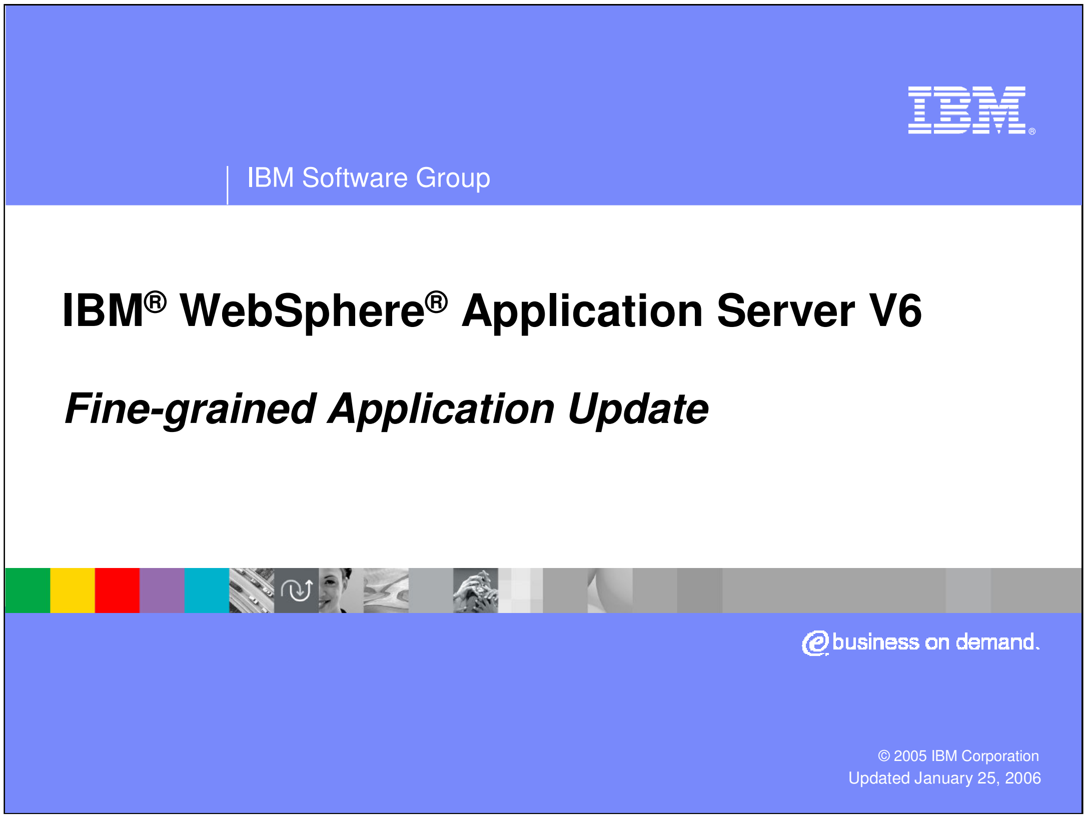


IBM Software Group

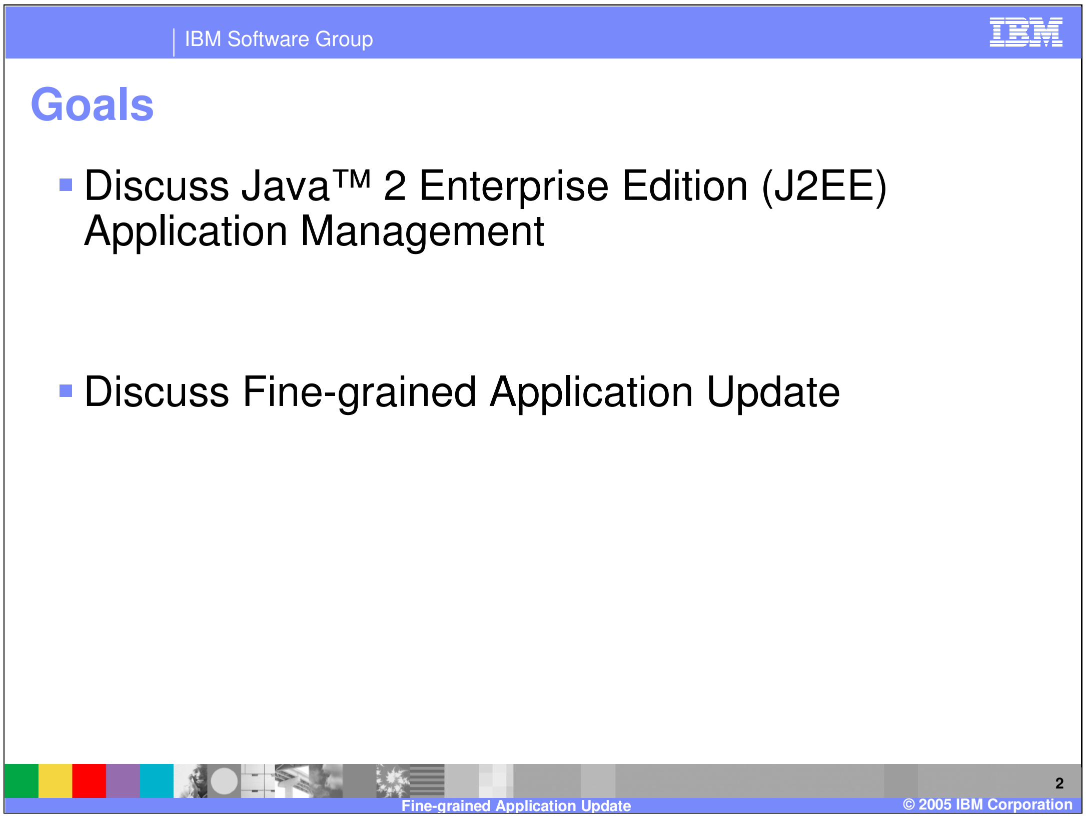
IBM® WebSphere® Application Server V6

Fine-grained Application Update

business on demand.

Updated January 25, 2006

This presentation will focus on the basics of Fine-grained Application Update in WebSphere Application Server, Version 6.

