagnostic Engine collects and logs information about the SQLCODE from the JDBC layer - a security incident may require a completely different set of data and may involve different components.

The logs produced by the Diagnostic Engine are stored in a special FFDC directory.

## FFDC on z/OS -- enabling

- Three property files that control FFDC (found in each node in the home directory on z/OS):
  - ▶ `ffdcStart.properties` - used while the server is starting
  - ▶ `ffdcRun.properties` - used after the server is ready
  - ▶ `ffdcStop.properties` - used while the server is in the process of stopping.
- Default values in property files:
  - ▶ `FFDC=true`
  - ▶ `Level=4`
  - ▶ `ExceptionFileMaximumAge=7`
- To actually run with FFDC, need to set the Level > 0
- FFDC files will be found in `<WAS_HOME>/profiles/default/logs/ffdc`



There are three properties files that control FFDC. They are found with the product binaries. You should not change these files unless instructed by IBM Service to do so OR if you need to change the number of days between first failure data capture log file purges.

There are a couple of properties found in the properties files that will affect FFDC. The interesting defaults are shown on this slide. For example, `ExceptionFileMaximumAge` is used to tell WebSphere how often it should purge the log files. The level determines what type of processing you expect to perform. Level '0' would basically turn FFDC processing OFF.

## Section

# ***APAR PK07234***



APAR PK07234 is the 6.0 equivalent of an APAR that was created originally for V5.0. It is available in 6.0.2.1.

## APAR PK07234

- Problem 1 (the “bouncing servant”)
  1. Application running with 1 Servant on a 2 system sysplex
  2. Servant fails due to timeout
  3. Work queues up on the workload management queue
  4. When the servant comes up it gets the oldest piece of work and GOTO bullet 2
- Problem 2
  1. Application in production and running on a 2 system sysplex using DB2 locally
  2. WebSphere receives errors from DB2
  3. Workload routing causes most users to now fail



This APAR addresses a couple of workload balancing problems. The first one has to do with a CR that has only one servant defined to it. When that servant fails due to a timeout, a workload balance problem may be introduced while a replacement servant is being started. While the servant is being started, work continues to pile up in the workload management queue as well as ‘age’ in that queue. When the servant is finally ready to accept new work, it will get the oldest piece of work available which may be close to timing out. The new servant fails quickly when this piece of work times out and around and around it goes!

The other workload balancing problem it addresses has to do with a problem in a backend resource such as DB2. The work that was queued to the servant finishes (maybe in record time!) so workload management continues to send work to the servant but what the workload management process does not know is that every piece of work it is sending to that servant is failing with an error from the backend.



## APAR PK07234...

- Solution...
  - ▶ If WebSphere detects no servants are available, will now optionally pause listeners
    - **control\_region\_dreg\_on\_no\_srs** environment variable controls whether CR will reject requests for dispatch within a servant when it detects that there are no servants available to process requests.
      - When a minimum number of servants becomes available, the CR will again begin accepting dispatch requests.
    - **control\_region\_confirm\_recovery\_on\_no\_srs** environment variable determines whether the WebSphere Application Server administrator must respond to WTOR before the CR will begin accepting dispatch requests again
  - ▶ Server stays up!

In order to allow work to be balanced to servers that are functional, WebSphere will de-register the affected application server with workload management and stop the HTTP and MDB listeners when the **control\_region\_dreg\_on\_no\_srs** is set to 1. If this property is set to 0 (zero), the function is disabled. At the point where the CR detects a minimum number of servants are available, it will re-register the application server with workload management, start the HTTP and MDB listeners, and allows requests to be dispatched to the servants.

Another environment variable, **control\_region\_confirm\_recovery\_on\_no\_srs**, indicates whether the CR should immediately allow requests to be dispatched to the servants after a no-servant condition is relieved. If this property is set to 1, requests are not dispatched to the servants until after the WebSphere Application Server administrator responds to WTOR message BBOO0297A. This message is issued following a no-servant situation when the server detects that the required minimal number of servants are available to process requests. If this is set to 0 (zero), the CR determines when to allow requests to be dispatched to the servants after a no-servant condition is detected.

## APAR PK07234....

- Manual alternative...
  - ▶ New option on the modify command:
    - **f <server>,pauselisteners**
      - Prevents work from being accepted into the server.
    - **f <server>,resumelisteners**
      - Allows work to be accepted into the server again after PAUSELISTENERS command was issued.

```
f bbos001,pauselisteners
BBOO0211I MODIFY COMMAND PAUSELISTENERS COMPLETED SUCCESSFULLY

f bbos001,resumelisteners
BBOO0211I MODIFY COMMAND RESUMELISTENERS COMPLETED SUCCESSFULLY
```

A manual alternative has also been provided. This can be combined with an automation to handle the problem if you are able to detect that you are having a system problem before servants are terminated because of timeouts. Commands to manually pause the listeners are shown on the slide.

## Section

# *Summary*

And, in summary...

## Summary

- Explained the new availability of the Class Loader Viewer
- Showed the capabilities and benefits of new memory leak detection tools
- Explained how FFDC fits into z/OS now
- Looked at some support that will help with workload balancing problems



... this presentation has looked at new serviceability features that are available in WebSphere Application Server for z/OS, version 6.0.2. You have seen the newly integrated class loader viewer and the new memory leak detection tool. FFDC is now available on the z/OS platform, and an APAR that addresses some workload balancing problems.

## 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	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIK	DB2 Universal Database	Lotus	pSeries	zSeries

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

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

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 and/or changes in the product(s) and/or program(s) 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 NONINFRINGEMENT. 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 (e.g., 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 2005, 2006. 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.