Getting the Most from Eclipse

Darin Swanson

IBM Rational

Portland, Oregon

Darin_Swanson@us.ibm.com

March 17, 2005

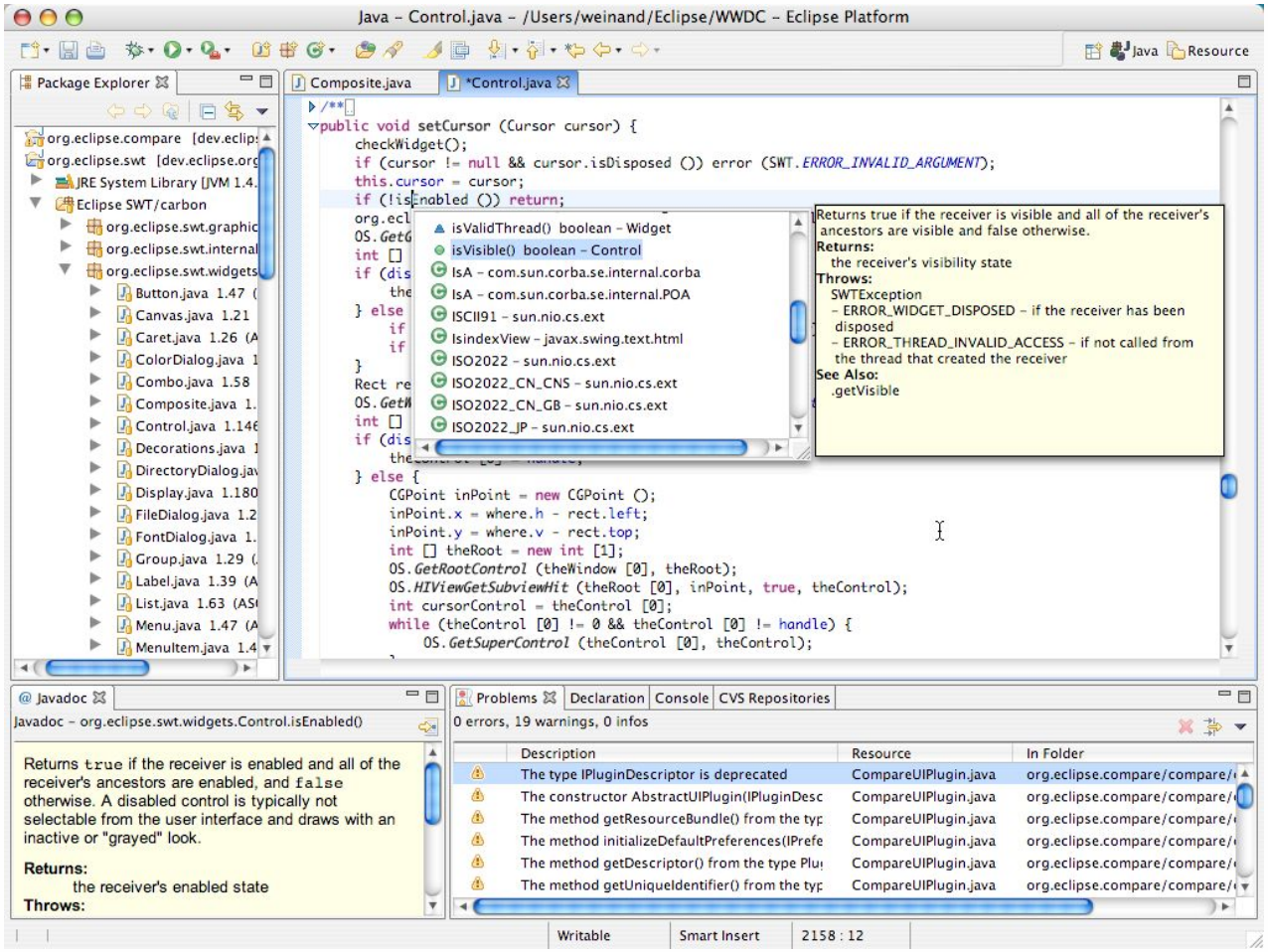
What is Eclipse

- An extensible tools platform
- Out-of-box function and quality to attract developers
  - a development environment for itself (Java IDE)
- An open source community to develop the technology
- An ecosystem to enable a total solution
  - Including Products by some major tool vendors
- A Foundation to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services.

Outline

- Overview of the IDE
  - Highlights of the Java IDE
- Dig Deeper: Architecture of Eclipse
  - Built for more than just Java
- Review of PDE
  - Basic functionality
  - More advanced

Java Development Tools



JDT Features: Build and Debug

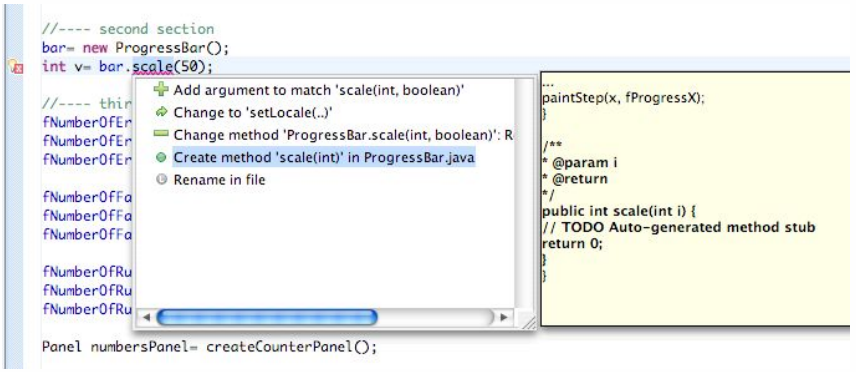
- Incremental automatic building in the background
- Debugging
  - Hot code replace / drop to frame
  - Expression evaluation
  - Run code with errors
  - Logical structures
  - Monitors and locks

JDT Features: Exploration at your fingertips

- Code folding
- Quick (in-place)
  - outlines
  - type hierarchies
  - diff
  - inspectors, evaluation results
- Linked views
  - declaration
  - Javadoc
- On the fly marking of…
  - references
  - exceptions
  - method exits
- Semantic coloring
- Override indicators

JDT Features:
Refactoring → continuous improvement

- More refactorings
  - generalize type, introduce parameter, introduce factory
- Tuning of existing refactorings
  - update references in Javadoc
  - find duplicate code when extracting a method
- Enable refactoring participation



