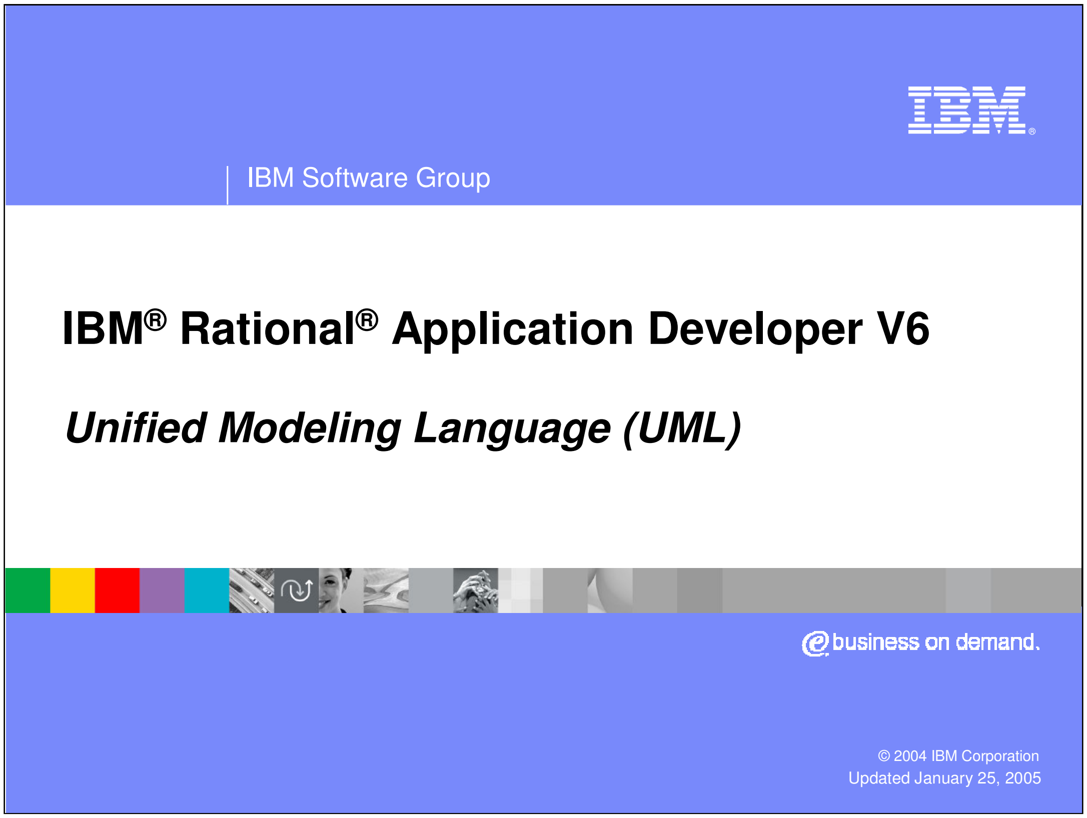


IBM Software Group

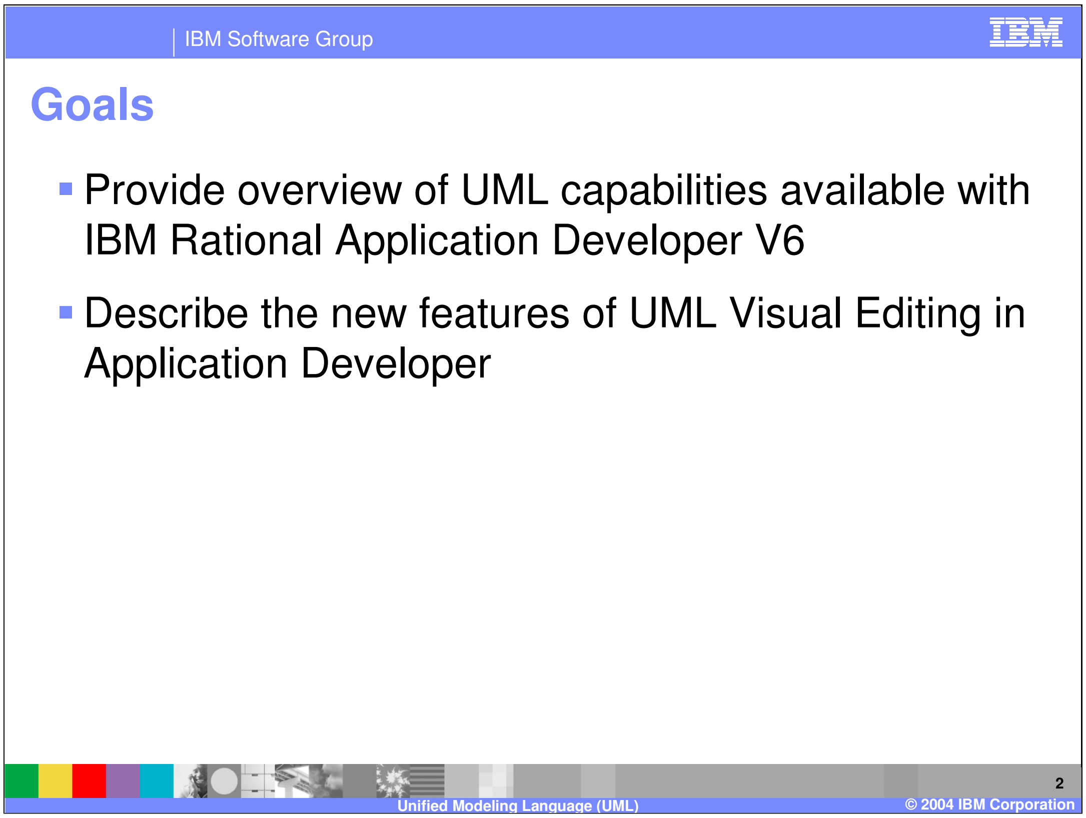
IBM® Rational® Application Developer V6

Unified Modeling Language (UML)

business on demand.

Updated January 25, 2005

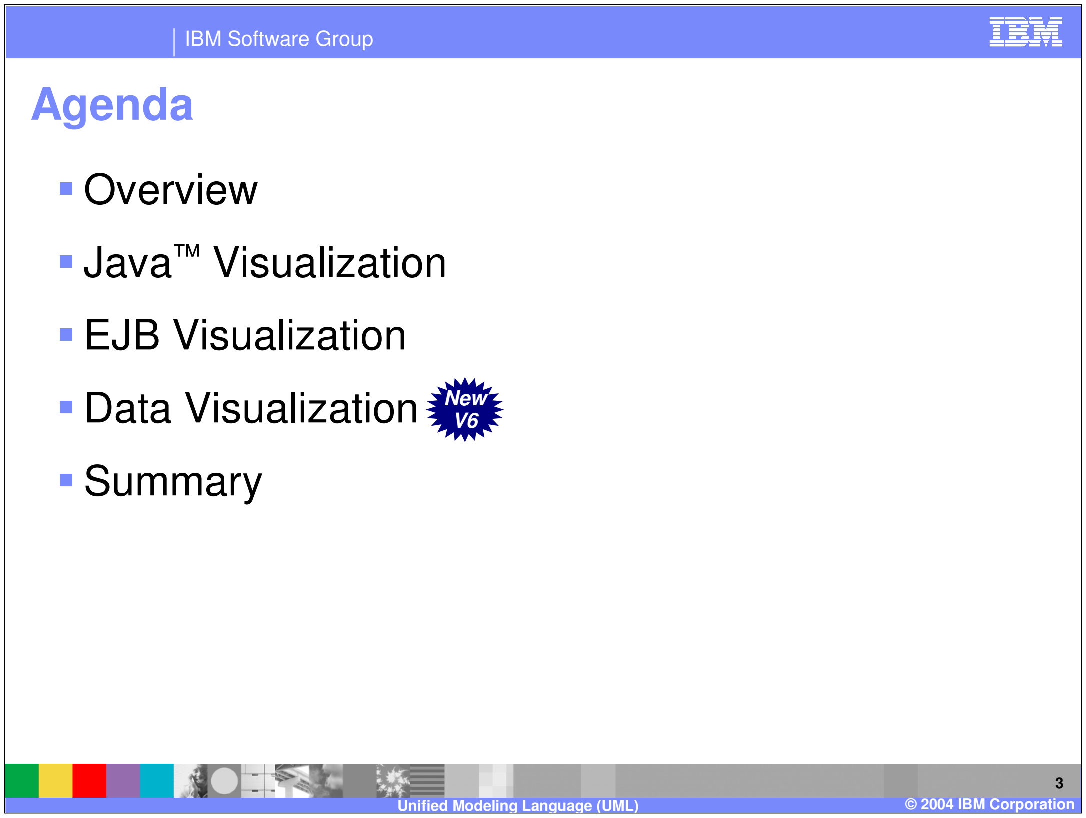
This presentation, will focus on UML visualization capabilities available in V6 of Application Developer.



Goals

- Provide overview of UML capabilities available with IBM Rational Application Developer V6
- Describe the new features of UML Visual Editing in Application Developer

The goals for this presentation are to provide an overview of UML visualization capabilities available in V6 of Application Developer, as well as highlight some of the new features offered in this release.



