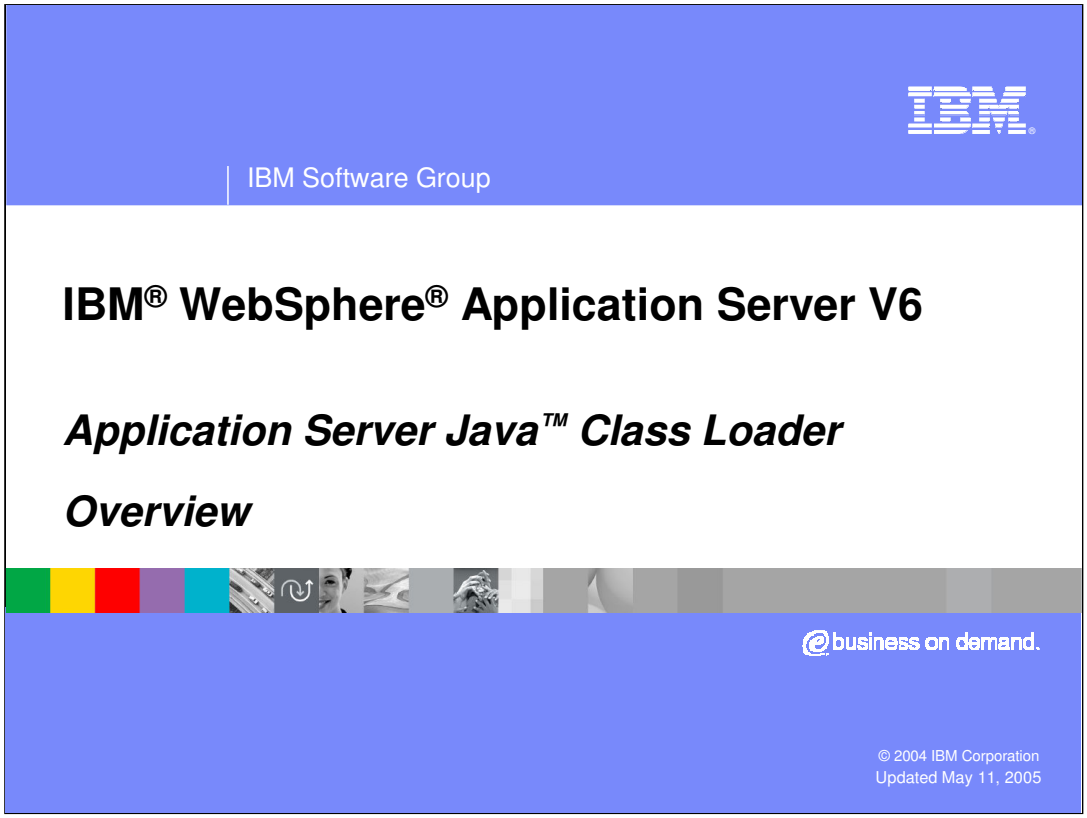
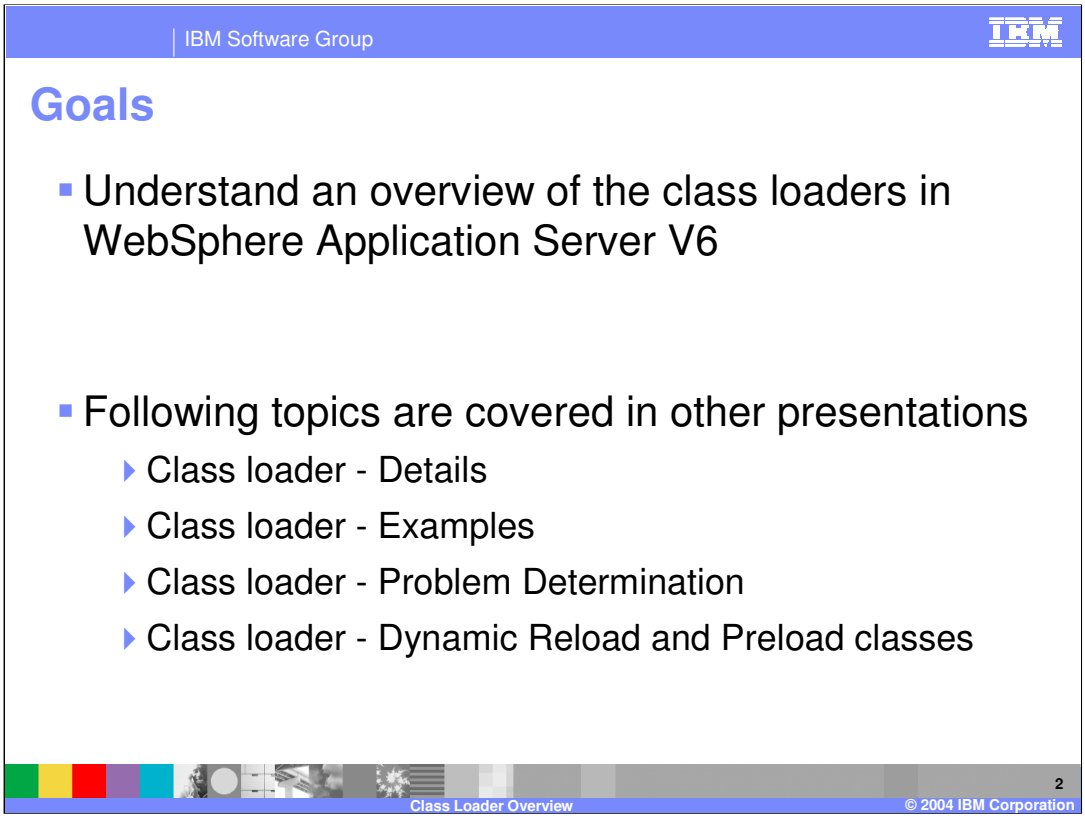