JDT Features: J2SE 5.0

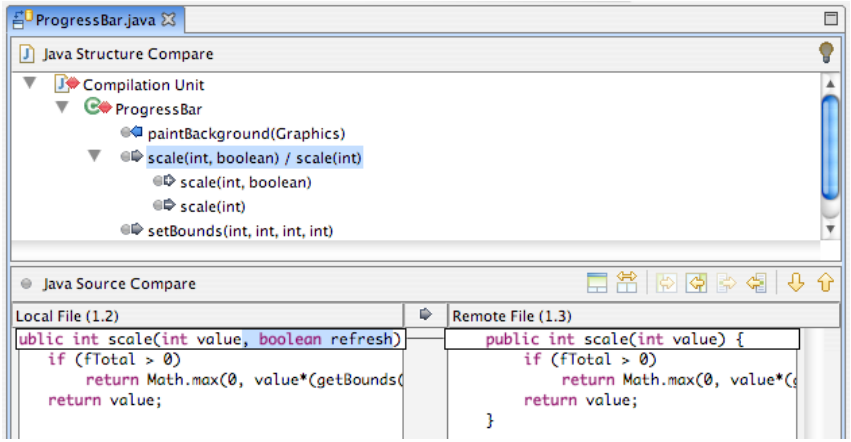
- Supported in 3.1 stream
- Significant changes to the Java language
  - generic types
  - enumerations
  - autoboxing
  - enhanced for loops
  - static imports
  - metadata facility

JDT Features: Cont'd

- Programming by intention
  - Many more quick fixes and quick assists
- More validation as you type
  - Javadoc tags
  - Unreachable code
  - Code style
  - Spelling in comments and Javadoc!
- New formatter
- Code assist everywhere
- User defined libraries
- Property file support

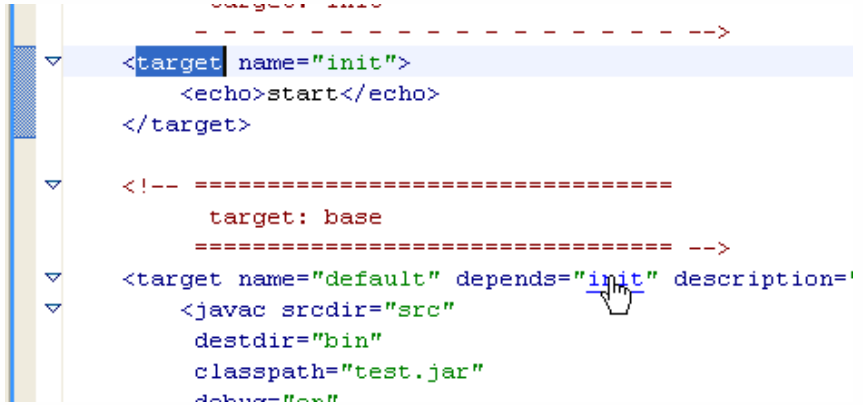
JDT Features: Cont'd

- Precise searching
  - References
  - Type hierarchies
  - All types
- Java aware structural comparisons/local history
- Open source tools integration
  - Ant
  - JUnit
- Extension APIs



Ant Integration

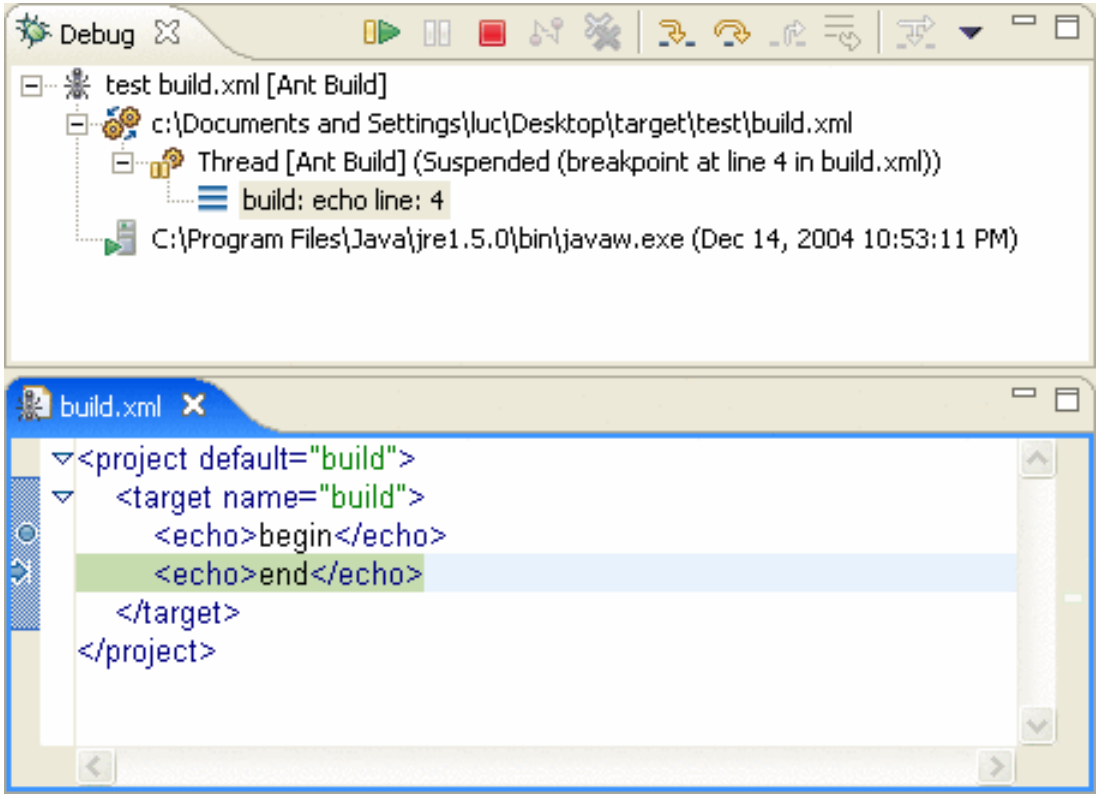
- Ant Editor
  - Content assist
  - Text hovering
  - Problem annotations
  - Navigation
  - Synchronized outline
  - Formatting
  - Open external documentation



- Ant View
  - Simplifies running a single Ant target

Ant Integration Cont'd

- Ant Debugger
  - Breakpoints
  - Stepping
  - Termination
  - Properties shown in the Variables View
    - System, User and Runtime
  - Run to line
- Ant Builders