Agenda

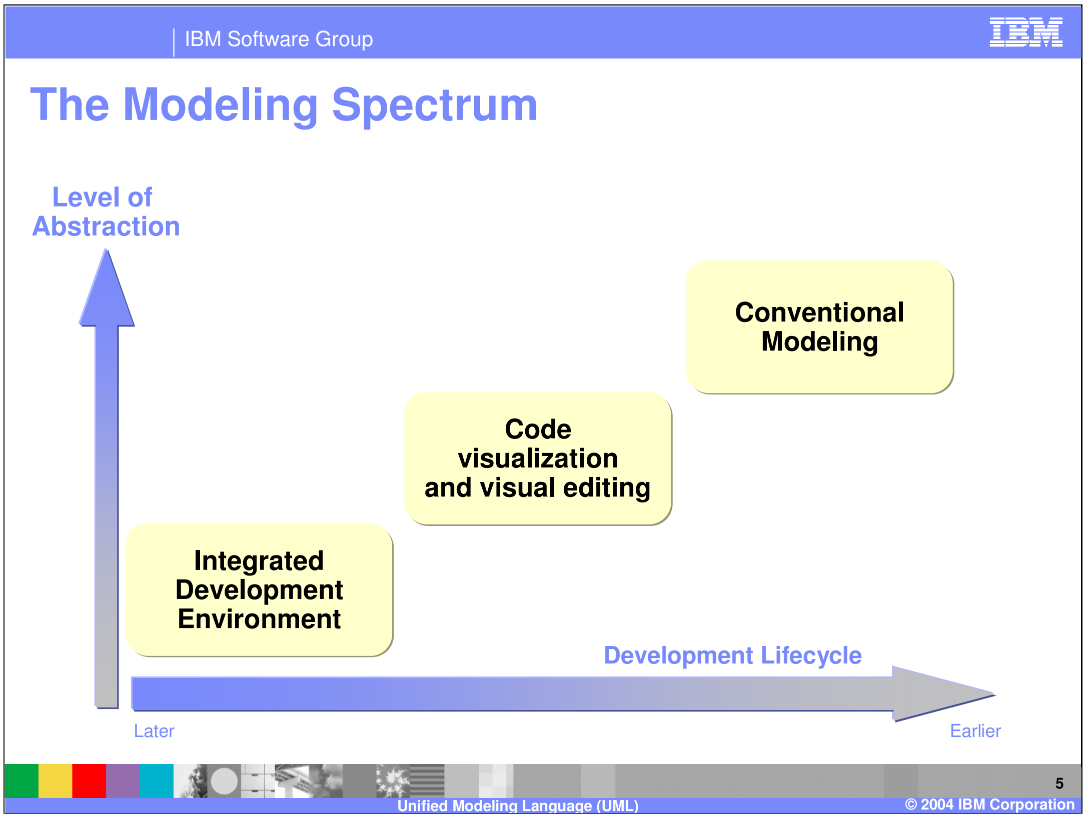
- Overview
- Java™ Visualization
- EJB Visualization
- Data Visualization New V6
- Summary

The agenda for this presentation is to start with a brief overview of general UML visualization capabilities. After this overview the remaining part of the presentation is divided into three sections that will describe specific features available with the various types of visualization capabilities. Specifically, these include: EJB, Java, and Data visualization.

Section

Overview

The next section will provide an overview of the UML visualization features available with Application Developer V6.



The Modeling Spectrum

Level of Abstraction

Conventional Modeling

Code visualization and visual editing

Integrated Development Environment

Development Lifecycle

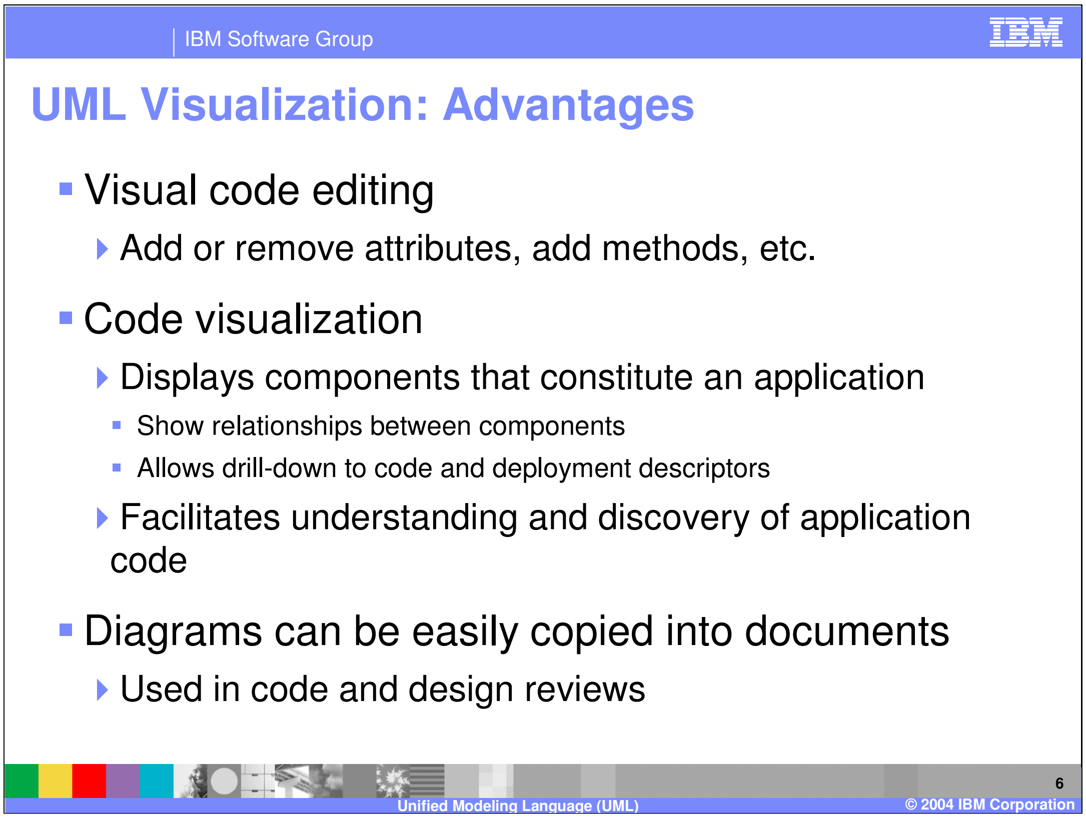
Later

Earlier

Before discussing some of the specific features available in application developer, let’s begin by talking a little bit about modeling in general. The term modeling is sometimes used to describe a range of development activities when building a software system, and the intent of this discussion is to provide a context for the UML functionality that is available with Application Developer. This slide shows a modeling spectrum that is described in a white paper entitled “The Value of Modeling”, and can be found at the following URL: http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/ValueOfModeling.pdf.

Strictly speaking, you generally do not classify tools provided with a basic IDE (such as editors, wizards, and views) as something that helps you develop or model your application. However, these tools do raise the level of abstraction in the development lifecycle and as such can be thought of as a very basic level of model-driven development. The next level of modeling is code visualization and visual editing. At this level, code is abstracted further such that the developer can visualize code with a set of diagrams instead of looking at the code as text within an editor window of the IDE. With code visualization, the diagram is built based upon existing code to better understand and communicate the code. Visual editing is also a further abstraction from basic code development that is based upon simple text editing. With visual editing, developers can build and edit code by manipulating diagrams rather than explicitly typing code in a editor window. Finally, conventional modeling is a phase when visual models are created to communicate the high level architecture of the software product. These visual models are built before there is code written, but can be used with an IDE to help generate initial code from the model.

The UML Visualization capabilities provided with Application Developer fall under the category of code visualization and visual editing. This will be the focus of the remaining part of the presentation. Those who are interested in doing conventional modeling should look into the Rational Studio Architect product.



IBM Software Group

UML Visualization: Advantages

- Visual code editing
  - Add or remove attributes, add methods, etc.
- Code visualization
  - Displays components that constitute an application
    - Show relationships between components
    - Allows drill-down to code and deployment descriptors
  - Facilitates understanding and discovery of application code
- Diagrams can be easily copied into documents
  - Used in code and design reviews