This presentation will provide an overview of the Java class loader in WebSphere Application Server V6.

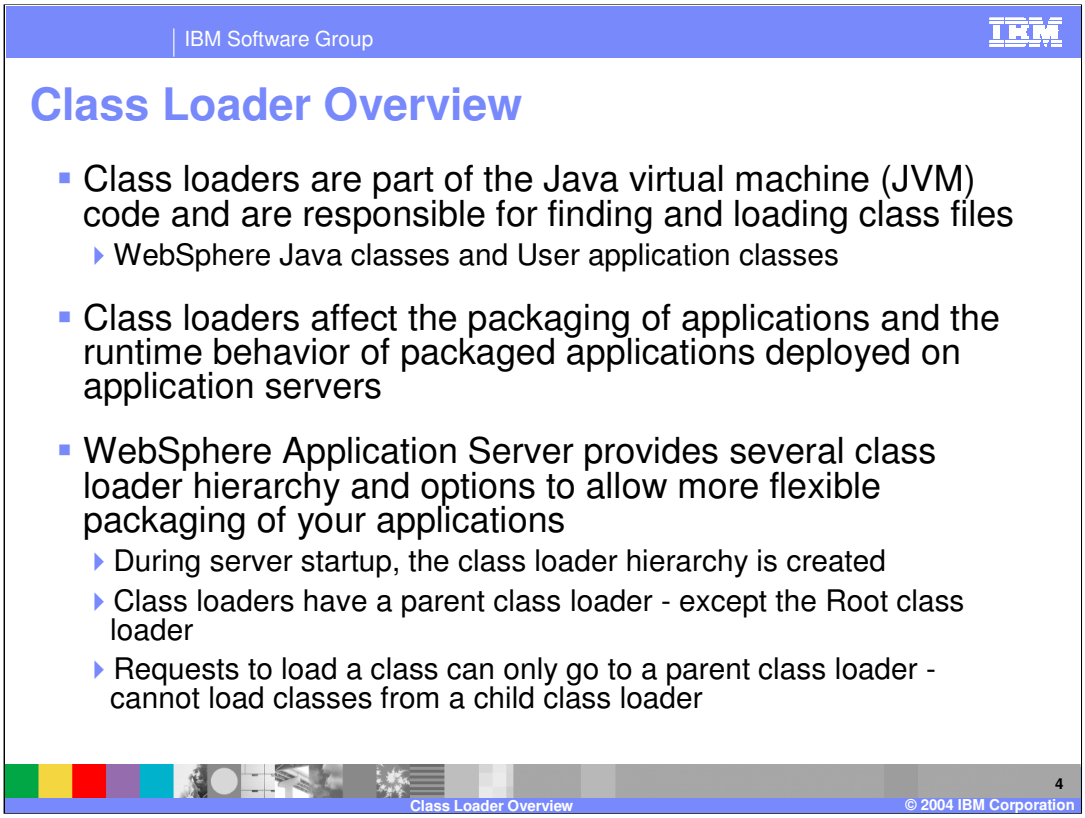


The goal of this presentation is to provide an overview of the class loaders in WebSphere Application Server V6. Other presentations cover details and examples on the class loaders, as well as similarities of the WebSphere Application Server V5 and V6 class loaders.

Agenda

- Class loader
  - Overview and hierarchy
  - Shared library
  - Loading native library
- Summary and References