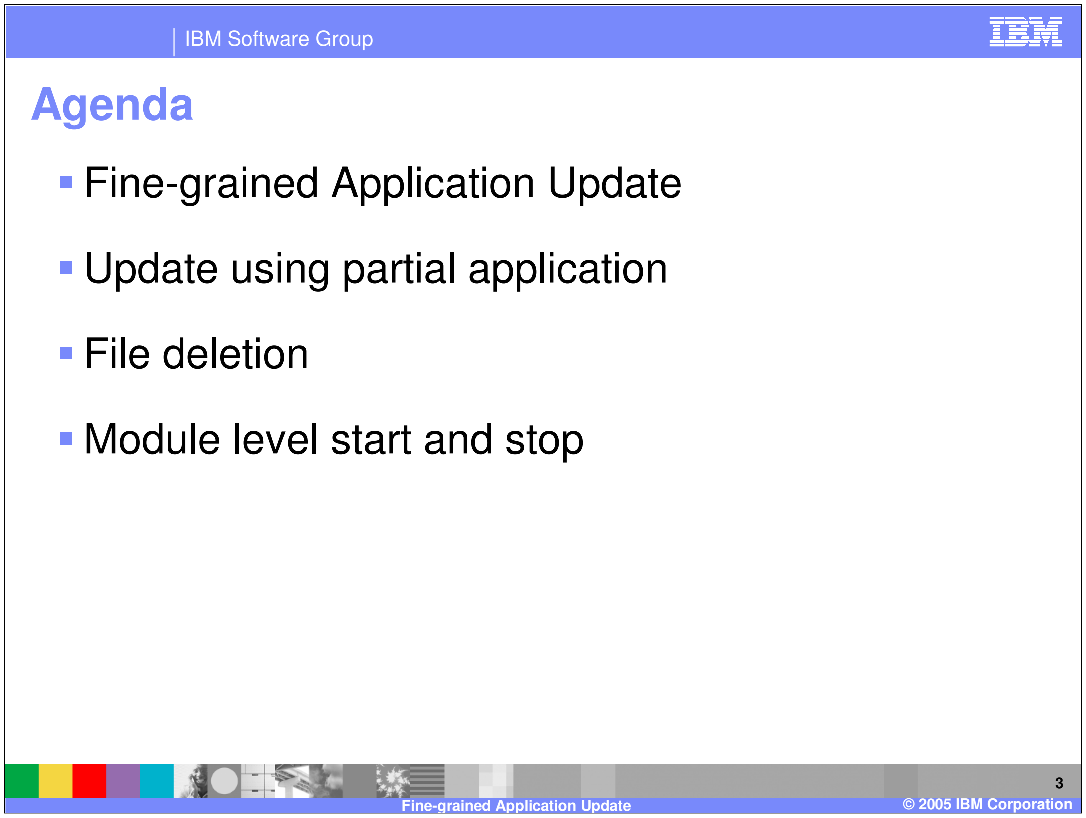


Goals

- Discuss Java™ 2 Enterprise Edition (J2EE) Application Management
- Discuss Fine-grained Application Update

The goal of this presentation is to discuss application management.

We will also talk about Fine-grained Application Update, which is one of the new features in WebSphere Application Server version 6.



Agenda

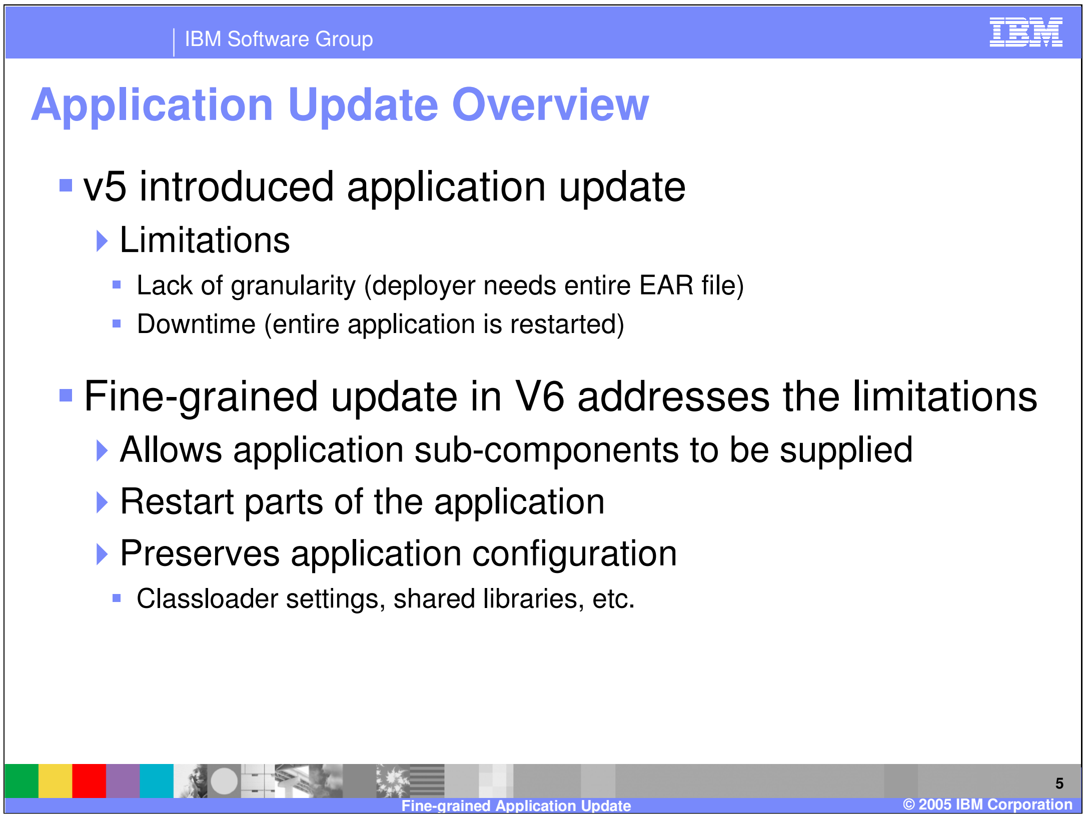
- Fine-grained Application Update
- Update using partial application
- File deletion
- Module level start and stop

The agenda for this presentation applies to both the Network Deployment packaging and the Express packaging of WebSphere Application Server. Fine-grained Application Update, file deletion, and how module level start and stop support effects the behavior of applications during the update process, will all be discussed within this presentation.