JDT Demo

Digging into Eclipse

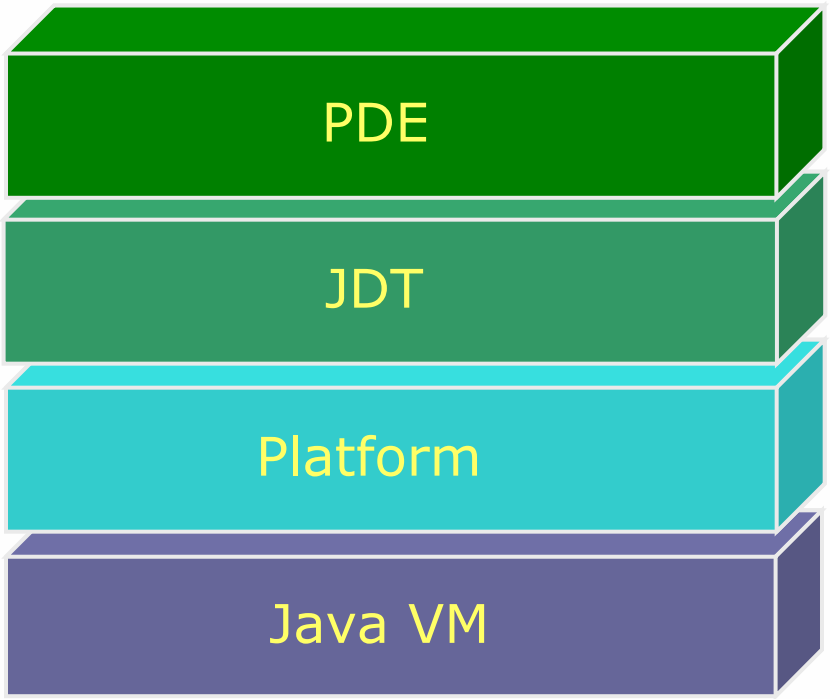
- Eclipse is a universal platform for integrating development tools
- Open, extensible architecture based on plug-ins

Plug-in development environment

Java development tools

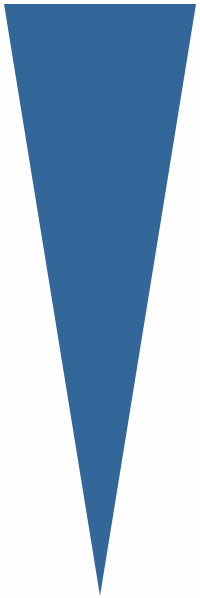
Eclipse Platform

Standard Java2 Virtual Machine



Moving forward

- eclipse perception changes
  - 3.0 “eclipse is a general application platform”
  -
  - 2.0 “eclipse is a general tooling platform”
  -
  - 1.0 “eclipse is a Java IDE”



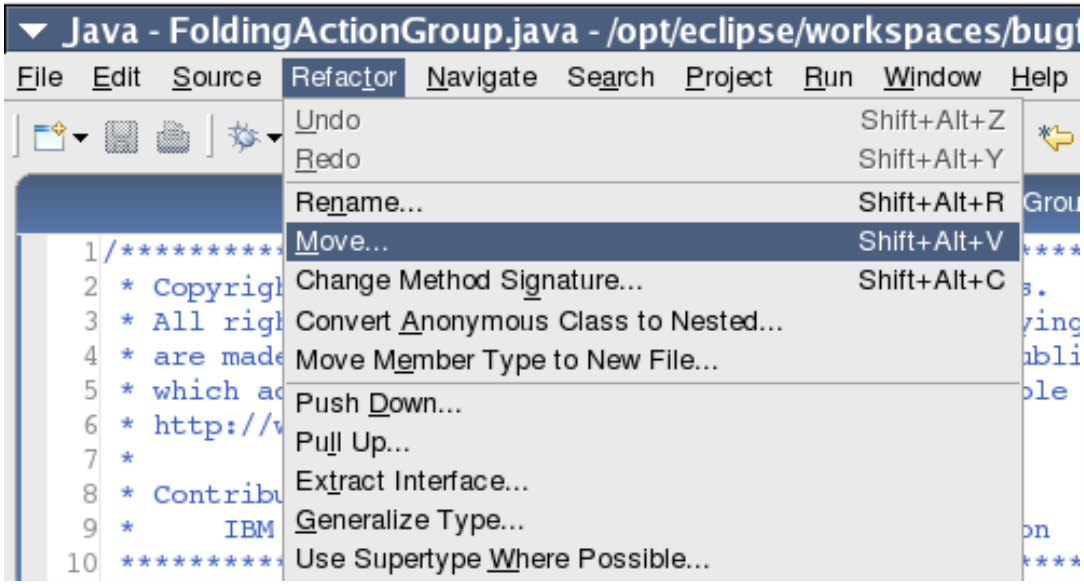
Towards a Rich Client Platform

- **Many workbench components are not IDE specific. Advanced desktop applications have similar needs**
  - open architecture
  - efficient, configurable, portable user interface
  - supports product branding, install/update support
  - integrated help, user configuration/preferences

- **Enable workbench to be used for non IDE applications**
  - remove IDE personality from workbench
    - no built-in editors, views, perspectives
  - remove assumption that workspace is the data model
  - make most other components optional
    - rich function, low footprint

What is a Rich Client Platform

- An application that uses the windowing and GUI features of the operation system they run on. This means:
  - Native widgets, menu and tool bars
  - Drag & Drop
  - Integrates with platform component model



Eclipse Rich Client Platform

A Rich Client Platform needs a strong component model with the following major characteristics:

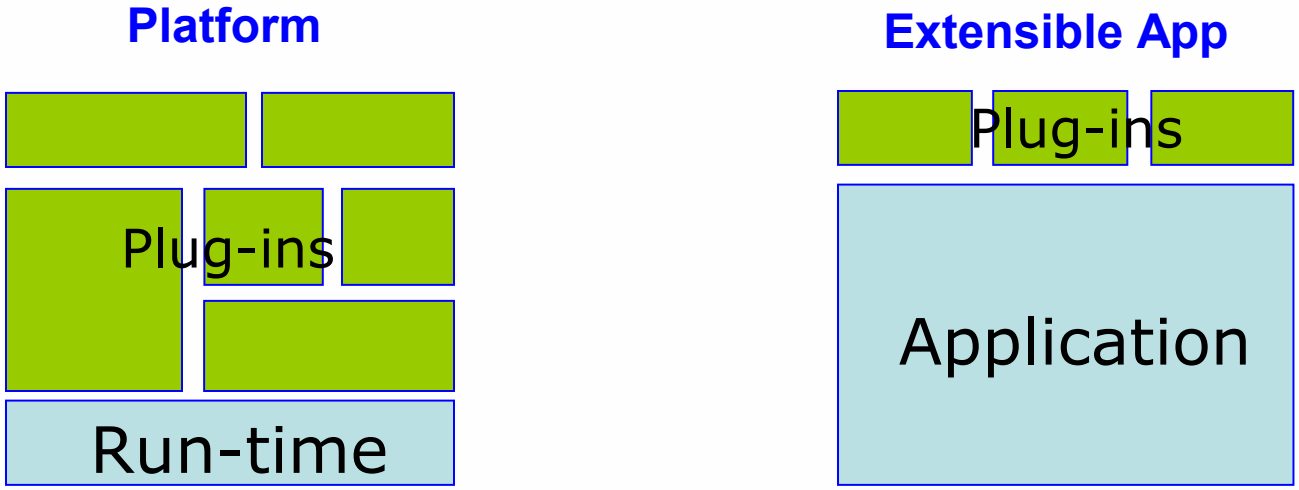
- Specified interfaces: a component must declare its public API and how it can be extended
- Lazy loading: components are loaded on demand not on startup
- Versioning: prerequisite components are reference by name and version
- Dynamic detection: components are detected dynamically (no need to restart)