The agenda for this presentation is to discuss the Application Server class loader, including hierarchy, shared library, and native library.

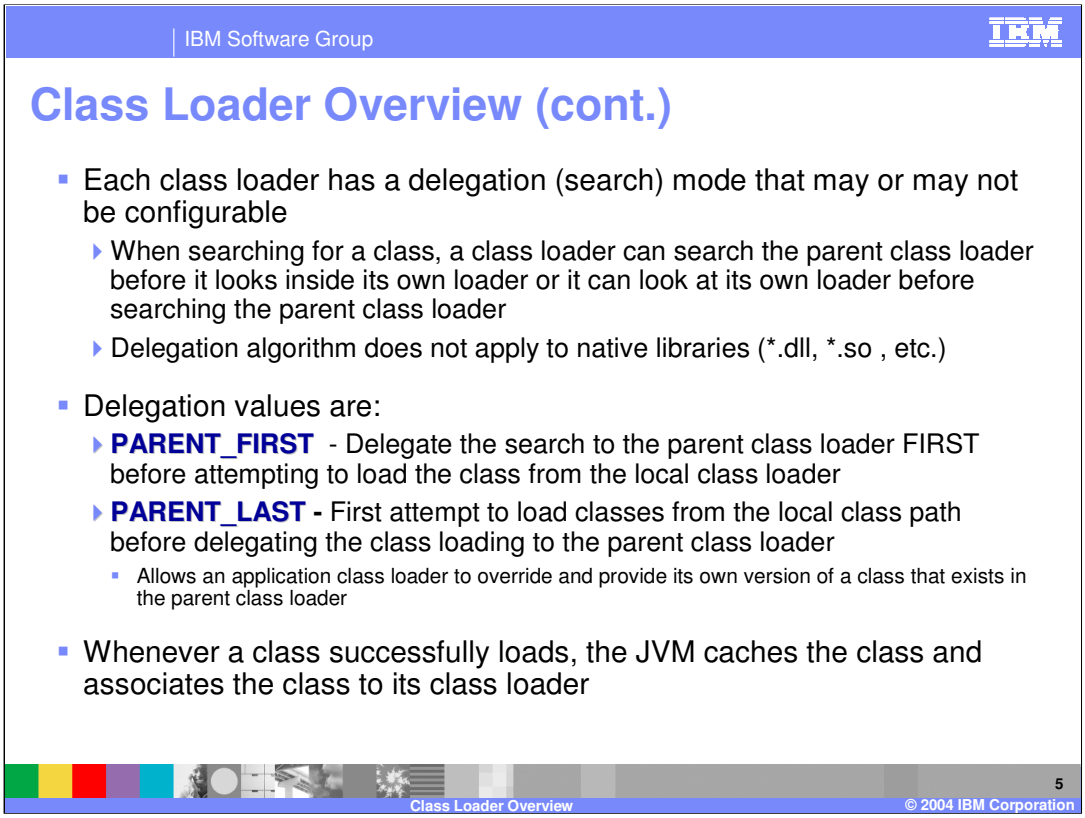


A class loader is an inherent part of the JVM and is used to find and load all Java classes and native libraries. The class loader in V6 is unchanged from V5.

There are several different class loaders provided by the JVM and WebSphere Application Server, forming a class loader hierarchy with class loaders having a parent class loader. The root class loader has no parent class loader.