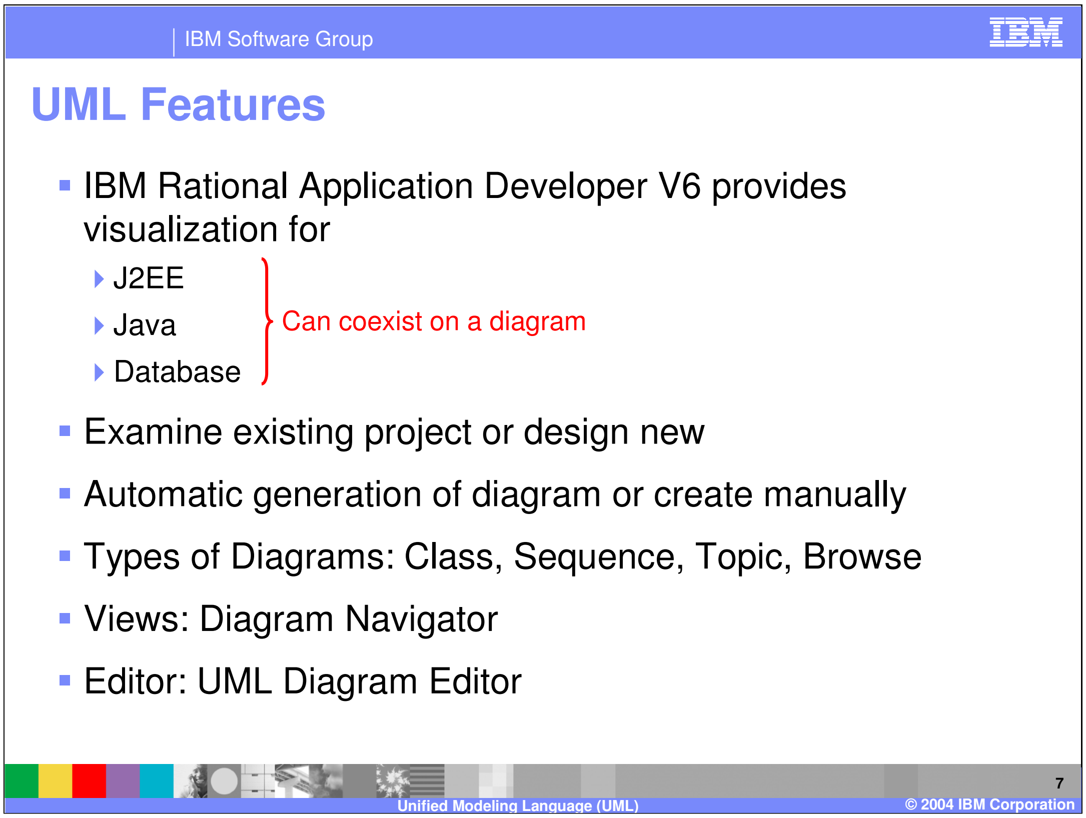
6

Unified Modeling Language (UML)

UML visualization features provided by Application Developer offers the advantage of code visualization and visual editing.

Visual editing offers the ability to build code without explicitly typing the code in a text editor, but rather adding attributes and methods through the diagram editor in a visual manner.

The code visualization capabilities allows developers to view an existing code base as either class or sequence diagrams. This type of visualization helps developers understand and communicate application code by showing relationships between components, and also provides insight into items included with deployment descriptors as is the case with J2EE visualization. The diagrams used to visualize code can also be copied into design documents and used in design and code reviews.



IBM Software Group

UML Features

- IBM Rational Application Developer V6 provides visualization for
  - J2EE
  - Java
  - Database

Can coexist on a diagram

- Examine existing project or design new
- Automatic generation of diagram or create manually
- Types of Diagrams: Class, Sequence, Topic, Browse
- Views: Diagram Navigator
- Editor: UML Diagram Editor

7

Unified Modeling Language (UML)

© 2004 IBM Corporation

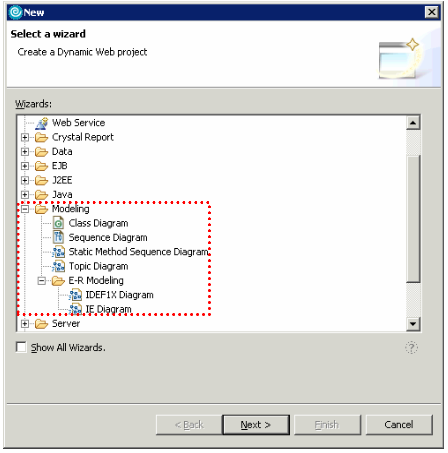
There are several types of visualization available with Application Developer version 6. As in the prior releases, version 6 still offers J2EE and Java visualization capabilities. This presentation will touch on some of the new features associated with each of these types of visualization. Also, new in this release is support for data visualization. Data visualization enables developers to edit and view their database schemas in a variety of diagram notations.

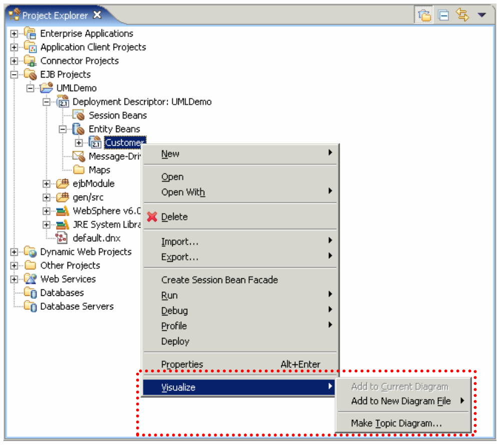
There are some visualization semantics and actions that are unique to the type of artifacts that are being visualized (J2EE, Java, or data). However, regardless of the type of project the visualization capabilities allow you to examine existing projects or build a new project from the start. To get started using the visualization diagrams you can create the particular diagram manually (File > New > Other > Modeling > …), or in some cases the diagram will be generated by default by running through a particular wizard.

There are several types of diagrams to help visualize your project. These diagrams include: class, sequence, topic, and browse. The type of diagram you choose will depend on what you are building or what type of information you would like to learn or show about your project. Later in this presentation you will learn more about each of these diagrams.

UML Visualization: Overview

New > Other… > Modeling





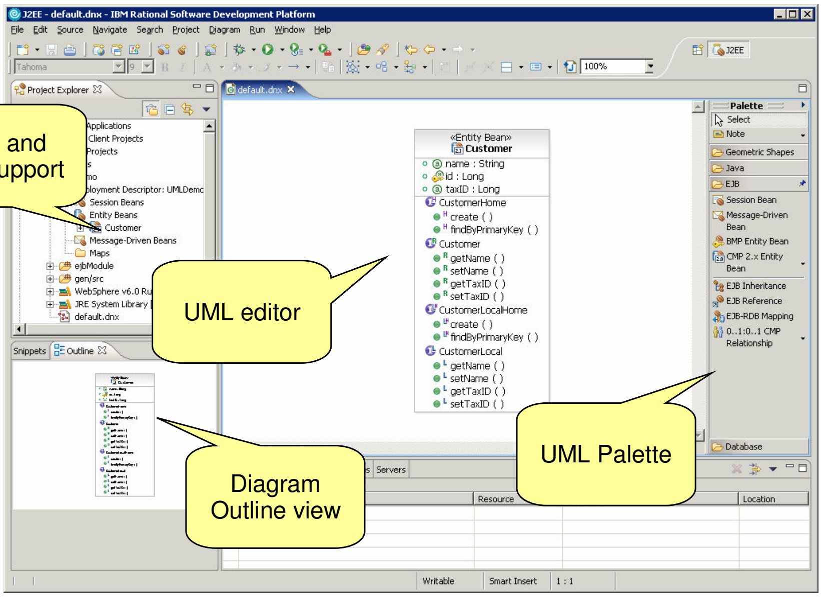
Visualization **option on**

Java, EJB, and Data resources

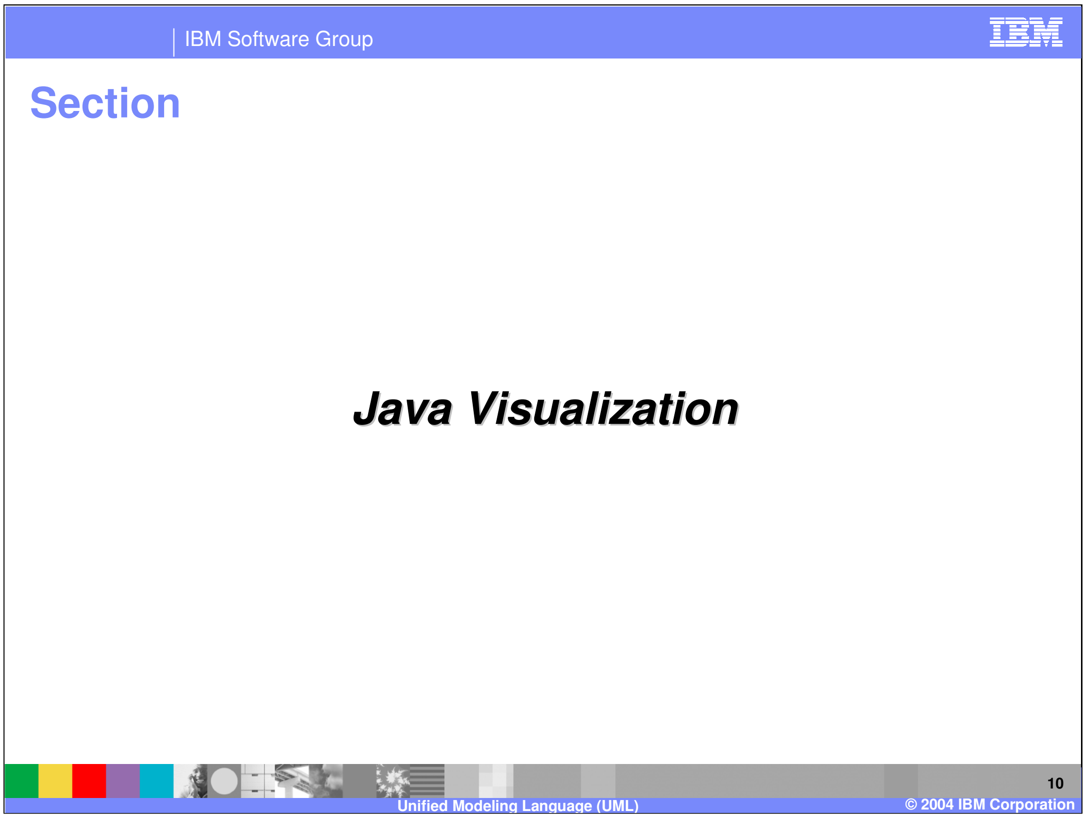
There are a couple ways to get started using the visualization capabilities. If for example you are using a class diagram to visually edit or build your project artifacts, then you will likely create a new class diagram by selecting File > New > Other…> Modeling > Class Diagram. In addition to this, several create wizards by default create a new class diagram at the end of the create process.