Section

Fine-grained Application Update

This section will discuss Fine-grained Application Update.



IBM Software Group

Application Update Overview

- v5 introduced application update
  - Limitations
    - Lack of granularity (deployer needs entire EAR file)
    - Downtime (entire application is restarted)
- Fine-grained update in V6 addresses the limitations
  - Allows application sub-components to be supplied
  - Restart parts of the application
  - Preserves application configuration
    - Classloader settings, shared libraries, etc.

5

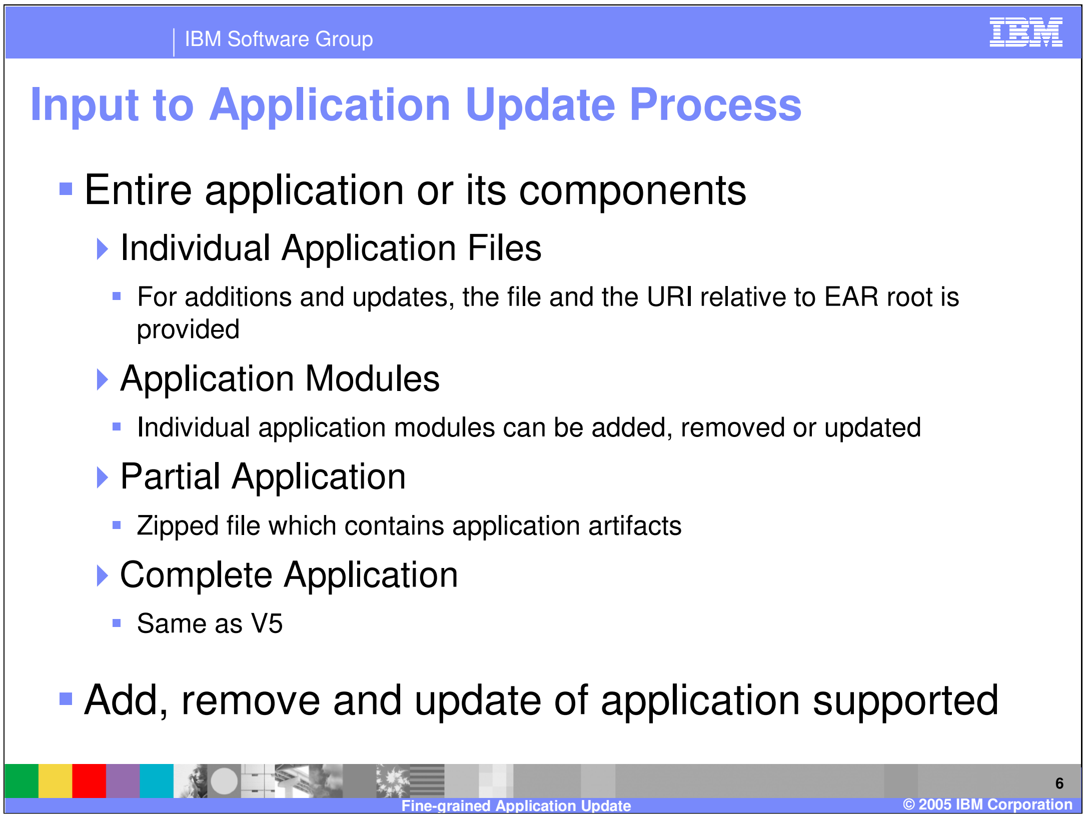
Fine-grained Application Update

The application management framework in WebSphere Application Server 5.0 introduced an *application update* function which was exposed via various administrative programs such as Web-based Administrative Console, wsadmin scripting tool as well as the programmatic MBean interface. This update function took a new application EAR file for an application deployed on the WebSphere Application Server platform and performed the necessary deployment steps to replace the existing application configuration with the new one. To make the new application configuration take effect the application management logic stopped the running application, replaced the application files and restarted the application. There were two main problems with the update function –

Granularity of update input (or the lack thereof) - The update could only be initiated if the deployer had the entire EAR file that was to be used to update the deployed application. It was not possible to *patch* a deployed application using partial contents; for example, a single file or a module.

Downtime during update - The entire application had to be recycled (stopped and restarted) in order to apply the updates. Extra steps needed to be taken to ensure continuous availability of application artifacts.

The fine-grained update feature in WebSphere Application Server 6.0 is intended to address the above problems. You can now apply a patch – for example, supply a single JSP and have WebSphere Application Server replace the same named JSP in the running application. It is now possible to restart parts of the application, and the application update process preserves configuration settings such as bindings, Classloader settings, shared libraries, and so on.