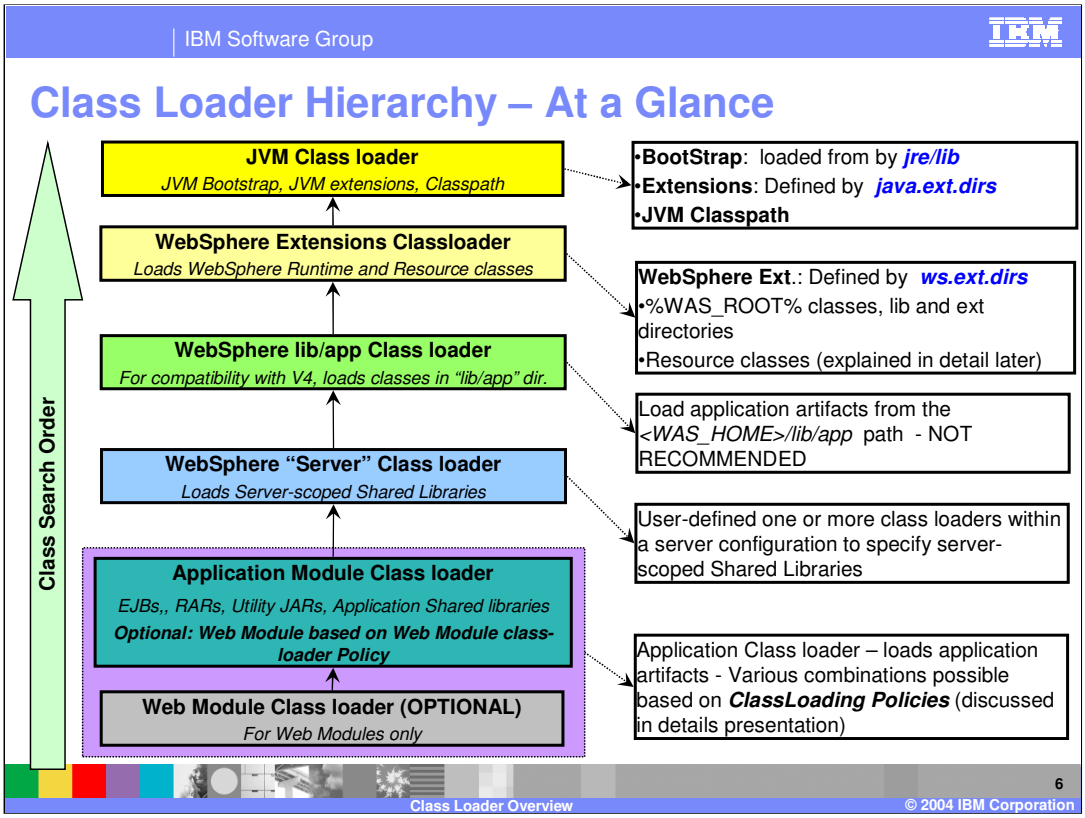
In the class loader hierarchy, a request to load a class can go from a child class loader to a parent class loader but never from a parent class loader to a child class loader.

If a class is not found by a specific class loader or any of its parent class loaders, then a class not found exception will result.



Java class loading allows for a search mode that lets a class loader search its own class path before requesting a parent class loader or search via the parent class loader before its local class path. These searches, or delegation modes, are PARENT_LAST and PARENT_FIRST respectively.

The JVM will cache the class on a successful load and will associate the class with its specific class loader.



Class loaders are organized in a hierarchy. This means that a child class loader can delegate class finding and loading to its parent, should it fail to load a class.

The root of the hierarchy is occupied by the JVM class loader and its Bootstrap class loader that loads the JVM classes. The JVM class loader loads the JVM classes, the JVM extension classes, and the classes defined in the classpath environment variable.