Alternatively if you would like to visualize an existing project you can create a class diagram (or another type of diagram) by right clicking on the artifact in the project explorer view and selecting the Visualize option.

UML Visualization: Overview (Continued)



This slide provides an overview of a workspace that you might see if you were using the UML visualization capabilities. There are several things to note on this slide. The center area is the UML editor. Built into this editor is a palette that is used to drag and drop artifacts onto the editor work area. The items that appear in the palette are specific to the type of project that is being visualized. The Outline view helps you to see the area of the diagram that you are zoomed into in the editor area. Finally, you can drag and drop project artifacts onto the editor from the Project Explorer view add these items in the diagram.



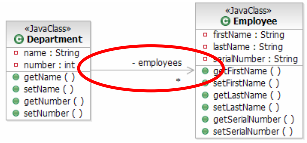
Section

Java Visualization

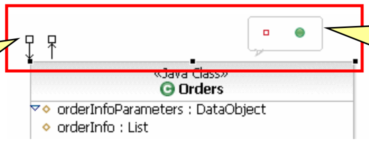
The next section will provide an overview of the Java UML visualization features available with Application Developer V6.

Java Visualization: New Features

- New directed association tool on palette
  - Provides the ability to show a field as an association and vice versa



- Diagram assistants
  - Hover over element
  - Select element and hit space key

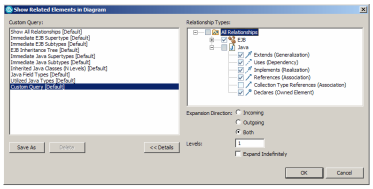


Alternate way to create/show connectors

Alternate way to create fields and methods

Java Visualization: New Features (Continued)

- Improved support for
  - Auto Navigate To Source
  - Filter > Show Related Elements



Show Related Elements in Diagram

Custom Query:

Show All Relationships [Default]
Immediate EJB Supertype [Default]
Immediate EJB Subtypes [Default]
EJB Inheritance Tree [Default]
Immediate Java Supertypes [Default]
Immediate Java Subtypes [Default]
Inherited Java Classes (N Levels) [Default]
Java Field Types [Default]
Utilized Java Types [Default]
Custom Query [Default]

Relationship Types:

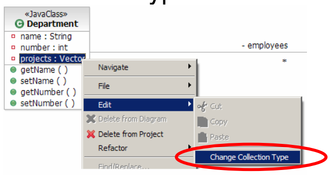
- □ All Relationships
  - ☑ EJB
  - □ Java
    - ☑ Extends (Generalization)
    - ☑ Uses (Dependency)
    - ☑ Implements (Realization)
    - ☑ References (Association)
    - □ Collection Type References (Association)
    - ☑ Declares (Owned Element)

Expansion Direction: ○ Incoming ○ Outgoing ● Both

Levels: 1

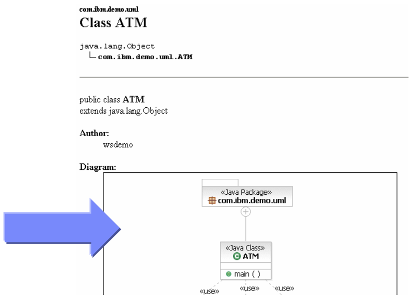
□ Expand Indefinitely

  - Visualization of collection types
    - Contained type to be something other than Object



Java Visualization: New Features (Continued)

- Refactor options available from UML Diagram
- Javadoc can be generated to include the UML diagram images
  - @viz.diagram tag
  - Project > Generate Javadoc with Diagrams > Automatically



com.ibm.demo.uml

Class ATM

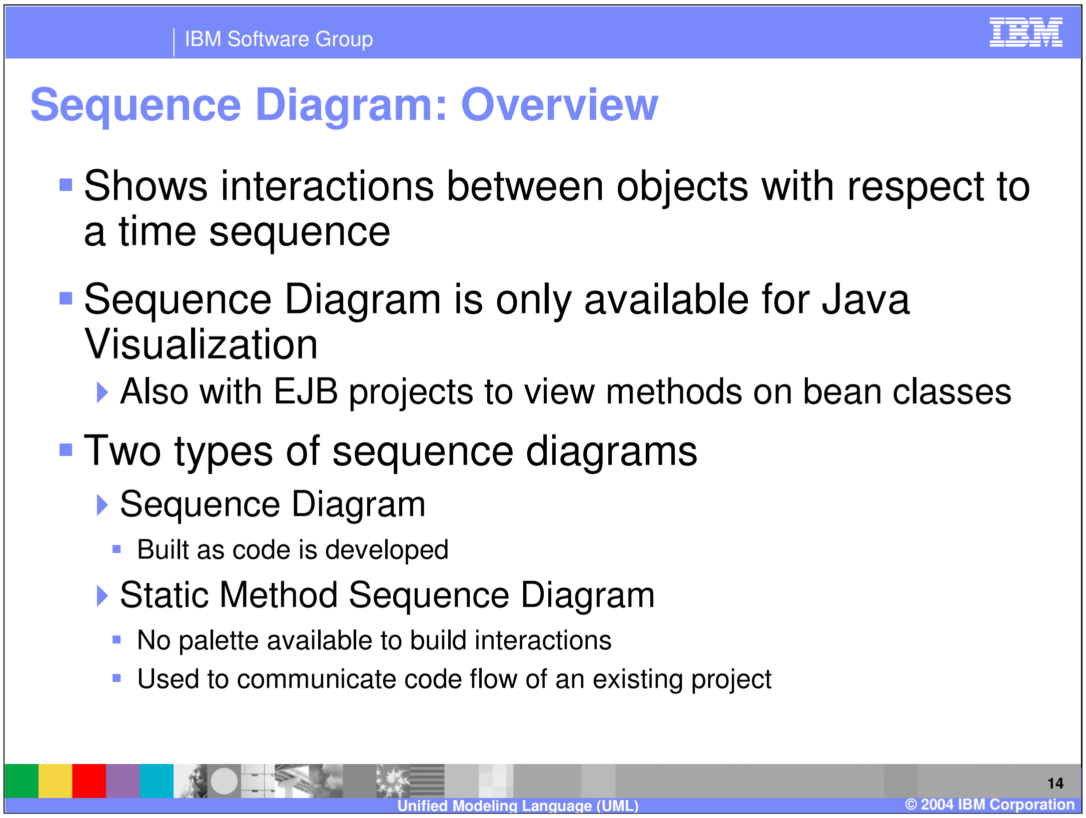
```
java.lang.Object
  └ com.ibm.demo.uml.ATM
```

public class **ATM**
extends java.lang.Object

Author:
wsdemo

Diagram:

As mentioned at the beginning of this presentation, Java visualization was available in prior releases of Application Developer. There have been some minor enhancements in this release, however. First there is now a refactoring option available form the UML diagram. Also, there is a new feature that allows a UML class diagram to be added to the Javadoc for a particular Java class. There are two approaches to including this image into the javadoc file. The first approach you can use is to embed this image into the javadoc file is to incorporate the @viz.diagrm tag in your java documentation. The second option is to select Project > Generate Javadoc with Diagrams > Automatically from the menu.



IBM Software Group

Sequence Diagram: Overview

- Shows interactions between objects with respect to a time sequence
- Sequence Diagram is only available for Java Visualization
  - Also with EJB projects to view methods on bean classes
- Two types of sequence diagrams
  - Sequence Diagram
    - Built as code is developed
  - Static Method Sequence Diagram
    - No palette available to build interactions
    - Used to communicate code flow of an existing project

14

Unified Modeling Language (UML)

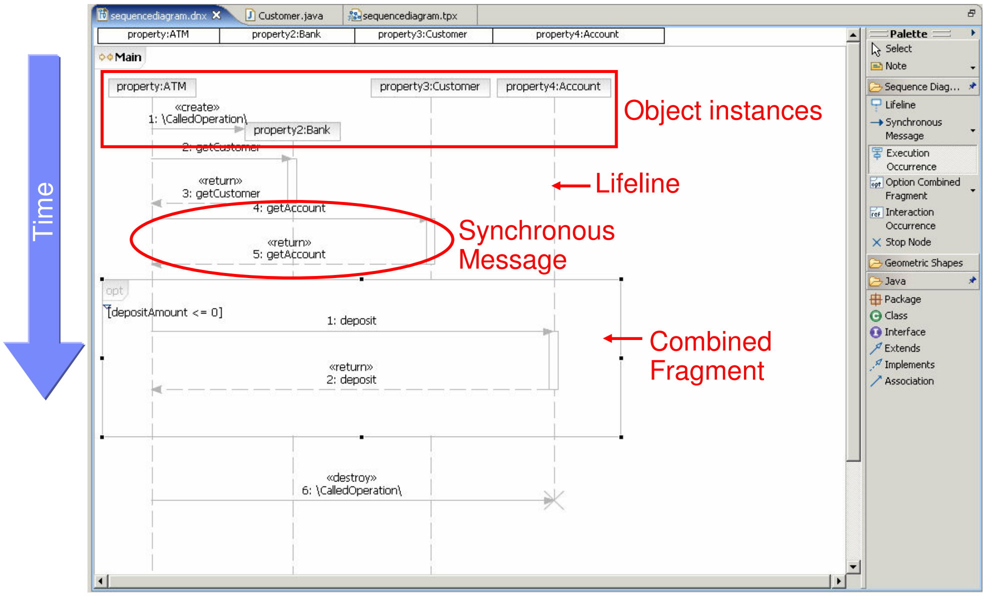
© 2004 IBM Corporation

A significant new addition to Java visualization is the support for sequence diagrams. Sequence diagrams are used to show interactions between objects with respect to time. You will see an example of a sequence diagram on the next slide.

