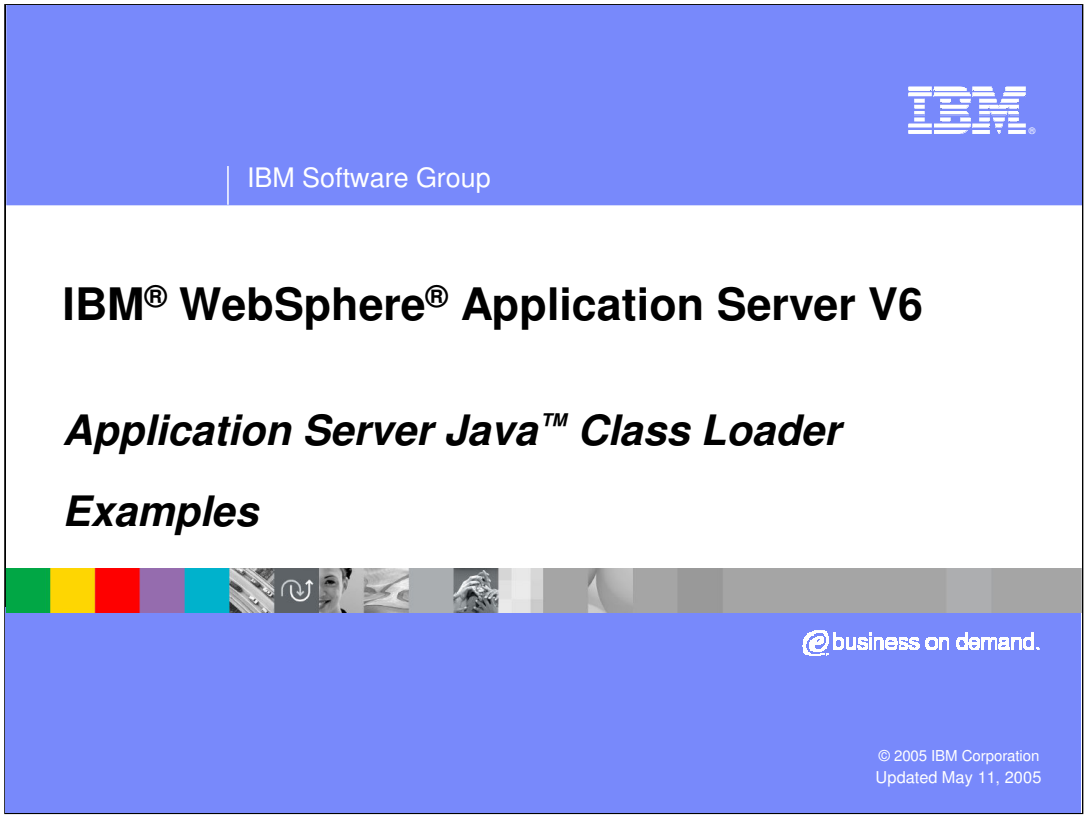
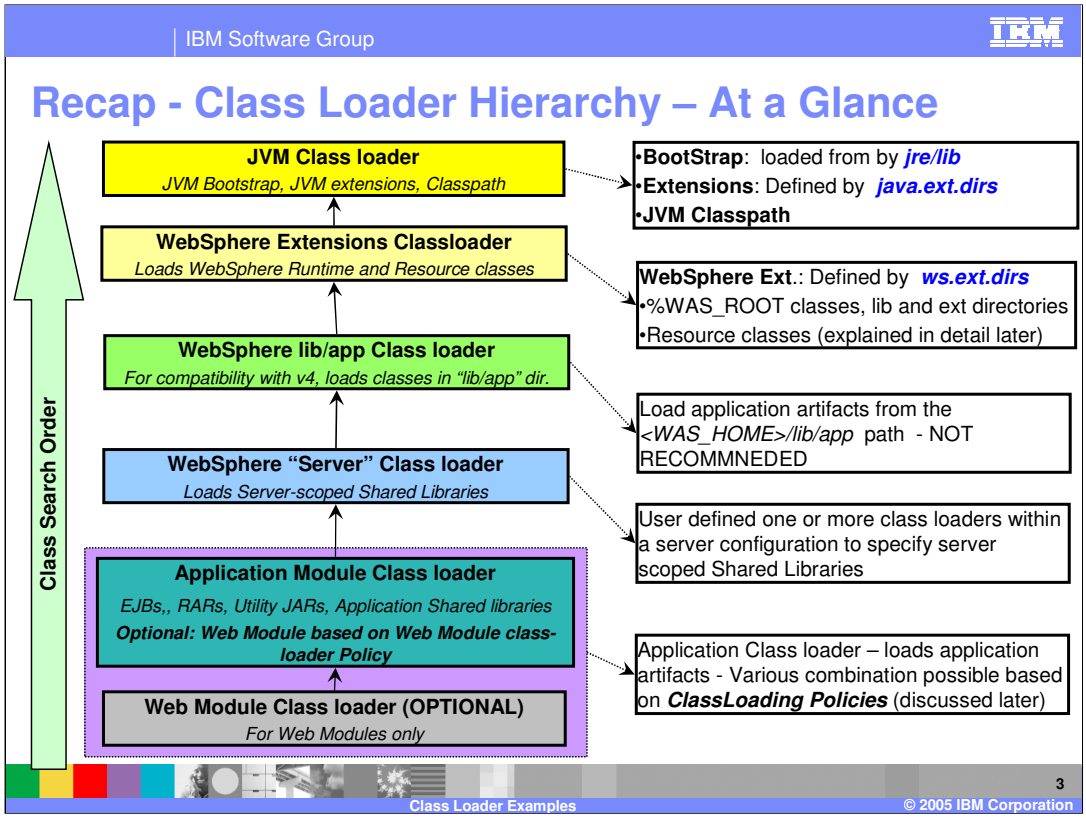