Additionally the following issue must be addressed:

- Managing: install, update, remove & discover components
- Development: IDE to develop components
- Security: based on Java 2 security

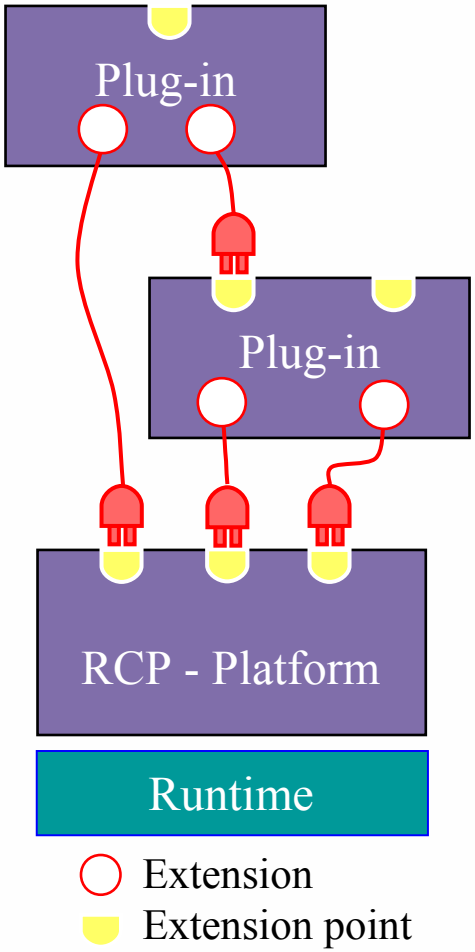
Platform vs. Extensible Application

- Eclipse Rich Client Platform
  - It has an open, extensible architecture
  - Built out of layers of plug-ins



Eclipse plug-in Architecture

- **Plug-in == Component**
  - Set of contributions
  - Smallest unit of Eclipse function
  - Details spelled out in plug-in manifest
  - Big example: mail client
  - Small example: action to calculate the number of lines of a mail
- **Extension point** – named entity for collecting contributions
  - Example: extension point to add additional spam filtering tools
- **Extension** – a contribution
  - Example: a specific spam filter tool
- **RCP - Platform** – set of standard plug-ins
- **Runtime** – controls and manages contributions



Plug-in Manifest

```
<plugin
   id= "com.example.tool.mail"
   name= "Example Mail Plug-in"
   version= "1.0.0"
   class = "com.example.tool.MailPlugin">
  <requires>
   <import plugin= "org.eclipse.ui" version="3.0.0"/>
  </requires>
  <runtime>
   <library name = "mail.jar">
    <export name= "org.example.tool.mail"/>
   </library>
  </runtime>
  <extension point = "org.eclipse.ui.preferencepages">
    <page id = "com.example.tool.mail.preferences"
      title = "Mailing"
      class = "com.example.tool.mail.PreferencePage"/>
  </extension>
  <extension-point
    name= "spamFilters"
    id = "com.example.tool.mail.spamFilters"/>
</plugin>
```

Plug-in identification

Required Plug-ins

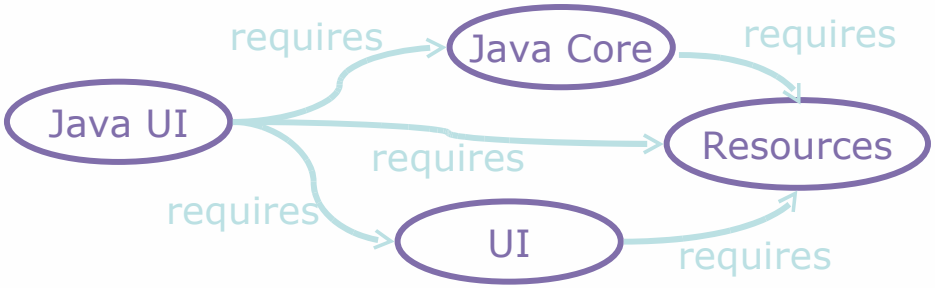
Plug-in code

Declare contribution this plug-in makes

Define new extension point open for contributions

Eclipse Runtime

- Java component (*plug-in*) model
  - dependency management
  - activation management
- Extension registry - manages
  - extension points and
  - corresponding extensions
- OSGI based (Open Service Gateway Initiative):
  - Nokia, NTT, Motorola, Philips, Siemens, Oracle
  - dynamic install/uninstall/update of components
  - service architecture
  - security (based on Java 2)
  - remote configuration API