Application Developer has two concepts of a sequence diagram, and the one you choose will depend on what you are doing. If you create a new sequence diagram, you can add objects and interactions to the diagram, and as you do the underlying code will be updated to reflect this level of visual editing. However there is also another type of diagram that is not editable, and as such does not include a palette to build interactions between objects. This type of diagram is referred to as a Static Method Sequence Diagram and is used to communicate code flow of an existing project.

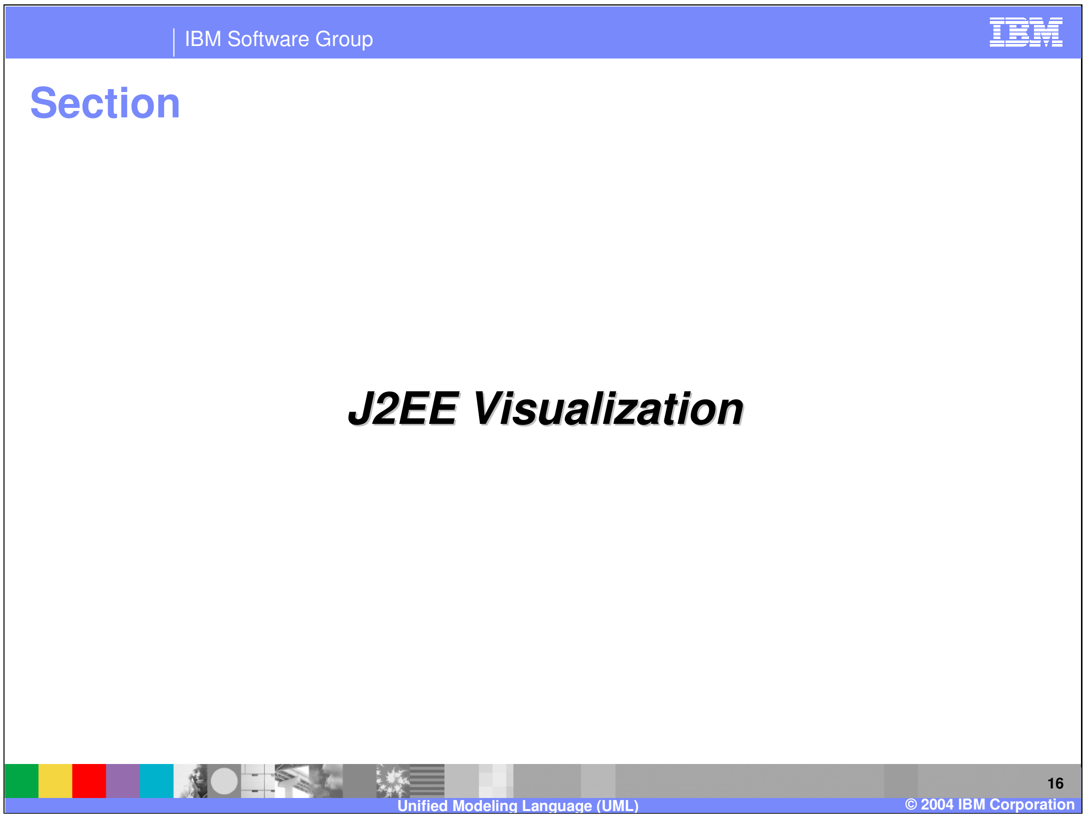
Note: state the difference between the Sequence diagrams in profiling and those in this UML

Sequence Diagram: Overview (Continued)



This slide provides an example of a sequence diagram. Notice that at the top of the diagram there is a list of object instances that run from left to right across the diagram. As you move down the diagram time is increasing. Each object instance includes a lifeline that indicates the life span of that particular object.

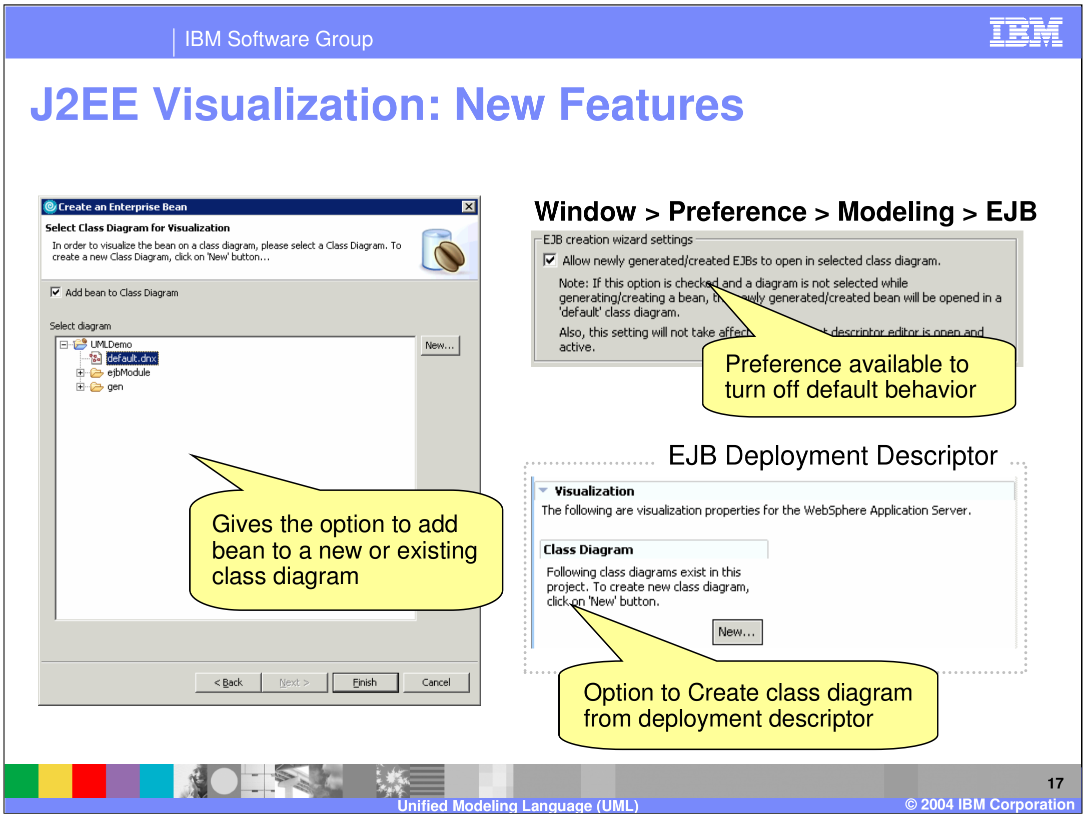
NOTE: Add a little explanation of what the code is doing.



Section

J2EE Visualization

The next section will provide an overview of the J2EE UML visualization features available with Application Developer V6.



IBM Software Group

J2EE Visualization: New Features

Create an Enterprise Bean

Select Class Diagram for Visualization

In order to visualize the bean on a class diagram, please select a Class Diagram. To create a new Class Diagram, click on 'New' button...

☑ Add bean to Class Diagram

Select diagram

Gives the option to add bean to a new or existing class diagram

Window > Preference > Modeling > EJB

EJB creation wizard settings

☑ Allow newly generated/created EJBs to open in selected class diagram.

Note: If this option is checked and a diagram is not selected while generating/creating a bean, th ewly generated/created bean will be opened in a 'default' class diagram.

Also, this setting will not take affect t descriptor editor is open and active.

Preference available to turn off default behavior

EJB Deployment Descriptor

Visualization

The following are visualization properties for the WebSphere Application Server.

Class Diagram

Following class diagrams exist in this project. To create new class diagram, click on 'New' button.