This presentation will focus on Examples of the Application Server Java Class Loader.

Goals

- Provide Class loaders examples with different policies and delegation modes
- Pre-requisite:
  - Understanding of WebSphere V6 Class loader Overview

The goal of this presentation is to provide some class loader examples with different policies and delegation modes.



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.

This hierarchy was discussed in detail in the Class loader overview presentation.

This presentation will provide examples of some class loader scenarios.

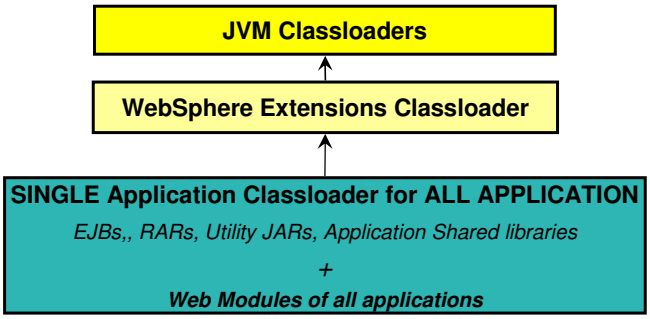
Section

Class Loader Policies - Examples

This section will provide examples of Class Loader Policies.

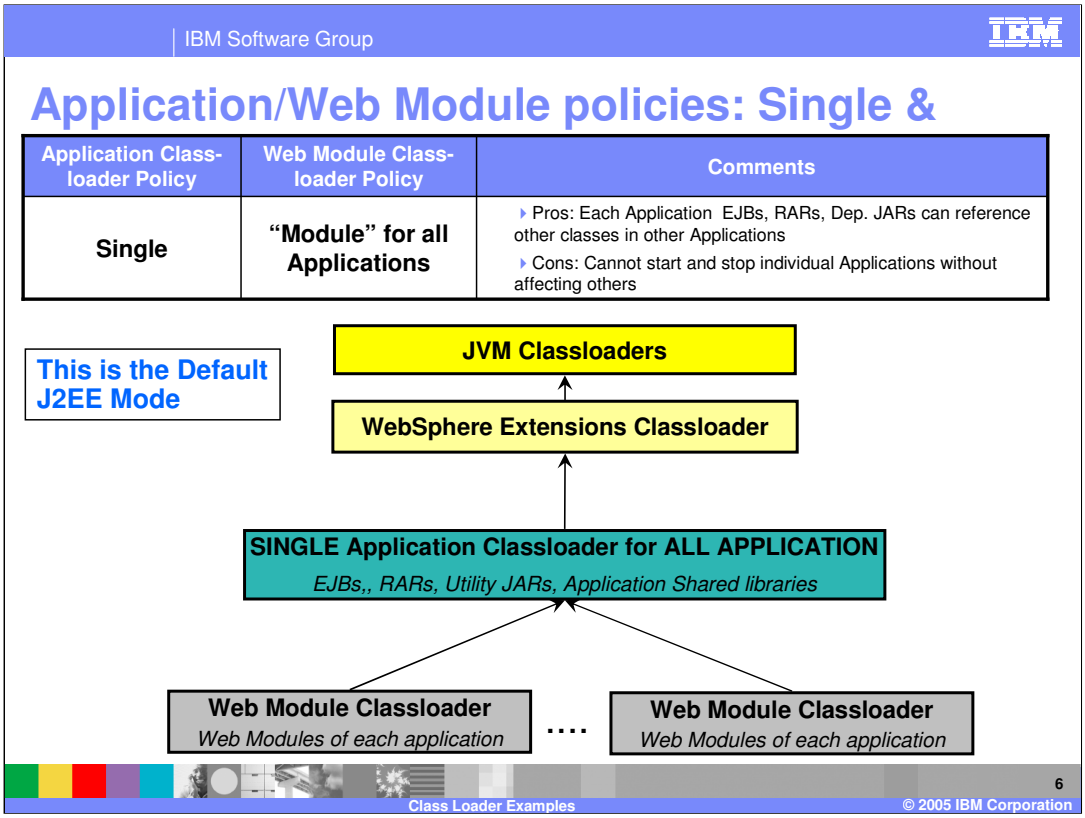
Application/Web Module policies: Single &

Application Class-loader Policy	Web Module Class-loader Policy	Comments
Single	**"Application" for all applications**	▸ Pros: Each Application EJBs, RARs, Dep. JARs can reference other classes in other Applications ▸ Cons: Cannot start and stop individual Applications without affecting others



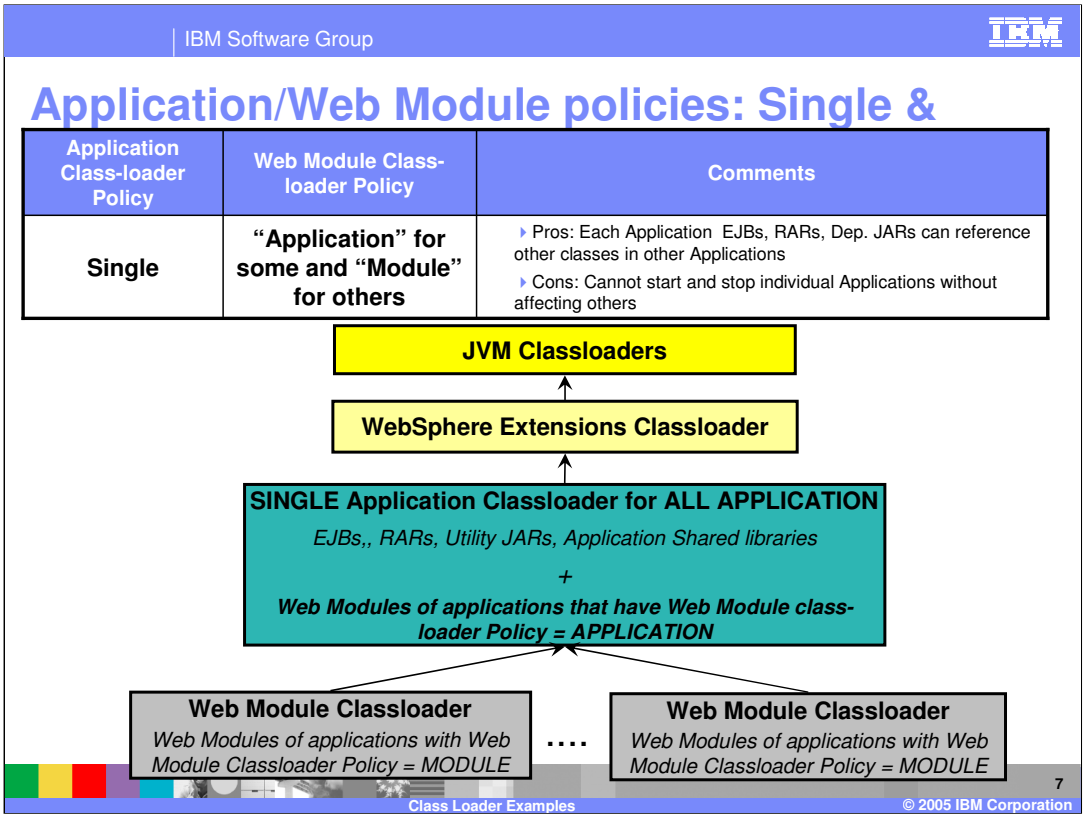
Not Recommended to have Single Application Class loader
You loose isolation, dynamic reloading