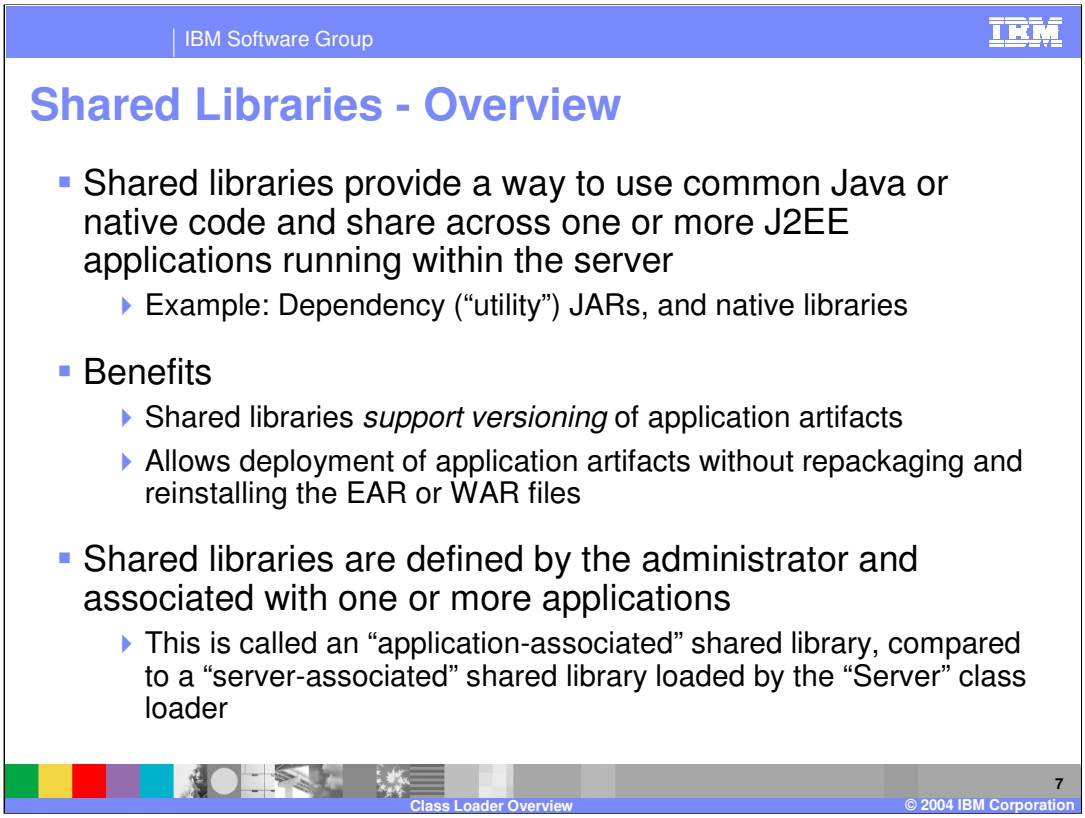
Next in the hierarchy is the WebSphere Extension class loader. This loads all WebSphere Application Server classes, resource adapters, and other classes.

Next is the WebSphere Application Server application class loader. This loads classes from the WebSphere Application Server library application directory. In V4, this was used to specify classes shared by all applications. Beginning with V5, the shared library function provides a better option to share classes across one or more applications. Therefore, this class loader is provided mainly for backward compatibility.

Next is the WebSphere Application Server **Server** class loader. This loads shared libraries that are defined at the server level and can be accessed by all applications.

The last class loader is the Application class loader that loads the J2EE applications. It has several options and class loading policies.

Providing this hierarchy allows for the flexibility that may be required by a set of applications. In most cases, use of the default class loader and options are sufficient.



In WebSphere Application Server V4 if you required JARs to be shared by more than one application, you would put them in the *Install_Root*/lib/app directory. The drawback to this is that the JARs in this directory are exposed to all the applications. In WebSphere Application Server V5 and V6, shared libraries provide a better mechanism where only applications that need the JARs are exposed to them. Other applications are not affected by the shared libraries. The administrator defines the shared libraries by assigning a name and specifying the file or directories that contain the code to be shared. These defined shared libraries can then be associated with one or more applications running within the Application Server.

The advantage of using an application-scoped shared library, is the capability of using different versions of common application artifacts by different applications. A unique shared library can be defined for each version of a particular artifact, for example, a utility JAR.