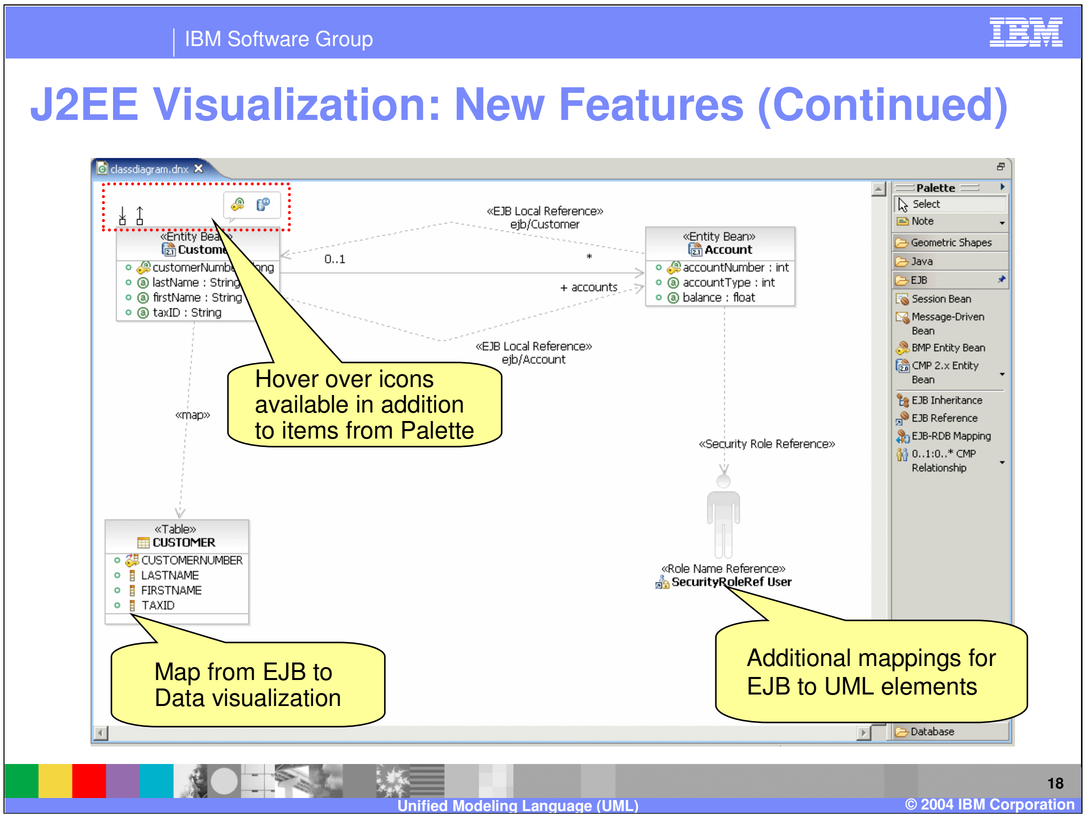
Option to Create class diagram from deployment descriptor

17

Unified Modeling Language (UML)

A new feature in the version 6 release of Application Developer is the option to add an EJB to a new or existing Class diagram from the Create an Enterprise Bean wizard. This option is available by default, however this option can be made unavailable from the preferences menu under Modeling > EJB. Additionally, the EJB Deployment Descriptor editor now includes a Visualization section on the Overview tab. In this area of the editor there is a list of existing class diagrams in this project, and a button for creating a new class diagram.

J2EE Visualization: New Features (Continued)



Hover over icons available in addition to items from Palette

Map from EJB to Data visualization

Additional mappings for EJB to UML elements

18

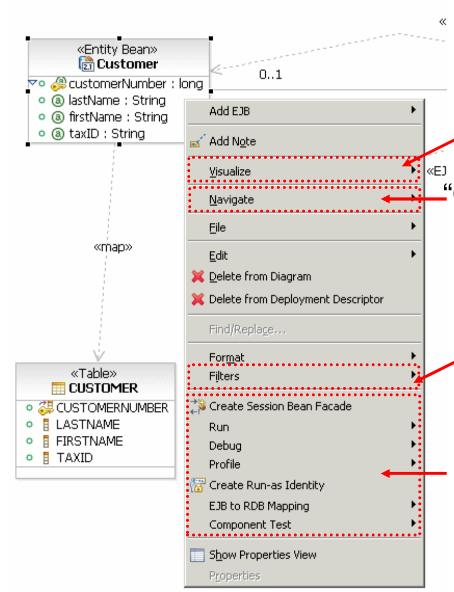
Unified Modeling Language (UML)

© 2004 IBM Corporation

There are also several new enhancements to the UML Diagram editor that are specific to J2EE application development. First there is a new set of hover over icons that appear as you positions their cursor on the edge of an artifact in the diagram. The hover over menus provide the same sort of capabilities available from the palette, however the intent here is to provide an alternative editing approach that does not require interaction with the palette. There is also a new option to map an Entity EJB to the back end data source. This is available from the EJB to RDB mapping context menu option and will create the DB element and visualize it. Finally, there are a number of new mappings available for EJB to UML elements. Some of these include, support for security roles, security role references, and exclusion lists. These mappings are listed in the product help contents.

In this release there is also the ability to use a UML component to visualize a bean (EJB) instead of class. This alternative view provides the following key differences: interfaces are grouped under provided/required compartments, methods are now grouped by interface, and interfaces can be collapsed to hide methods.

J2EE Visualization: New Features (Continued)



Visualize with topic or browse diagram

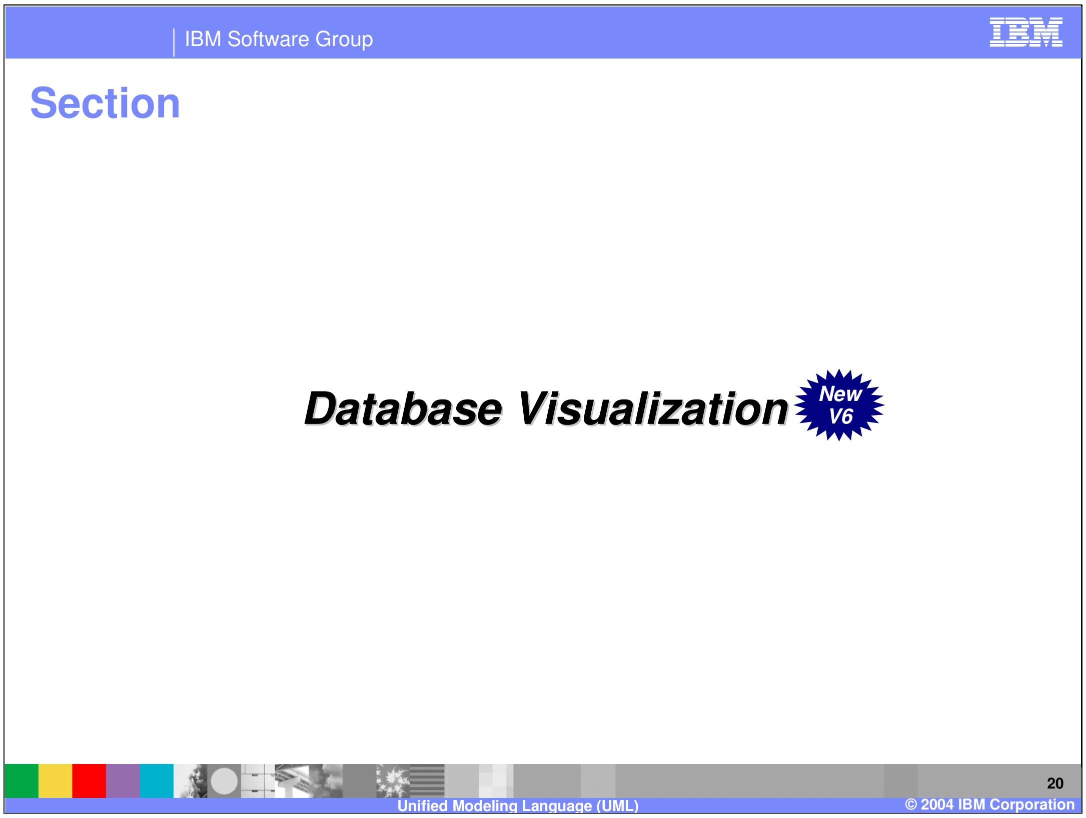
“Open with” and “Show in” options

Filter attributes and relationships shown in diagram

Menu items available from Project Navigator are available from UML Diagram

There have been several additions to the context menu of J2EE artifacts accessible from a class diagram. If you right click on an EJB from a class diagram you will notice that many menu items that are available from the Project Explorer view are also available from the UML Diagram. For example, Create Session Bean Facade is available from this context menu and choosing this option will initiate the create session bean façade wizard and allow you to visualize the new session bean façade from the current diagram.

Also available from the context menu is the option to filter some of the attributes and relationships shown in the diagram. This capability is available using the Filter option. This option can help clean up a diagram and focus on the significant relationships in the diagram. The Navigate option enables you to open the artifact with an editor of your choosing. Finally, the visualize options allow you to open this artifact in a browse or topic diagram. Browse and topic diagrams will be discussed later in this presentation.



Section

Database Visualization

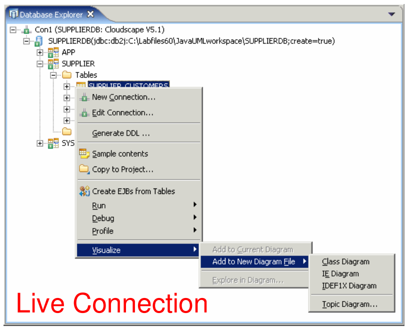
The next section will provide an overview of the Data UML visualization features available with Application Developer V6.