Shown here is an example hierarchy of an Application Class loader policy of **Single** and a Web module class loader policy of **Application** for all the J2EE applications. Using these class loader options sacrifices the isolation and dynamic reloading features.



Shown here is an example hierarchy of an Application Class loader policy of **Single** and a Web module class loader policy of **Module** for all the J2EE applications.

With these policies, the Web module class loader is loaded by its own separate class loader. This is the default J2EE class loader mode.

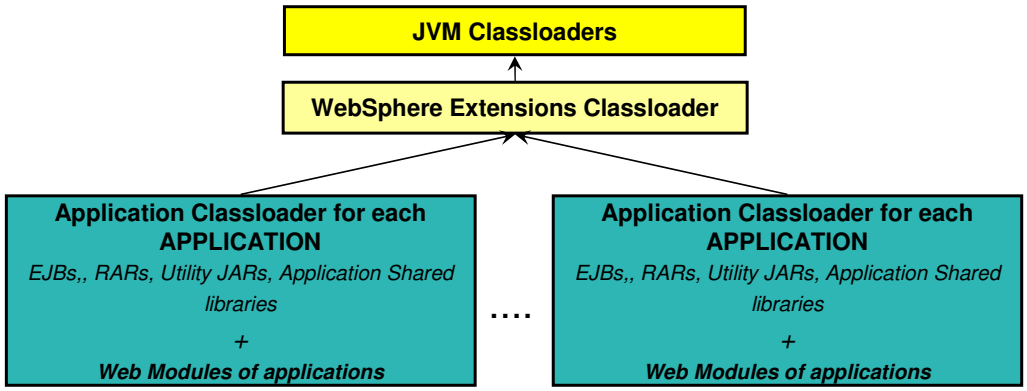


Shown here is an example hierarchy of the Application Class loader policy of **Single** and Web module class loader policy of **Module** for some of the J2EE applications and **Application** for the remaining applications.