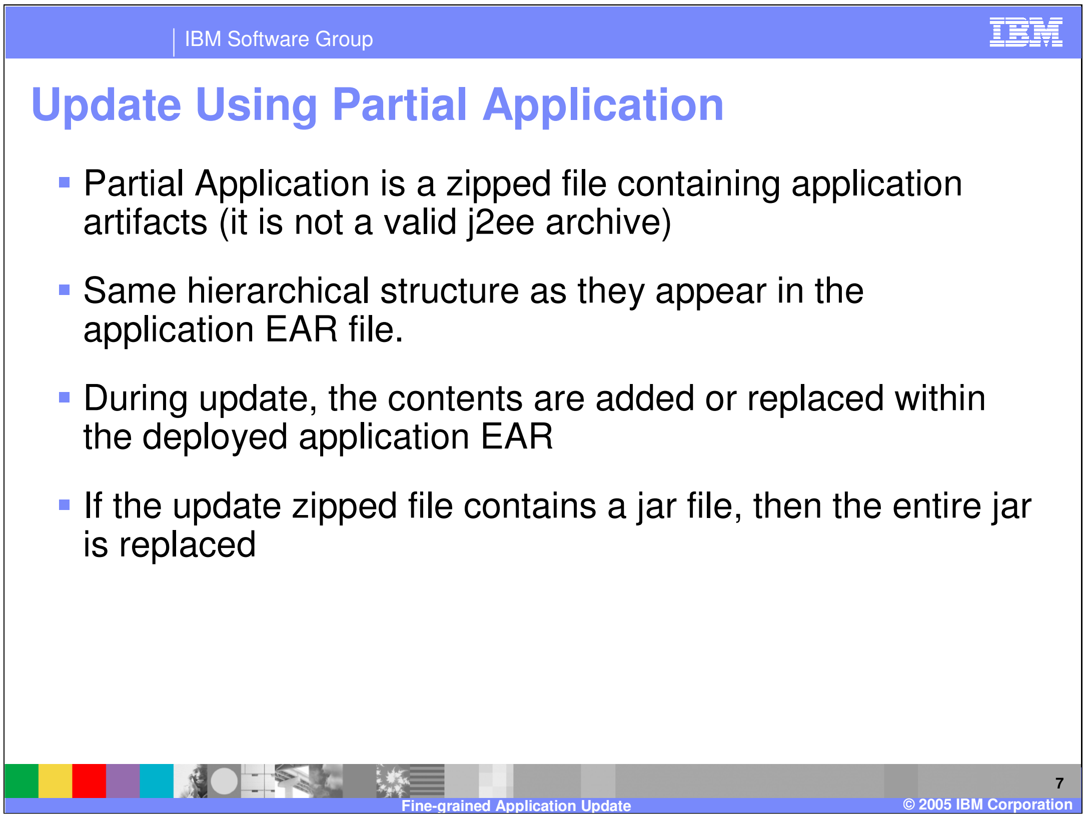
Input to Application Update Process

- Entire application or its components
  - Individual Application Files
    - For additions and updates, the file and the URI relative to EAR root is provided
  - Application Modules
    - Individual application modules can be added, removed or updated
  - Partial Application
    - Zipped file which contains application artifacts
  - Complete Application
    - Same as V5
- Add, remove and update of application supported

The input to the Application update process can either be an entire EAR file, or individual components. Those components may be individual files, such as a JSP or a servlet. The administrator also needs the URI relative to the EAR root. You can also update complete Web modules, or EJB Jar files in an application.

A third option is to provide a partial application, which would be a zipped file, which contains application artifacts. This provides a simple mechanism for updating multiple files in the application.

Finally, the three options for changing an application are Add, Remove, and Update.



IBM Software Group

Update Using Partial Application

- Partial Application is a zipped file containing application artifacts (it is not a valid j2ee archive)
- Same hierarchical structure as they appear in the application EAR file.
- During update, the contents are added or replaced within the deployed application EAR
- If the update zipped file contains a jar file, then the entire jar is replaced

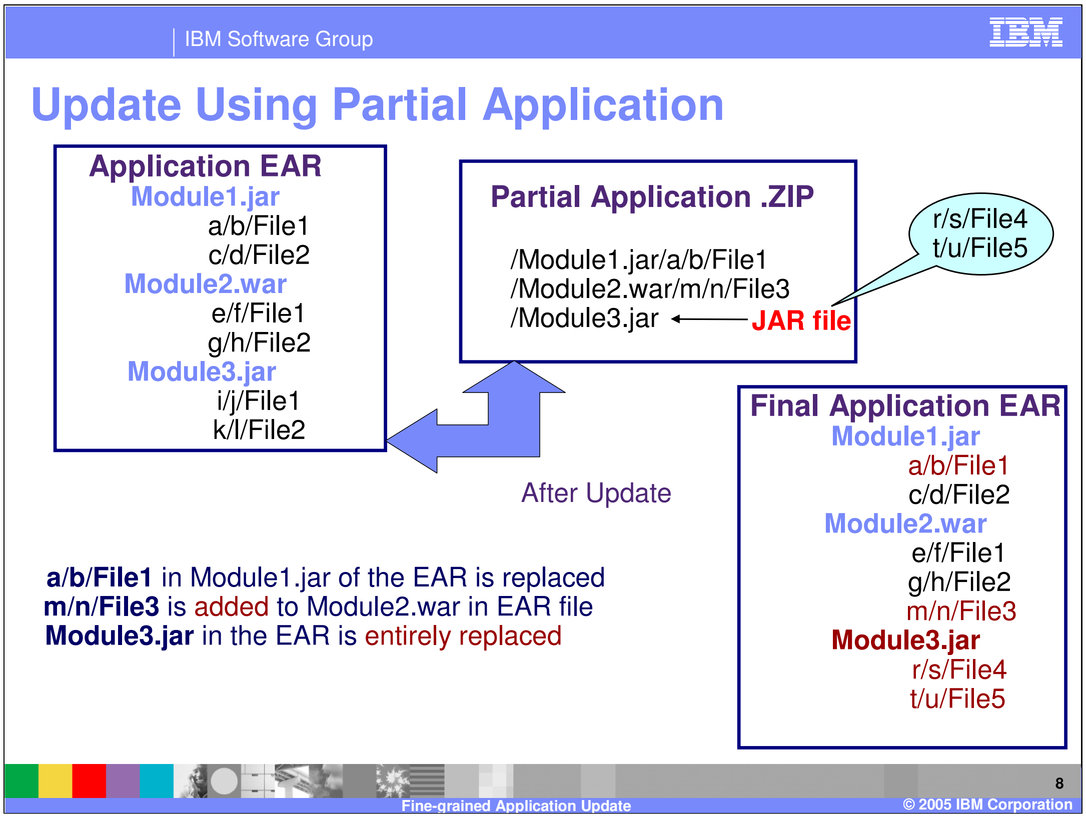
7

Fine-grained Application Update

© 2005 IBM Corporation

The partial application packaging concept is introduced specifically to handle fine-grained application updates. A partial application contains a subset of application components and is not a valid J2EE archive. It is a zipped file which contains application artifacts in the same hierarchical structure as they appear in an application EAR. A simple zipped file format has been chosen to represent a partial application so that no special tools are required to create a partial application.