Database Visualization: Overview

- Types of diagrams
  - Class Diagram
  - IE
  - IDEF1X

Targeted for database designers familiar with these notations

- Can create diagram from
  - Database objects
  - Live database connection

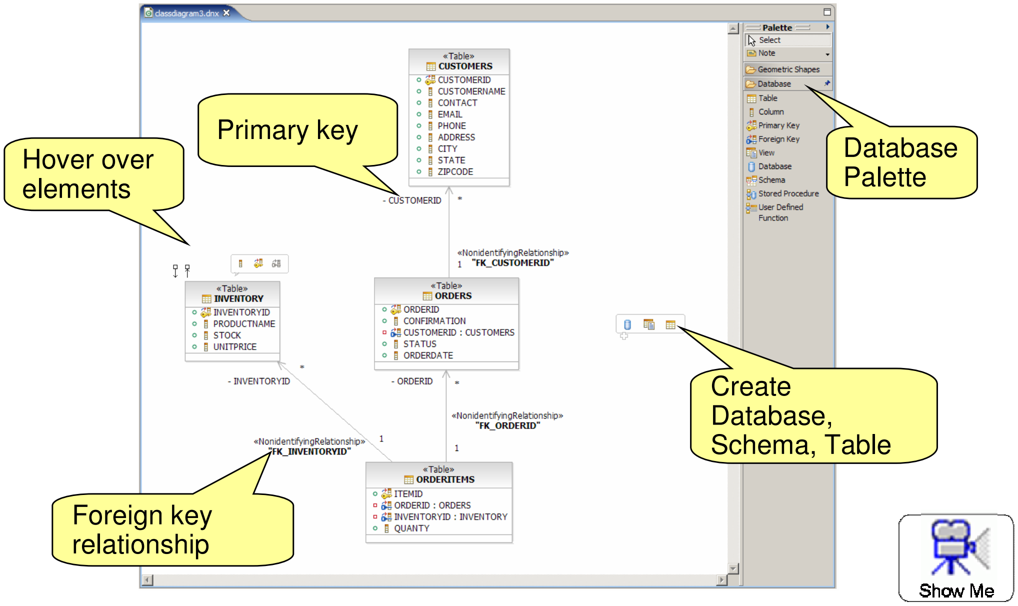


Live Connection

Data visualization is new in version 6 of Application Developer. There are several types of diagrams that can be associated with the data being visualized in the relational database. If you are familiar with UML visualization already, you are probably already familiar with the Class diagram. Application Developer also provides two other diagram types that differ only in the notation that is used to diagram the database schema.

Data visualization is very flexible because diagrams can be created from one or more database objects or from a live database connection.

Database Visualization: Class Diagram Example



Hover over elements

Primary key

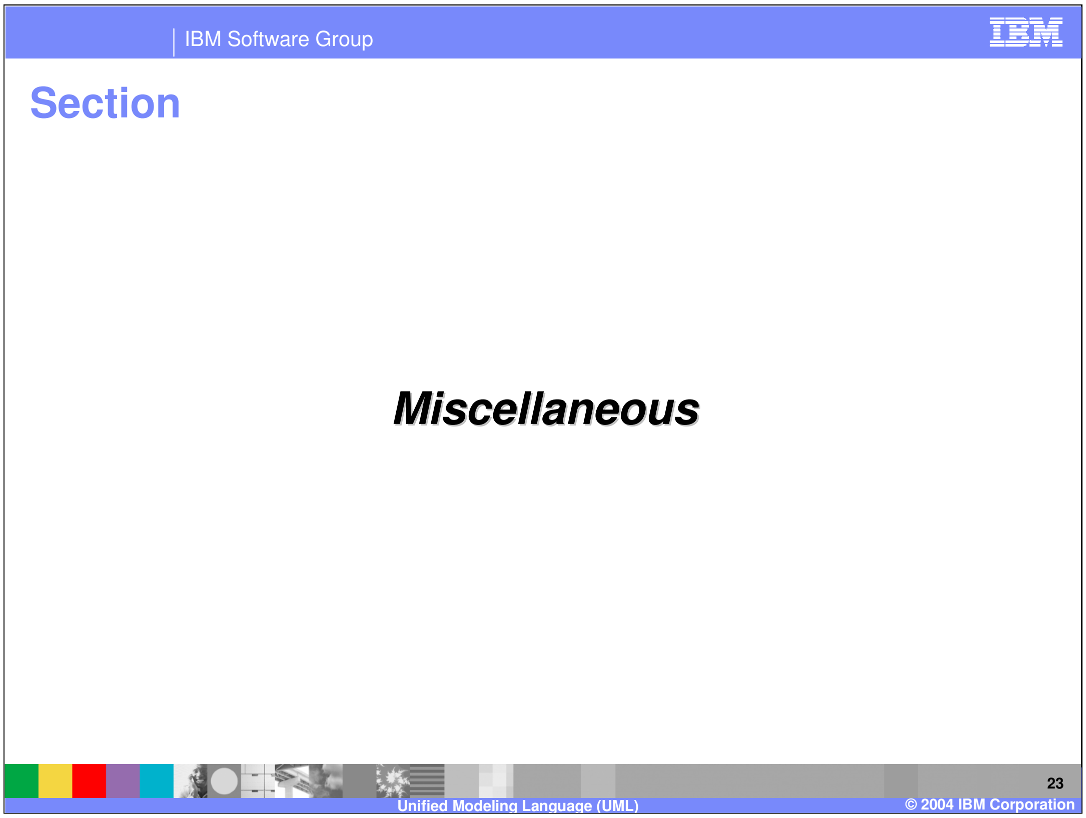
Database Palette

Create Database, Schema, Table

Foreign key relationship

Show Me

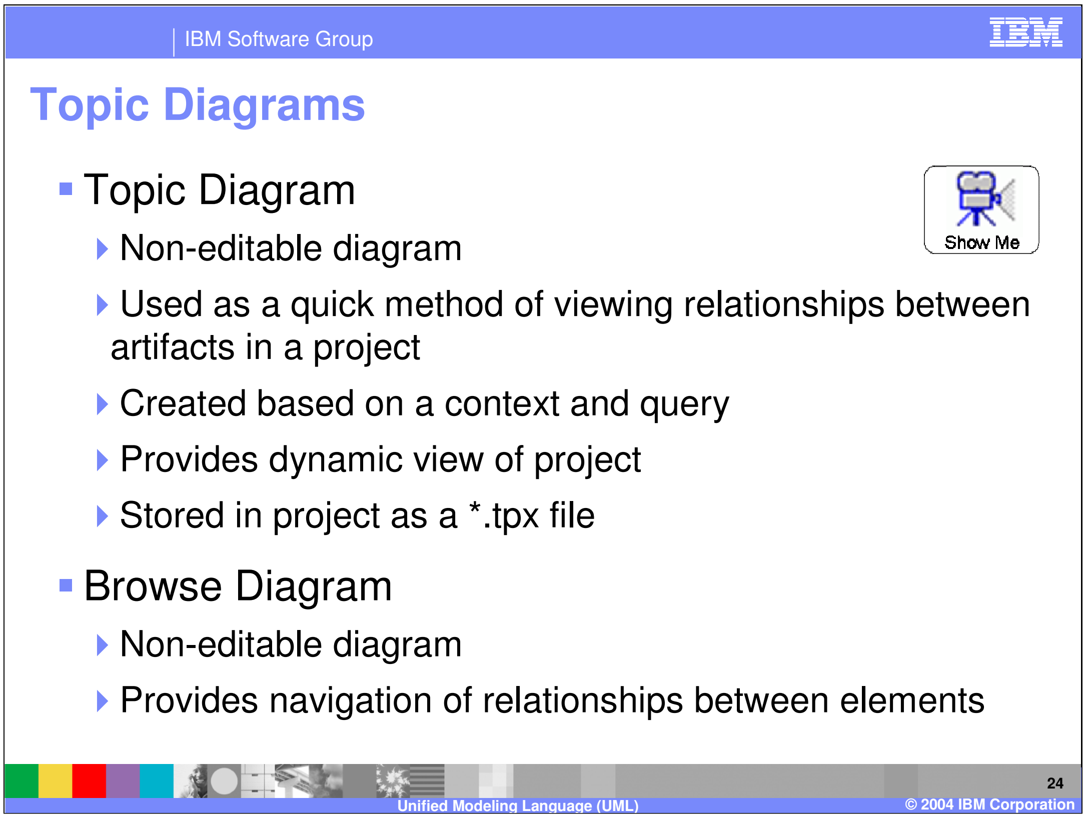
This slide shows an example of a class diagram showing the data visualization capabilities. As mentioned earlier, with the J2EE based data visualization there is the ability to do much of your activities from the hover over menu items rather than using the palette exclusively to do your visual editing.



Section

Miscellaneous

The next section will provide an overview of Topic and Browse diagrams.