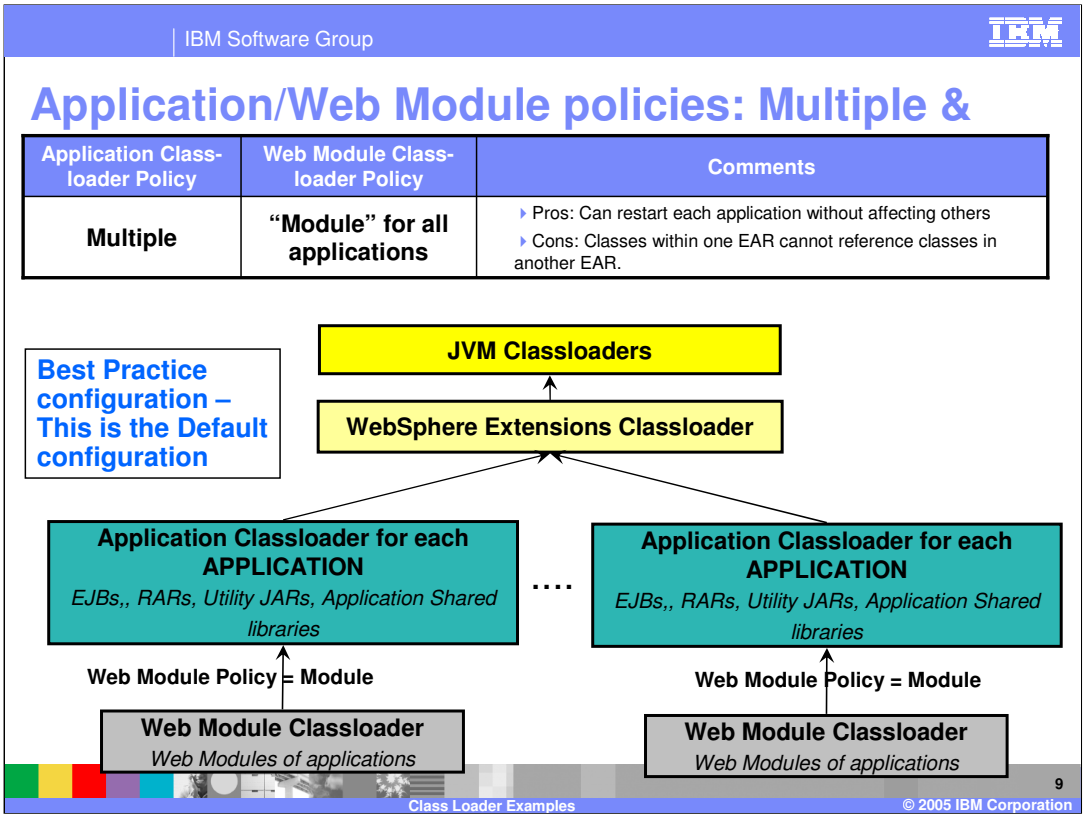
With these policies, the Web module class loader for **Module** policy option is loaded by its own separate class loader. For a Web module class loader policy of **Application**, they are loaded by the Application Class loader.

Application/Web Module policies: Multiple &

Application Class-loader Policy	Web Module Class-loader Policy	Comments
Multiple	**"Application" for all applications**	▸ Pros: Can restart each application without affecting others ▸ Cons: Classes within one EAR cannot reference classes in another EAR