During application update, the contents of the partial application are merged with the deployed application EAR, using simple file replacement logic based on the file location in the EAR structure. The contents of the files in the partial application are **not** merged with the corresponding files in the application EAR. Therefore a partial application should not contain another archive (e.g. JAR, WAR etc.) unless the corresponding archive in the application EAR it is to be entirely replaced. Individual files in the application modules that are to be updated or added, can be packaged directly in the partial application by adding module URI and file URI within the module in the file path. A file entry in the partial application is added or updated in the application EAR based on its presence in the application EAR file.



Update Using Partial Application

Application EAR

Module1.jar

a/b/File1

c/d/File2

Module2.war

e/f/File1

g/h/File2

Module3.jar

i/j/File1

k/l/File2

Partial Application .ZIP

/Module1.jar/a/b/File1

/Module2.war/m/n/File3

/Module3.jar ← **JAR file**

r/s/File4

t/u/File5

After Update

Final Application EAR

Module1.jar

a/b/File1

c/d/File2

Module2.war

e/f/File1

g/h/File2

m/n/File3

Module3.jar

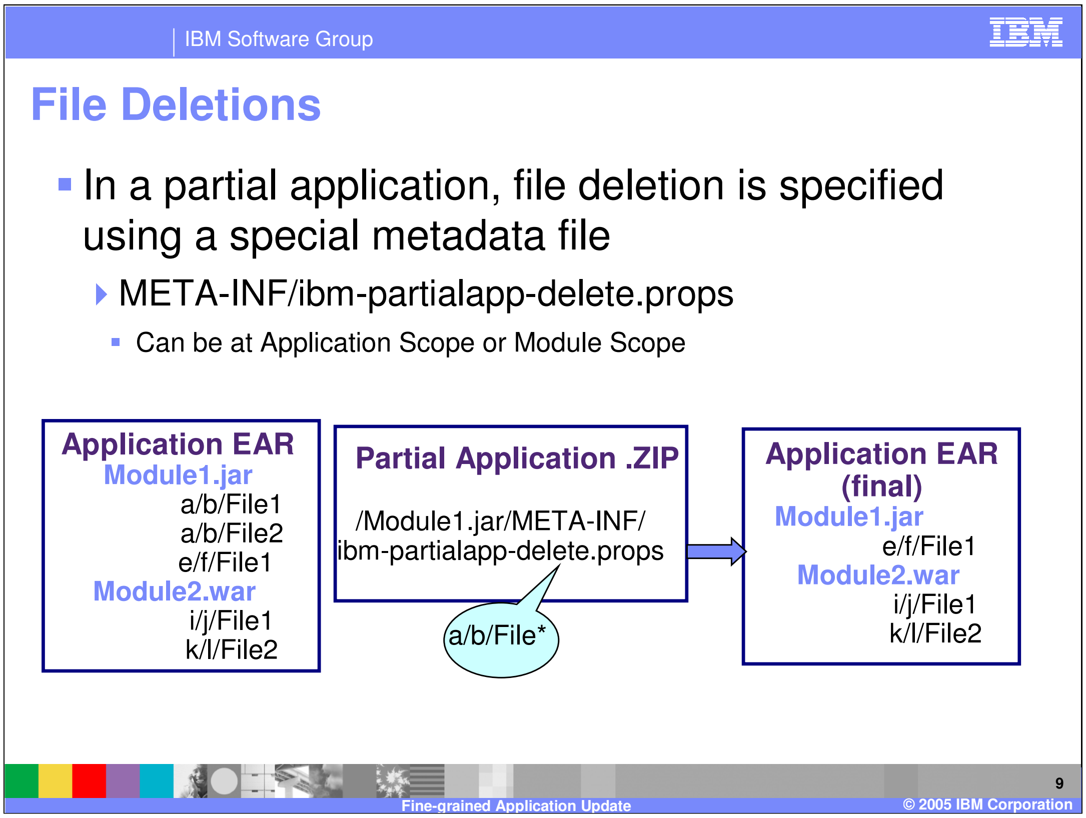
r/s/File4

t/u/File5

a/b/File1 in Module1.jar of the EAR is replaced
m/n/File3 is added to Module2.war in EAR file
Module3.jar in the EAR is entirely replaced

Here there is an application EAR file which is currently running in WebSphere Application Server. The application EAR file has three modules. A Module1.jar, which contains File 1 and File 2 in different directories, a Module2.war, which contains a File 1 and File 2 in different directories, and Module 3, which contains File 1 and File 2 in still other directories. In this example, you need to update the application with a new version of the code, so a partial application is provided. The partial application zipped file contains the following entries: /Module1.jar /a/b/file1, which will replace the a/b/file1 in the application, and Module2.war/m/n/File3, which will be added to the application since it does not already exist, and Module3.jar, which will replace the entire contents of the existing Module3.jar in the enterprise application.

So the final application EAR file will look like the one that you see at the lower right. In Module1, file a/b/File1 is replaced with the one from the partial application zipped file. In Module2.war, there is a new file: m/n/File3. And Module3.jar has been replaced, so i/j/File1 and k/l/File2 are removed from the application and r/s/File4 and t/u/File5 are added.