IBM Software Group

Topic Diagrams

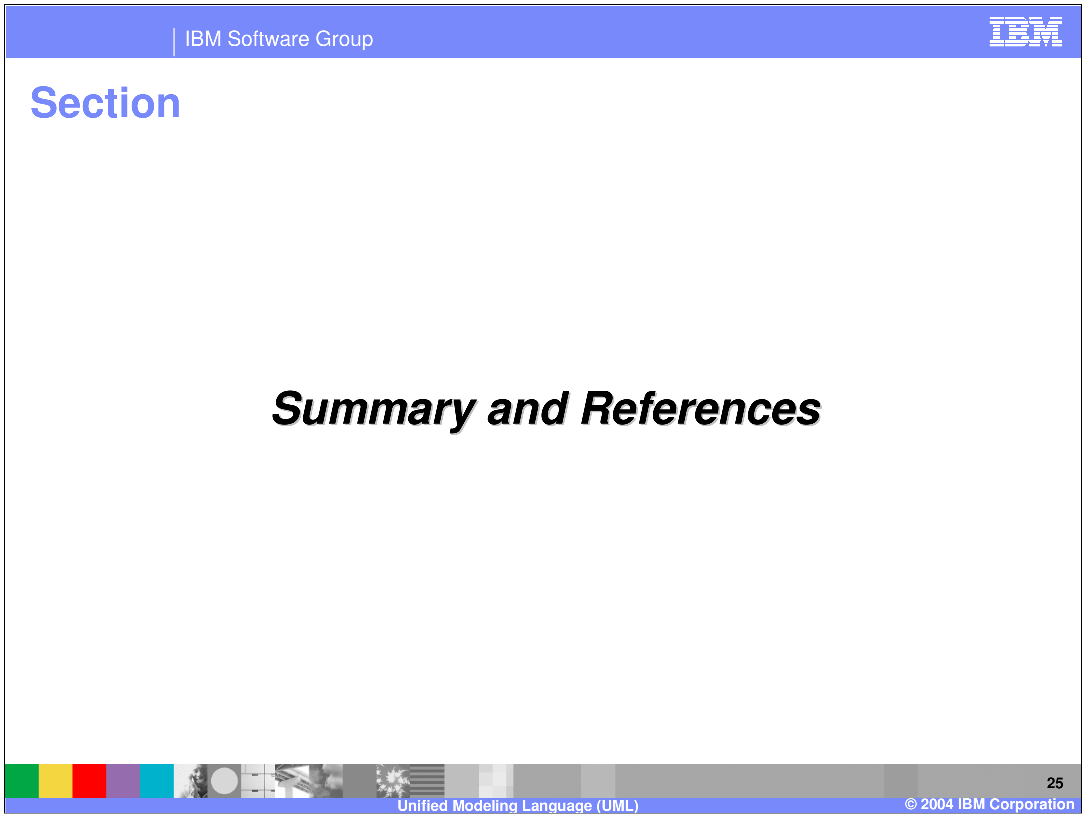
- Topic Diagram
  - Non-editable diagram
  - Used as a quick method of viewing relationships between artifacts in a project
  - Created based on a context and query
  - Provides dynamic view of project
  - Stored in project as a *.tpx file
- Browse Diagram
  - Non-editable diagram
  - Provides navigation of relationships between elements

Unified Modeling Language (UML)

24

A topic diagram is a non-editable diagram that is used as a quick method of viewing relationships between artifacts in a project. These diagrams are based upon a context and a query and as the underlying data changes the topic diagram changes to reflect the new state of the underlying code. In this way, the topic diagram provides a dynamic view of the project.

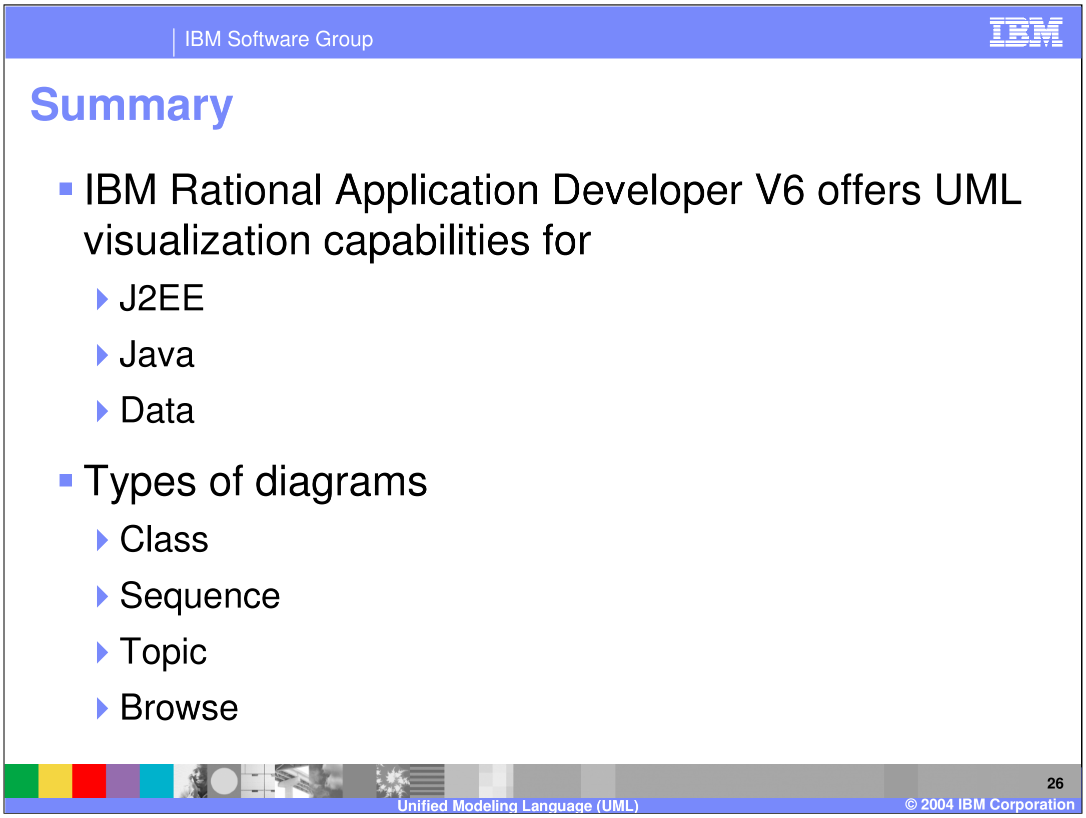
A browse diagram is like a topic diagram in that this type of diagram is also non-editable. However, the browse diagram is different from the topic diagram in that it allows you to navigate relationships between elements in a project.



Section

Summary and References

The next section will provide a summary of this presentation.



IBM Software Group

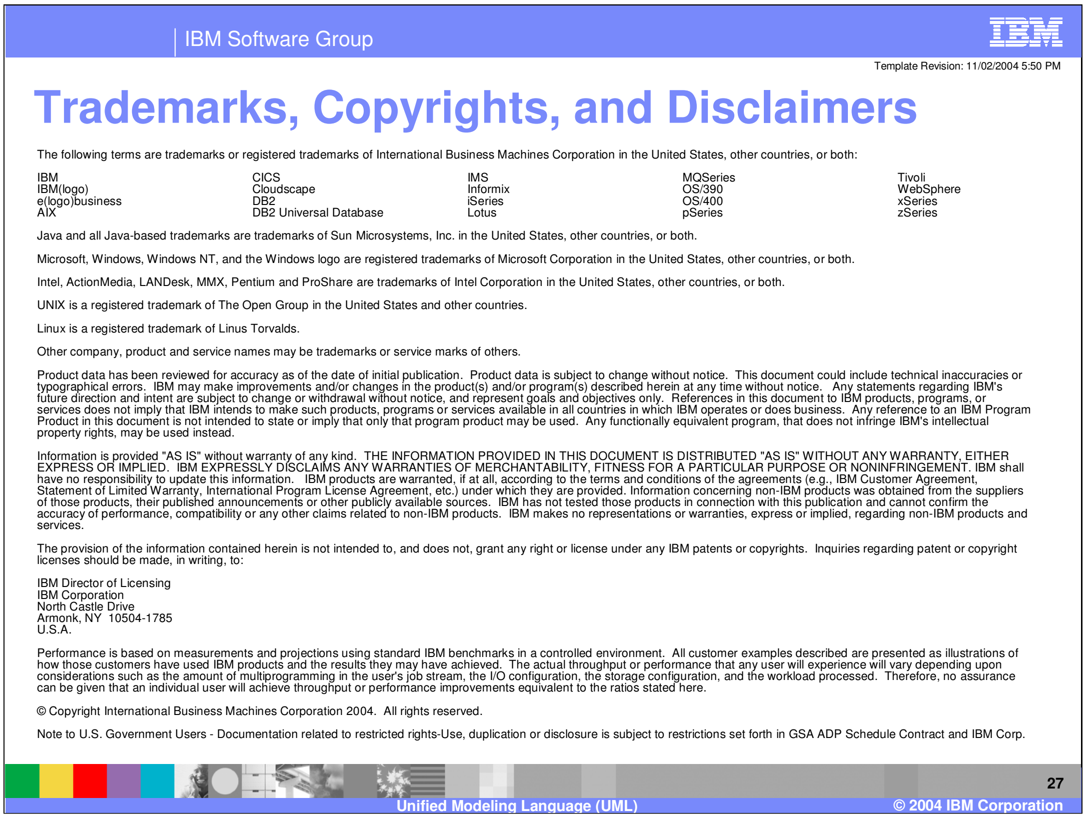
Summary

- IBM Rational Application Developer V6 offers UML visualization capabilities for
  - J2EE
  - Java
  - Data
- Types of diagrams
  - Class
  - Sequence
  - Topic
  - Browse

26

Unified Modeling Language (UML)

In summary, this presentation has focused on identifying several of the new features available in Application Developer V6. Particularly, there is now support for Data visualization and also a range of new visualization capabilities such as Sequence, topic, and browse diagrams.



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.