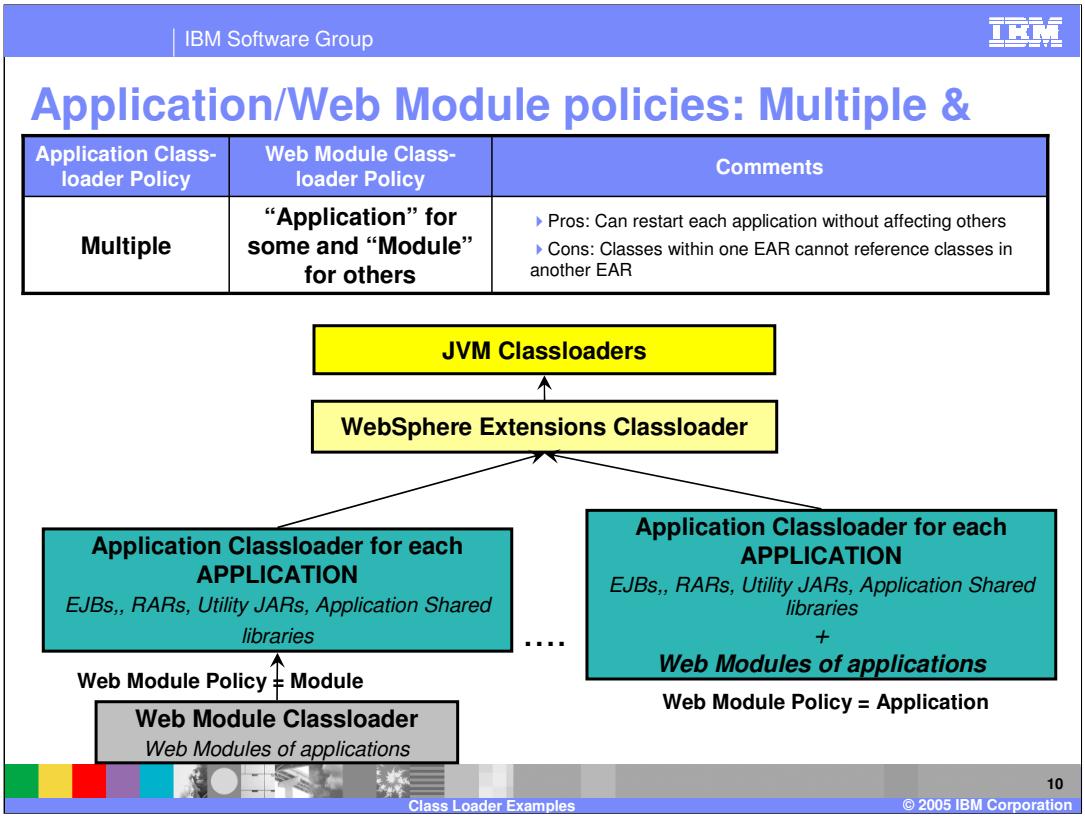


Shown here is an example hierarchy of an Application Class loader policy of **Multiple**. Each J2EE Application is loaded by its own class loader. Additionally, with the Web Module class loader policy of **Application**, the Web modules of those applications are loaded by the same class loader as the rest of the J2EE application classes.



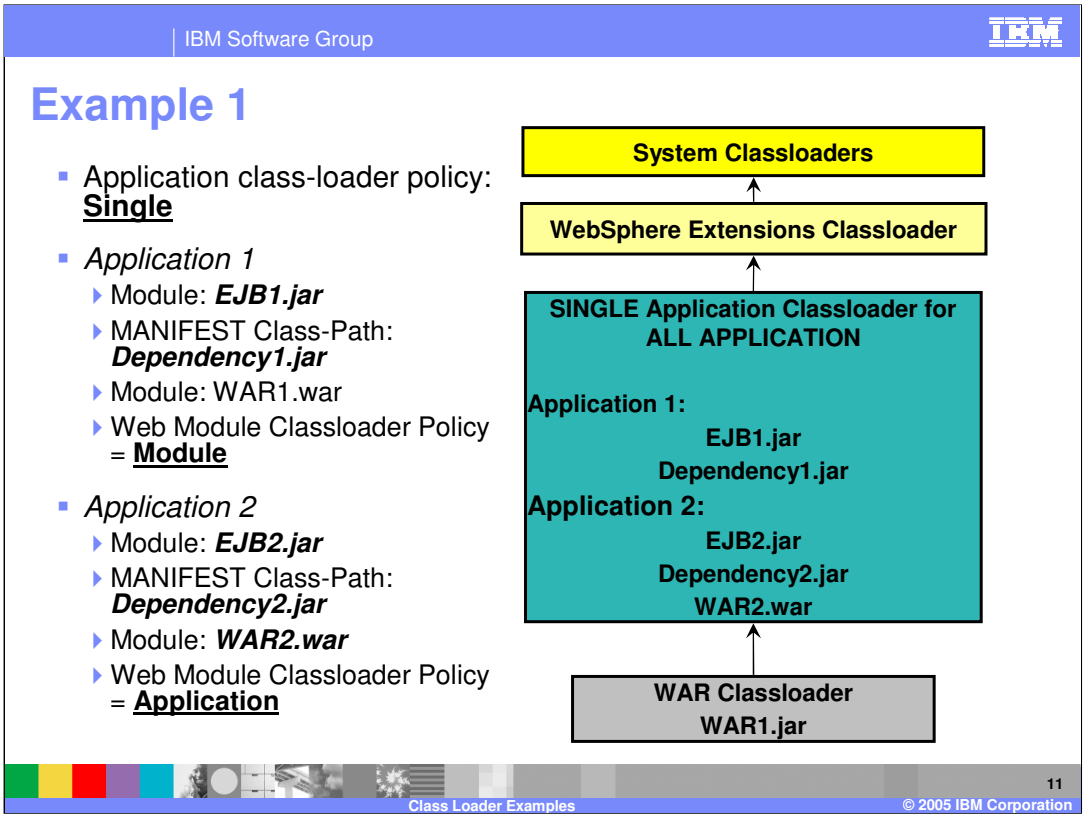
Shown here is an example hierarchy of an Application Class loader policy of **Multiple**. Each J2EE Application is loaded by its own class loader. However, note that the Wed Module class loader policy is **Module**. As a result, each Web module is loaded by separate class loader, lower in the class loader hierarchy.