File Deletions

- In a partial application, file deletion is specified using a special metadata file
  - META-INF/ibm-partialapp-delete.props
    - Can be at Application Scope or Module Scope

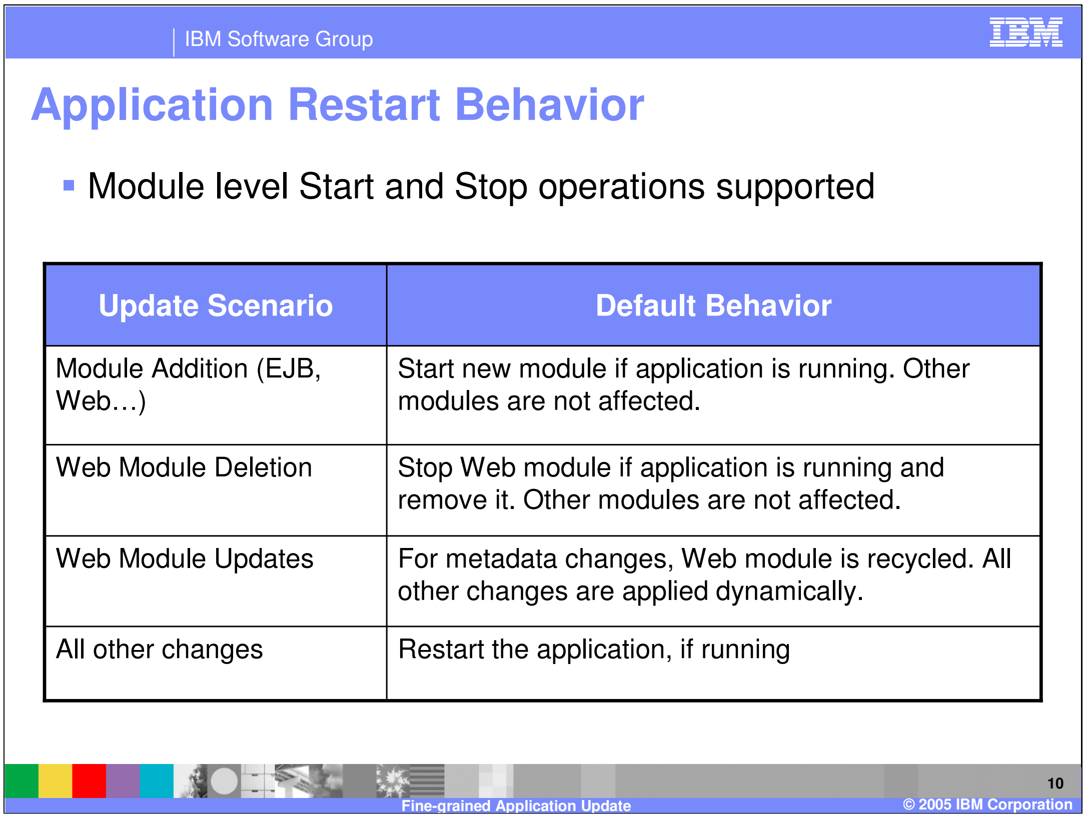
Application EAR
Module1.jar
a/b/File1
a/b/File2
e/f/File1
Module2.war
i/j/File1
k/l/File2

Partial Application .ZIP
/Module1.jar/META-INF/ibm-partialapp-delete.props

a/b/File*

Application EAR (final)
Module1.jar
e/f/File1
Module2.war
i/j/File1
k/l/File2

In a partial application, file deletions can be specified using a special metadata file META-INF/ibm-partialapp-delete.props at application or module scope in the hierarchical structure. Each line in the file represents a pattern of application or module scoped URIs that are to be deleted. The pattern can be specified as a regular expression using constructs defined in java.util.regex.Pattern.



Application Restart Behavior

- Module level Start and Stop operations supported

Update Scenario	Default Behavior
Module Addition (EJB, Web…)	Start new module if application is running. Other modules are not affected.
Web Module Deletion	Stop Web module if application is running and remove it. Other modules are not affected.
Web Module Updates	For metadata changes, Web module is recycled. All other changes are applied dynamically.
All other changes	Restart the application, if running

Prior to V6, WebSphere Application Server only allowed starting and stopping of the entire application. As a part of fine-grained application update support, the WebSphere Application Server runtime is enhanced to support module level start and stop operations, making it possible to recycle only parts of a running application when it is updated.

For Module Additions, modules can be added on the fly without stopping the running application. However, if there are security roles defined in the deployment descriptor for the module, and security provider does not support dynamic updates, then the application is recycled.

For Web Module Deletions, the module is stopped and the module files are deleted. For other modules (EJB, connector), the application is recycled.

For Updates of Web modules, if metadata is changed, then the module is recycled. If only non-metadata artifacts (servlets, JSPs) are updated, nothing is recycled.

For Update of other modules, the entire application is recycled.

The following rules apply to module level start/stop operations:

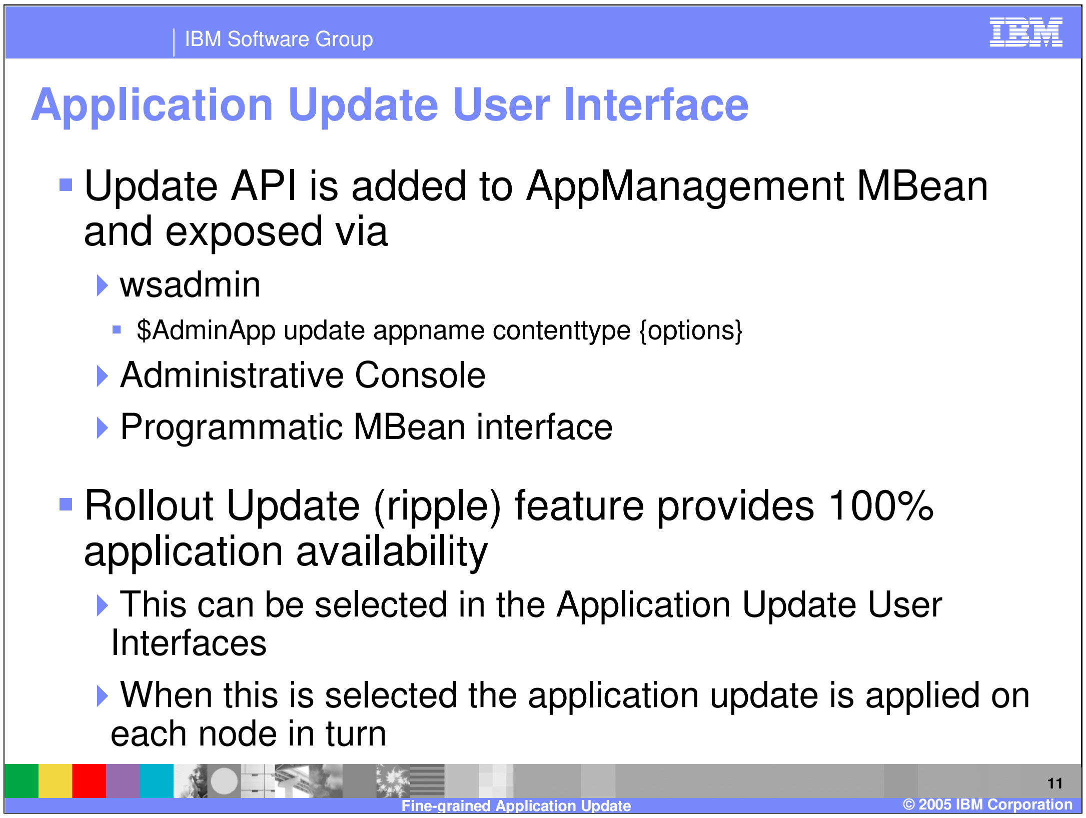
Starting a module makes module components accessible to clients. A module can be started only if its parent application is running. Starting an application, starts all the modules deployed on that server, but the converse is not true.