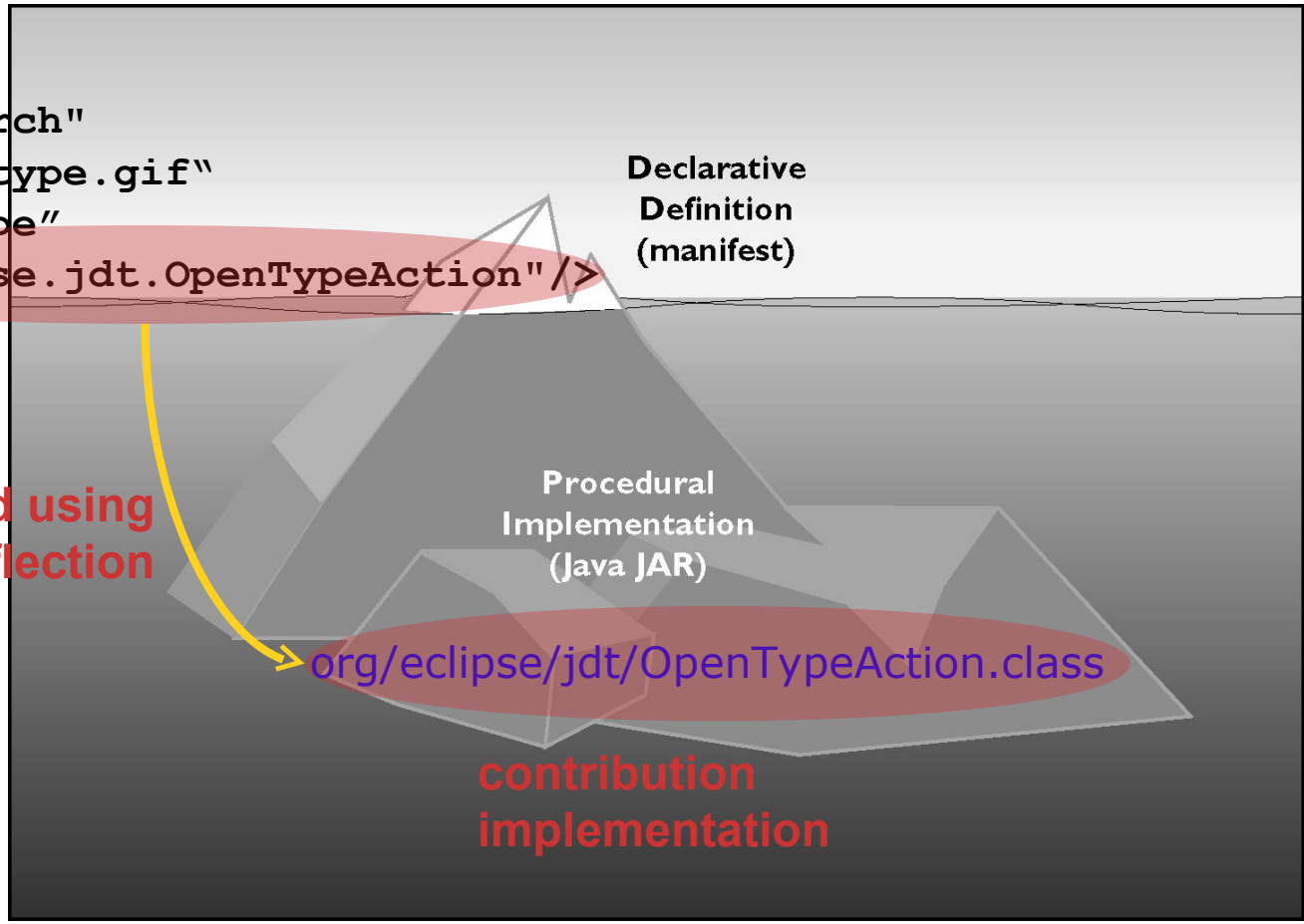


Lazy Loading

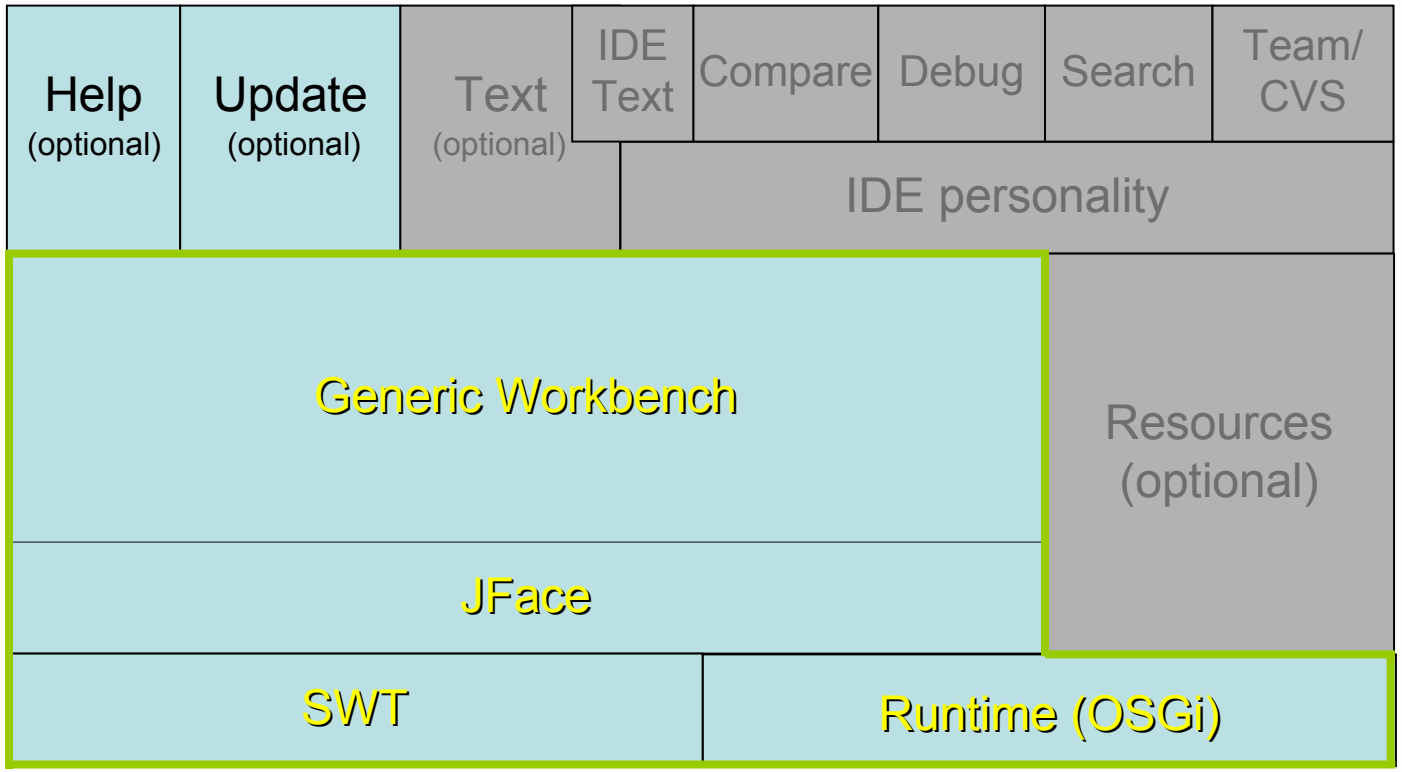
user visible appearance

```
<action
    toolbarPath="search"
    icon="icons/opentype.gif"
    toolTip="Open Type"
    class="org.eclipse.jdt.OpenTypeAction"/>
```