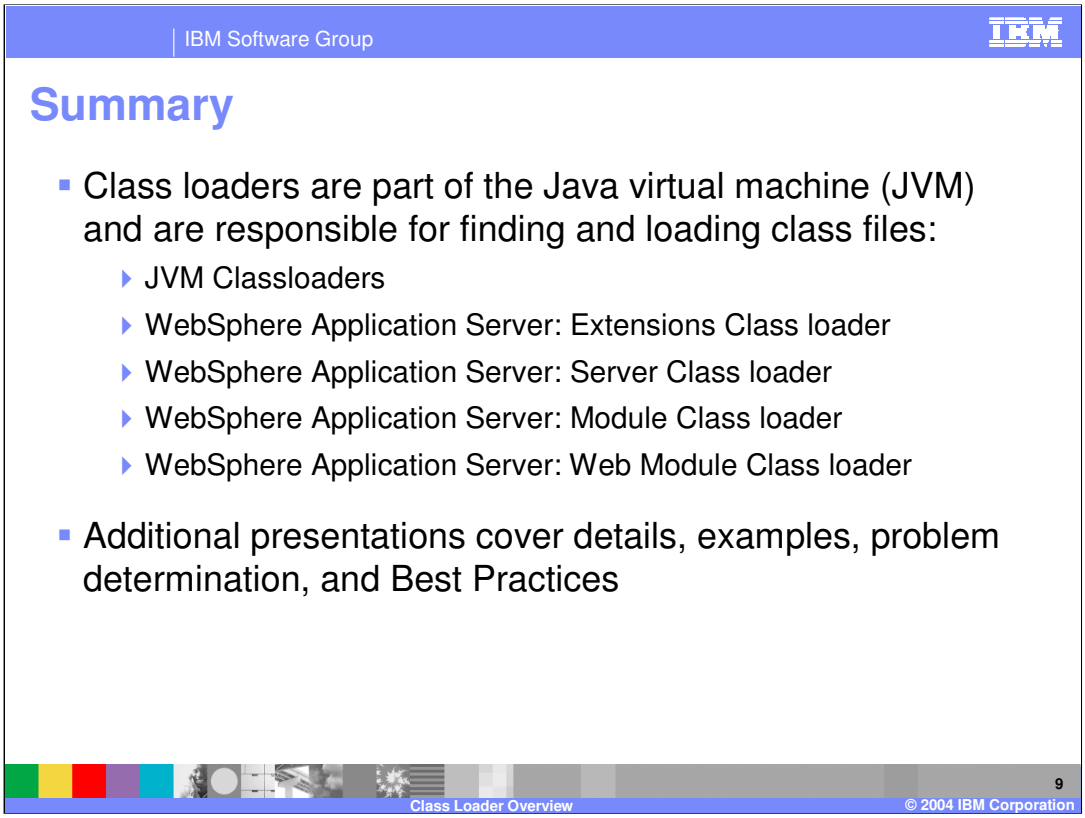
If a native library loaded by shared library requires a second native library, then it must not be specified in the shared library path. Rather, it must be specified in the JVM native library path and will be loaded by the JVM class loader.

Native Libraries - Overview

- Native libraries are non-Java code used by Java via the Java Native Interface (JNI). They are platform specific files, for example: '.dll' in Windows™, '.so' and '.a' in Unix
- Java applications use the System.loadLibrary(*libName*) method to load the native library. It is loaded at the time of the System.loadLibrary(…) call.
- The JVM uses the caller's class loader to load the native library. If that fails, it then uses the JVM System class loader
  - If both fail to load, an UnsatisfiedLinkError will result
- Native libraries are located on the native library paths of the JVM class loader and the WebSphere Application Server (Extensions, Server, and Application module) class loaders

Native libraries are loaded by Java using the System load library method. They are loaded on demand, when needed. Native libraries could be located in the JVM class loader or one of the WebSphere Application Server class loaders; namely, the Extension, Server, or the Application module class loader.

WebSphere Application Server Extensions, Server, and Application module class loaders define a local native library path, similar to the java.library.path supported by the JVM class loader.



In summary, this presentation has provided an overview of class loaders. Class loaders are a part of the java virtual machine and are responsible for finding and loading class files. Other presentations provide a more detailed discussion, providing examples, problem determination, and best practices information.

Template Revision: 11/02/2004 5:50 PM

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
AIX	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.

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.