This is the default class loader configuration of WebSphere Application Server V6.



Shown here is an example hierarchy of an Application Class loader policy of **Multiple**. Each J2EE Application is loaded by its own class loader. However, some of the Applications have Web Module class loader policy is **Module**. As a result, for those applications, the Web module is loaded by a separate class loader, lower in the class loader hierarchy.

For the remaining applications that have a Web module class loader policy of **Application**, those Web module classes are loaded by the same class loader that loads the J2EE application.

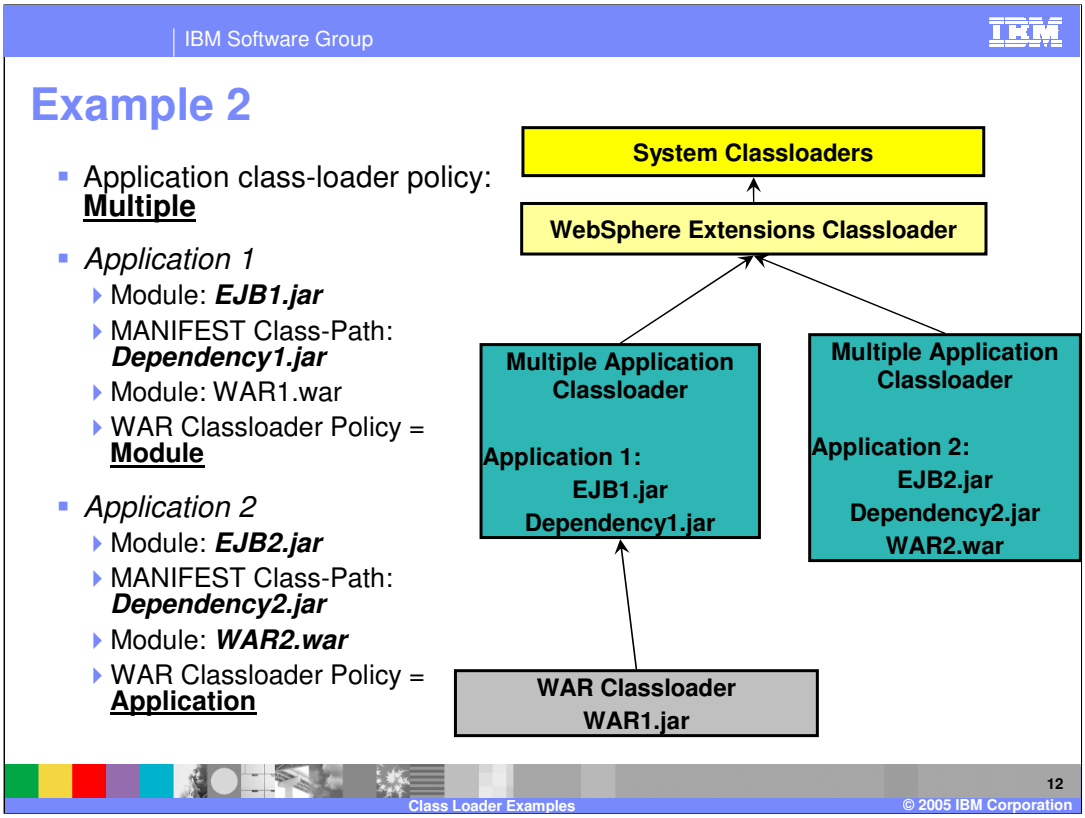


The example shown here has the Application class loader policy at the server level set to **Single**.

As a result, the classes for both the applications are loaded by the Single class loader. However, the Web module class loader policy defines how the Web modules will be loaded.

For Application 1, the Web module class loader policy is Module. As a result, each Web module of application 1 will have its own separate class loader, as shown by the WAR1 Web module.

For Application 2, the Web module class loader policy is Application. As a result, all Web modules of application 2 will be loaded by the same class loader that loaded Application 2 classes, as shown by the WAR2 Web module.



The example shown here has the Application class loader policy at the server level set to **Multiple**.

As a result, the classes for both the applications are loaded by their own separate class loader, as shown by the class loaders of Application 1 and 2.