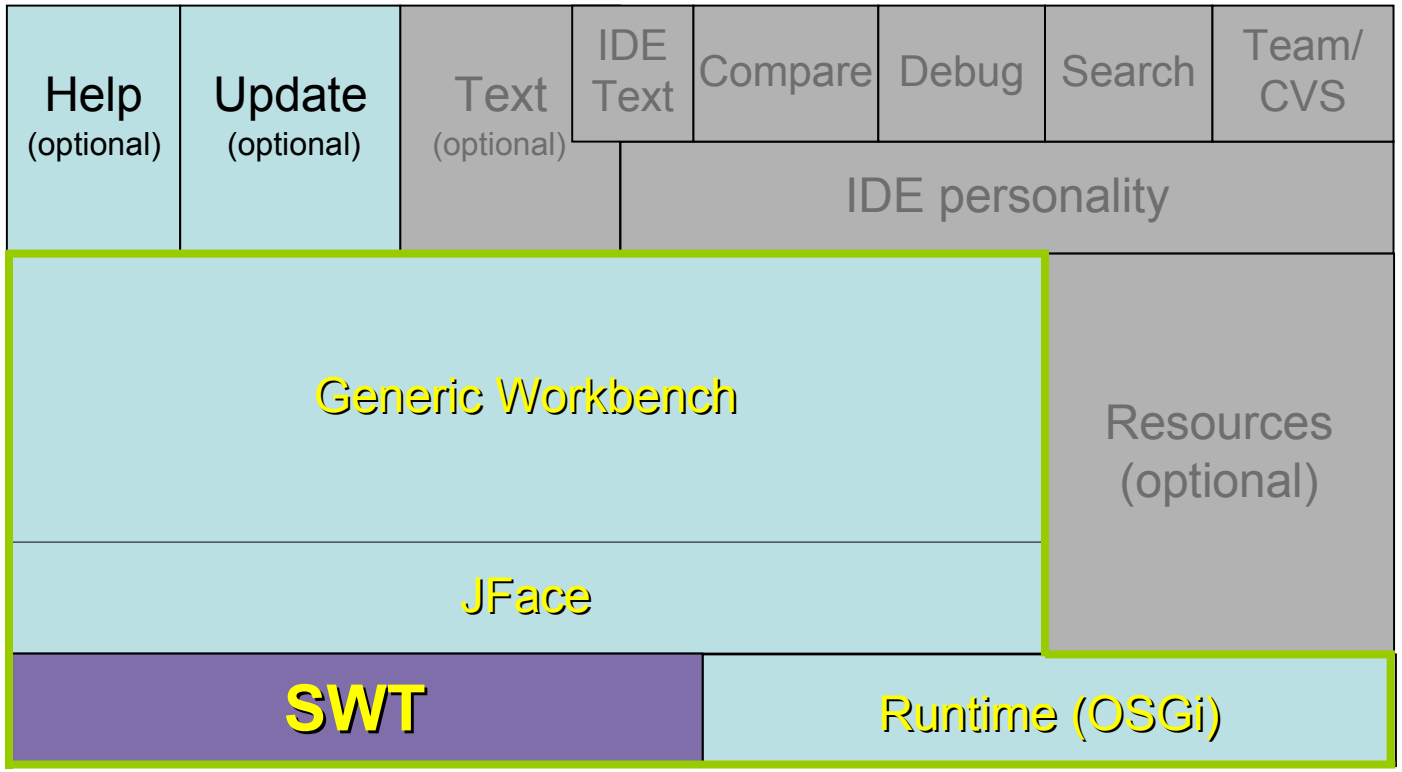
lazily instantiated using reflection



Eclipse Platform



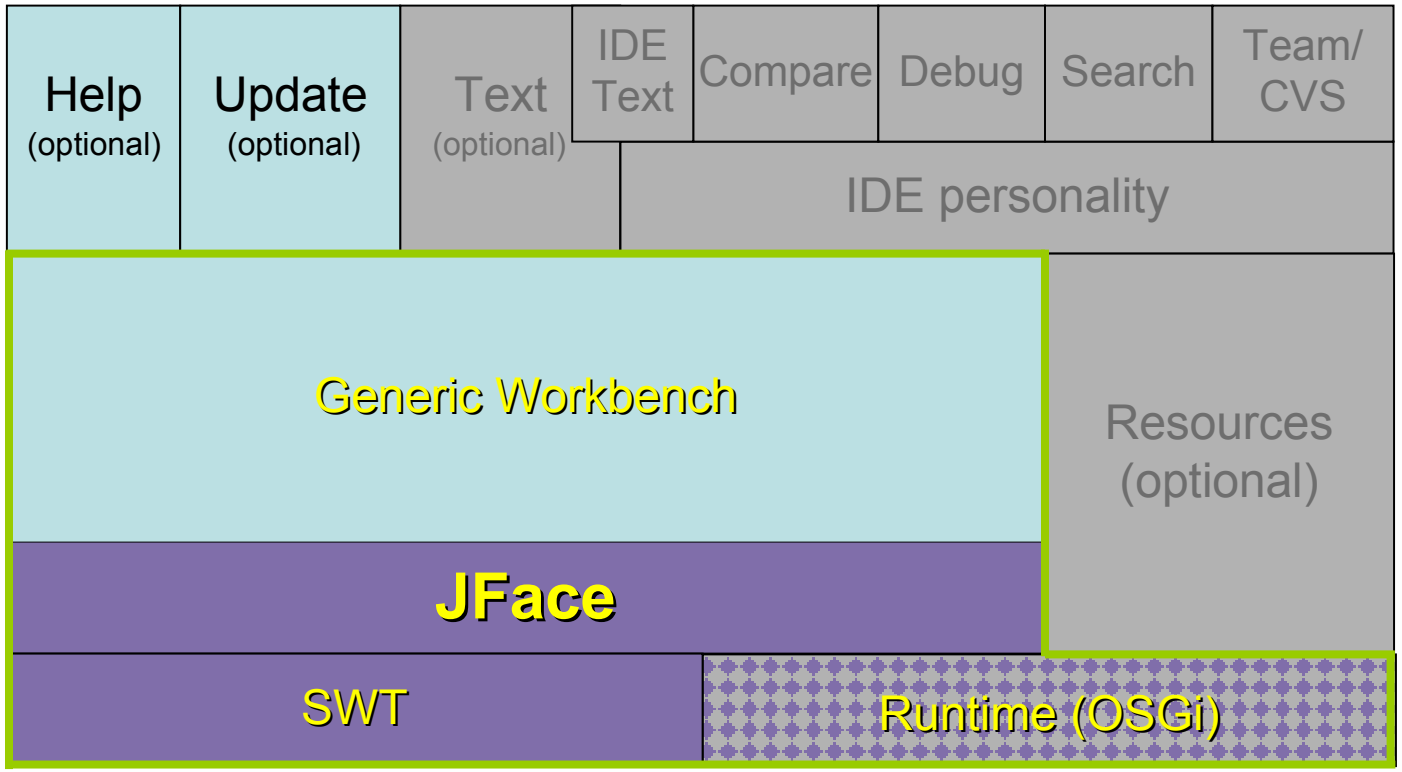
Standalone Applications



Standalone Applications

- Application model
  - Single not extensible application
- **S**tandard **W**idget **T**oolkit
  - Platform independent widget toolkit
    - Native widgets (button, tree, table, menu, …)
    - Win32, GTK, Motif and Mac
  - Integrates with other native application
  - Support for OS component model
    - OLE under Win32
- Programming model
  - OO widget library – no framework
  - API equivalent to native Win32 or GTK applications