Module start operation is supported on all module types such as EJB, Web and Connector.

Stopping an application makes module components inaccessible to clients. Stopping the last running application module on an application server does not stop the parent application on that server.

Module stop operation is supported only for Web modules, and *only* if the WAR classloader policy for the deployed application is MODULE (i.e. there is a classloader per WAR module). If the WAR classloader policy is APPLICATION (i.e. all WARs of the application share a single classloader which is the application classloader) then the entire application needs to be recycled (stopped and re-started).

Please note that you cannot start or stop a module manually. It only happens as part of update operation.



IBM Software Group

Application Update User Interface

- Update API is added to AppManagement MBean and exposed via
  - wsadmin
    - $AdminApp update appname contenttype {options}
  - Administrative Console
  - Programmatic MBean interface
- Rollout Update (ripple) feature provides 100% application availability
  - This can be selected in the Application Update User Interfaces
  - When this is selected the application update is applied on each node in turn

11

Fine-grained Application Update

© 2005 IBM Corporation

The update function is added to the AdminApp object exposed by the wsadmin command line utility. It is also exposed to using the Administrative Console, and the wizard is very similar to the install wizard. You can also install applications using the programmatic MBean interface.

The command line syntax is **$AdminApp update appname contenttype {options}**

where contenttype can be file, module, partialapp, or application,

and where options are:

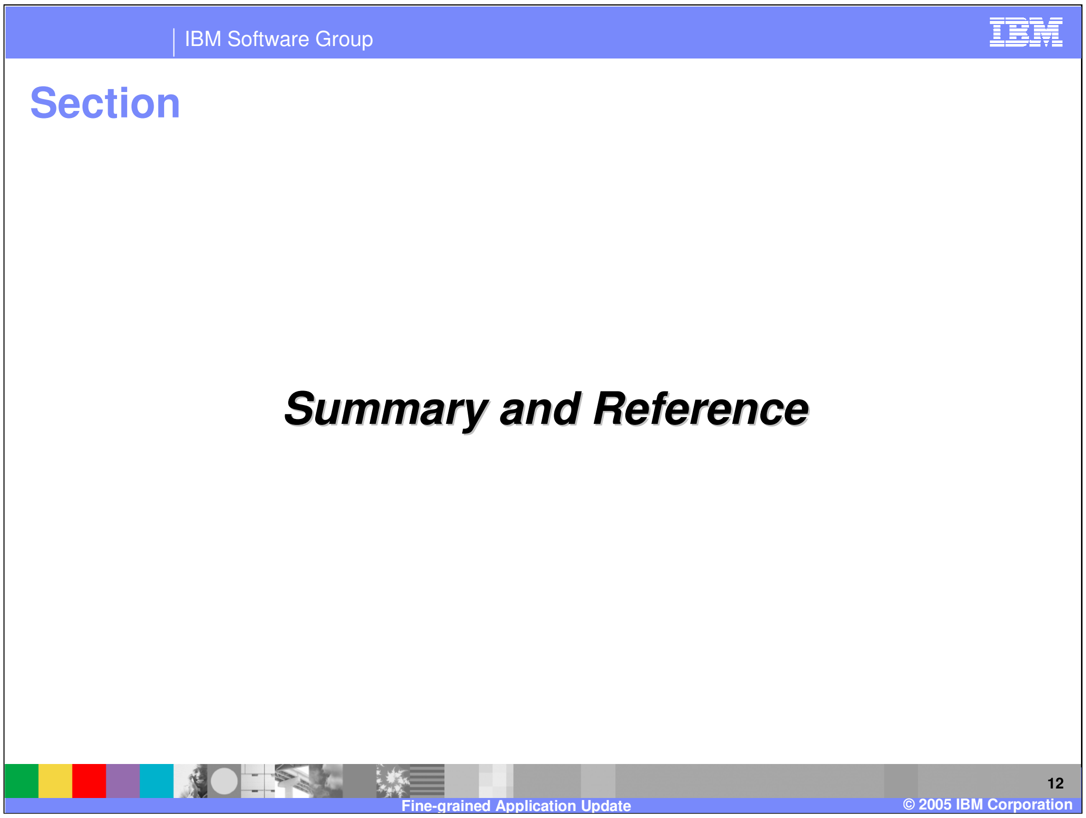
-contentURI : URI of the file/module

-contents : filename of the contents to be updated

-operation : add, delete, update, or addupdate

-recycle : can be default, all, none

The next bullet point, about Rollout Update, is for the Network Deployment package. With this option, the application updates are applied to one node at a time, so that the application is continuously available. This function is provided by the “rollout update” button on the Administrative Console .