However, the Web module class loader policy defines how the Web modules will be loaded.

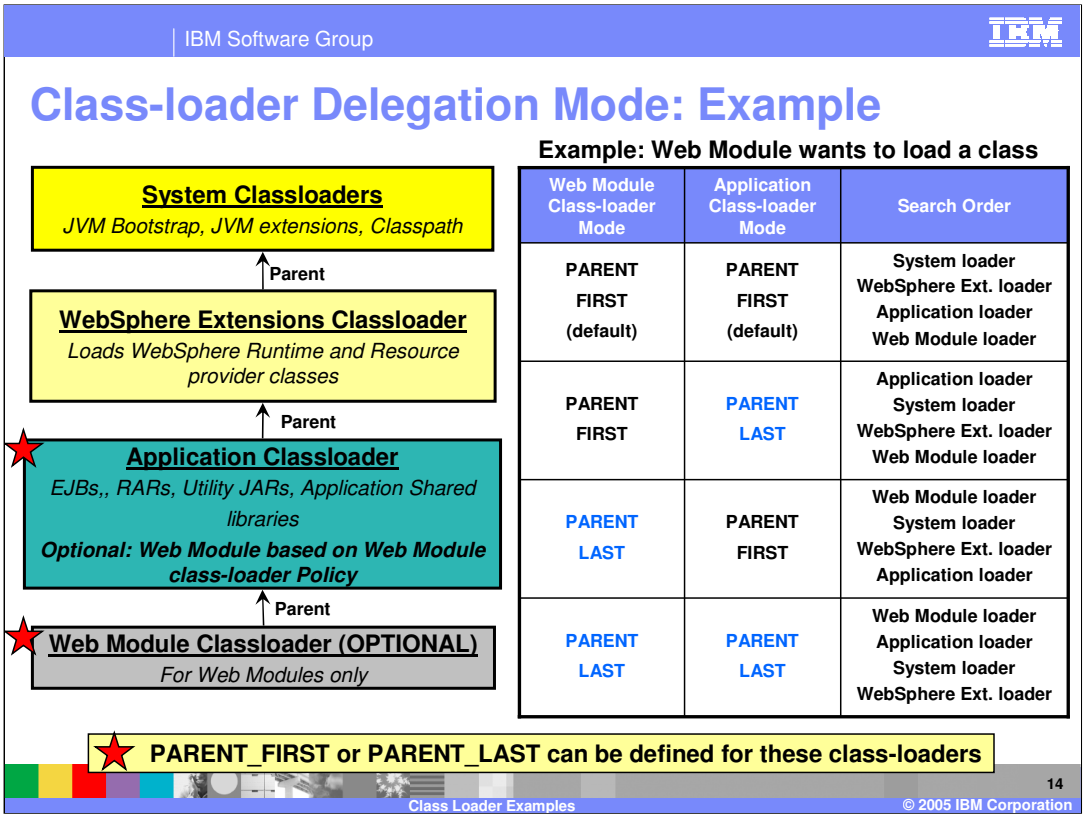
For Application 1, the Web module class loader policy is **Module**. As a result, each Web module of application 1 will have its own separate class loader, as shown by the WAR1 Web module.

For Application 2, the Web module class loader policy is **Application**. As a result, each Web modules of application 2 will be loaded by the same class loader that loaded Application 2 classes, as shown by the WAR2 Web module.

Section

Class Loader Delegation - Example

This section will provide some examples of Class Loader Delegation.



Delegation or search mode is defined at the Application class loader and at the Web module class loader levels. Based on the different delegation modes, the table shows the search order path. At each level, if the delegation mode is PARENT FIRST, the search goes to the parent. If the delegation mode is PARENT_LAST, the current class loader is searched before the parent. These delegation modes help when an application requires classes loaded from its own class loader rather than having it loaded by WebSphere supplied classes. This provides flexibility.

The default mode is the PARENT_FIRST for both the application class loader and the Web module class loader.

Section

Summary and References

This section will provide a summary of the concepts covered by this presentation.

Summary

- Provided some examples of Class-loader options with different policies and delegation modes

In summary, this presentation has provided some examples of class loader options with different policies and delegation modes.

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.