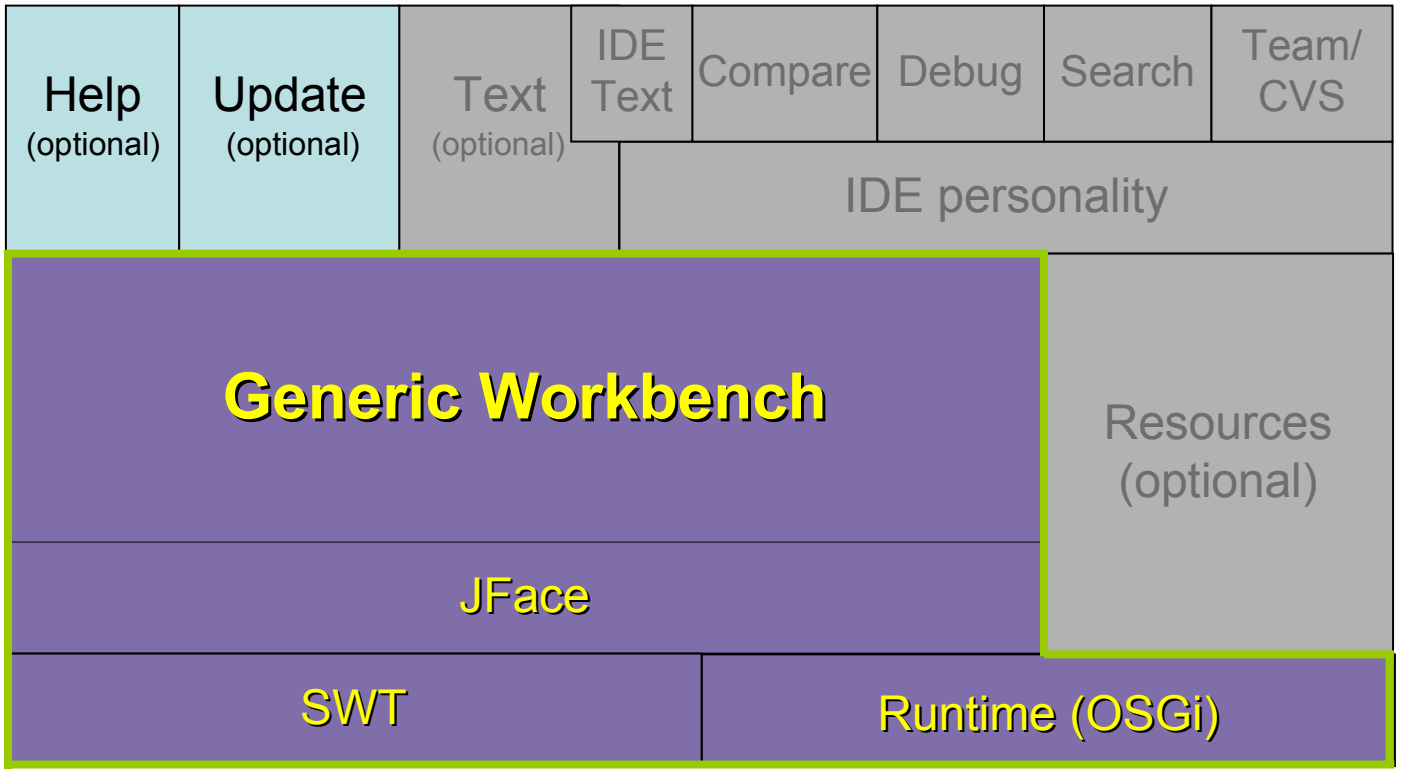
Extensible Applications



Extensible Applications

- Application model
  - Single application
- JFace – brings:
  - MVC concept: viewer & content provider
  - Application window: menu bar, tool bar, content area & status line
  - Action support: menu bar, toolbar, context menu
  - Preference and wizard framework
  - No extension points, API only
- Runtime – brings:
  - Change for extensiblilty
- Programming model
  - Formed by Model View Controller paradigm
  - „Frameworkish“