IBM Software Group

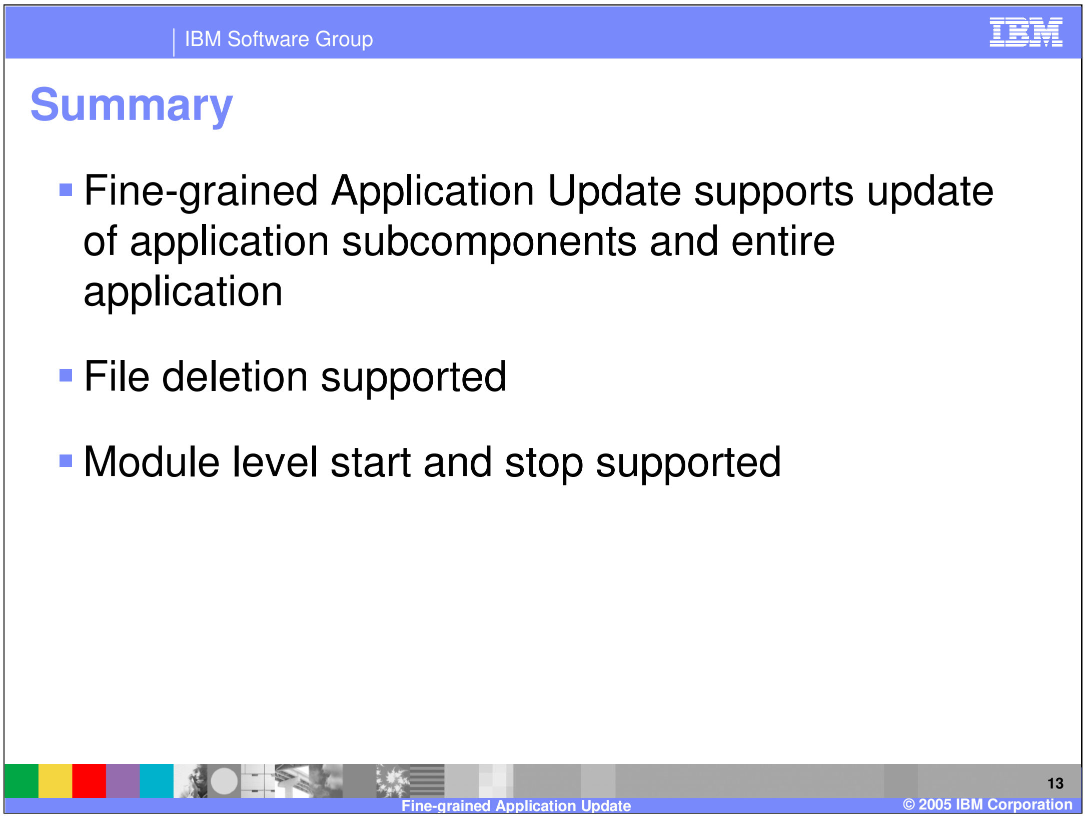
Section

Summary and Reference

12

Fine-grained Application Update

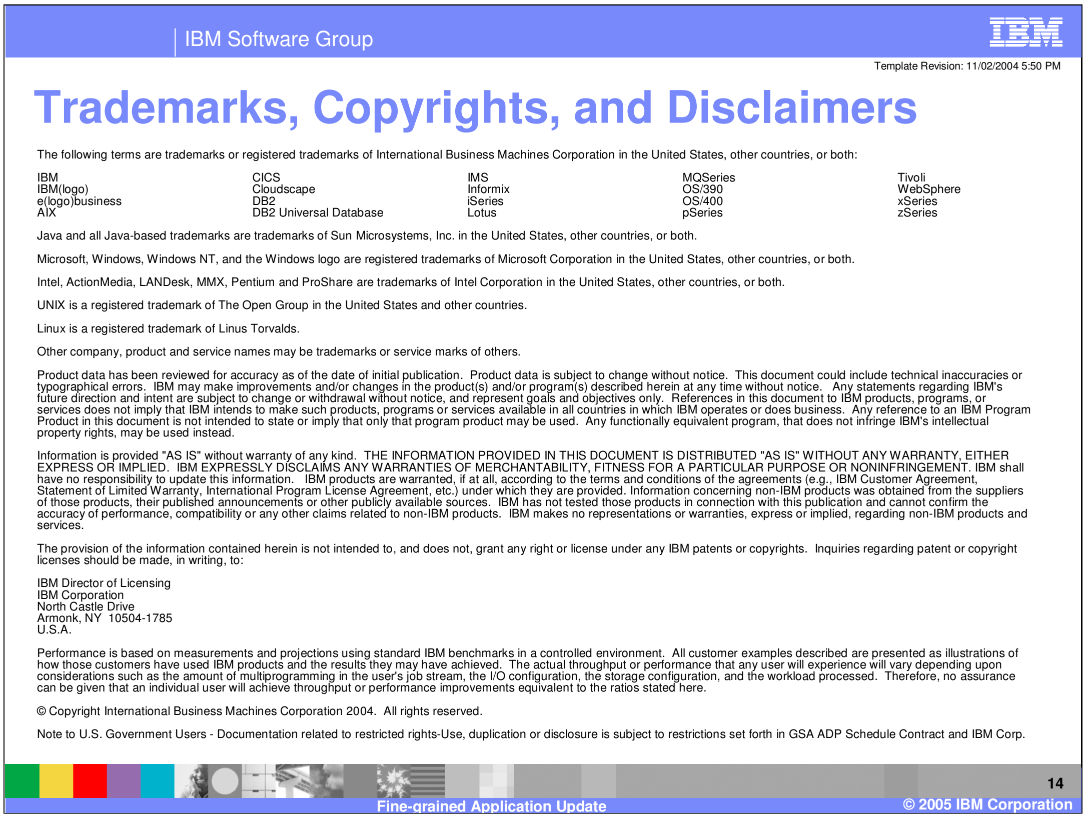
This section will review what was covered.



Summary

- Fine-grained Application Update supports update of application subcomponents and entire application
- File deletion supported
- Module level start and stop supported

In summary, Fine-grained Application Update allows updates of application subcomponents, as well as updating the entire application. WebSphere Application Server supports file deletion, and also supports module level start and stop during application update.



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