Application platform

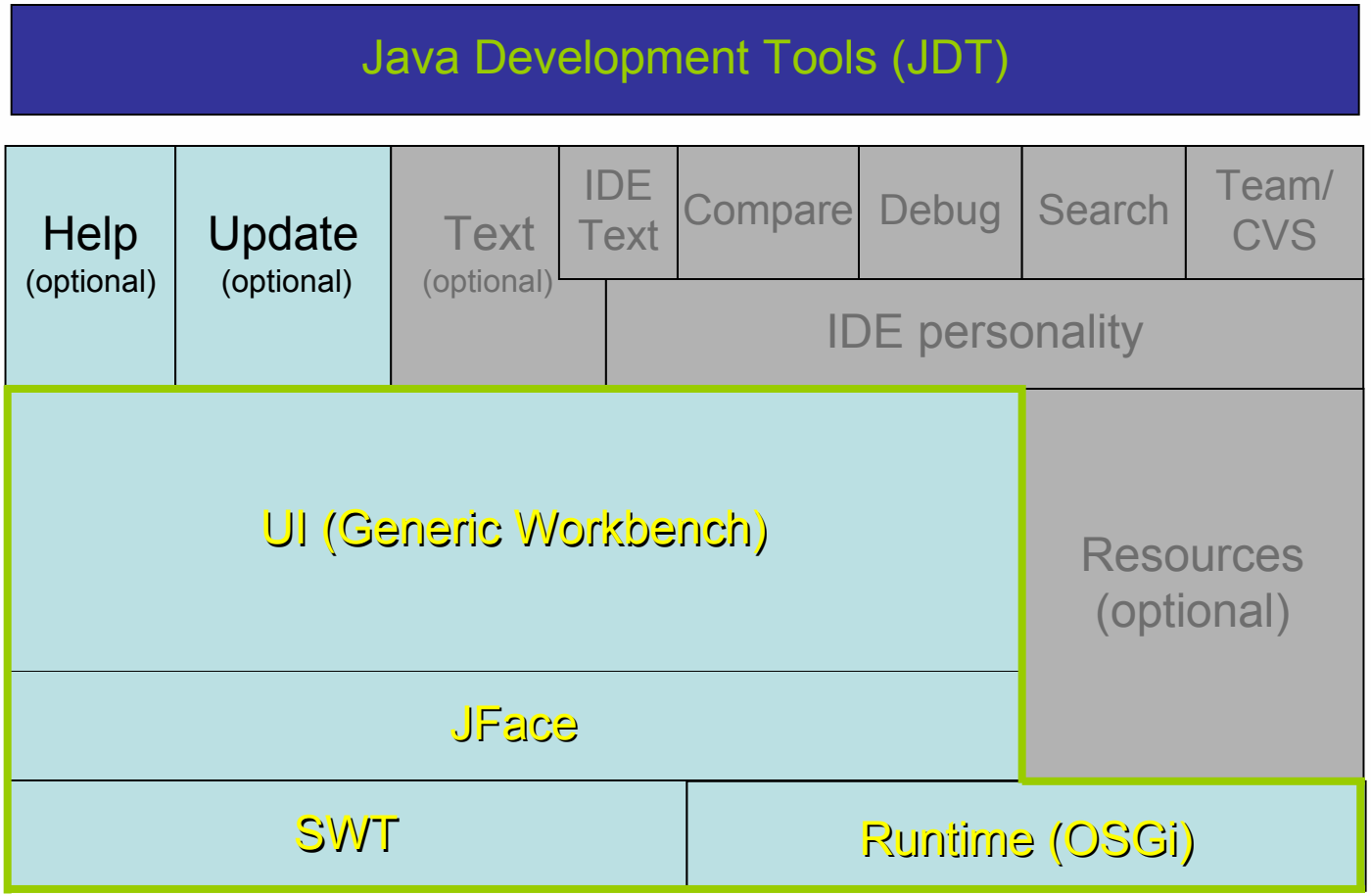


Application platform

- Application model
  - Family of components (Mailing, Organizer, Address-Book, …)
  - Different sets of components form different applications
- Workbench – brings:
  - Perspectives: define arrangement of editors and views
  - Editors: edit or browse a document or input object
  - Views: navigate a hierarchy of information
  - Manages shared resources like global menu, preference pages, …
- Programming model
  - Components contribute to workbench extension points
  - Components provide own extension points
  - Split between XML (plugin.xml) and Java code

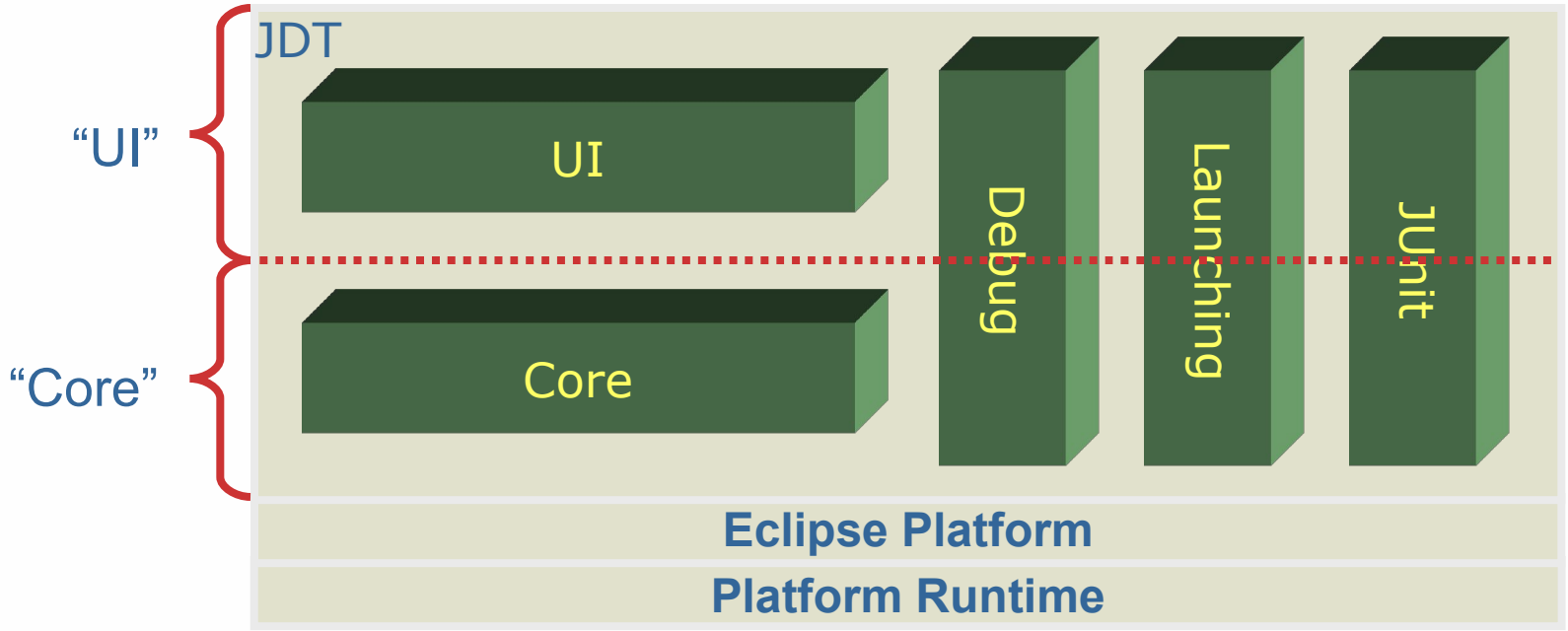
JDT is just a set of plug-Ins

Extends core Platform functionality



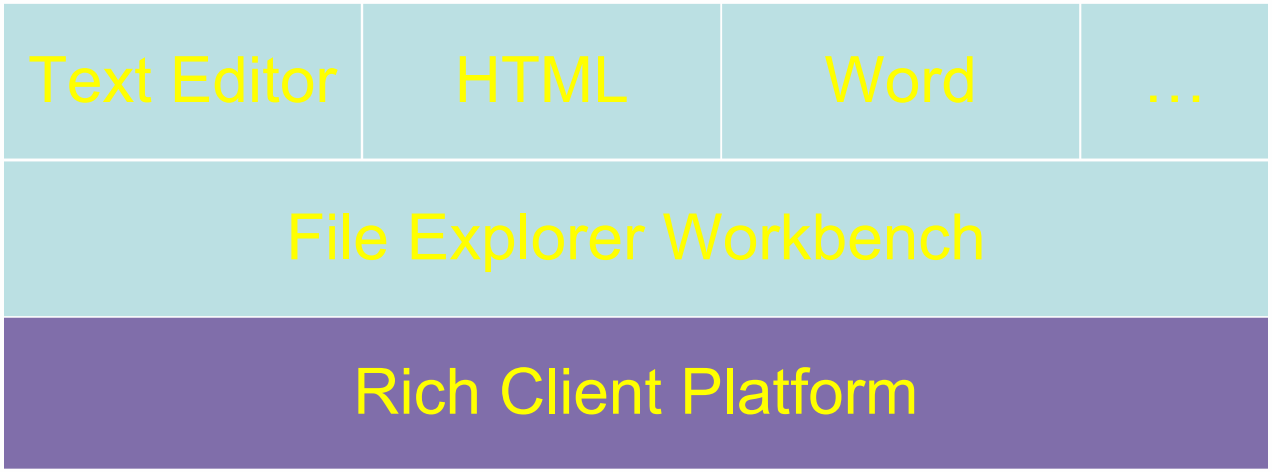
JDT Plug-ins

- JDT is the common base for Java™ tools
- Key components are:

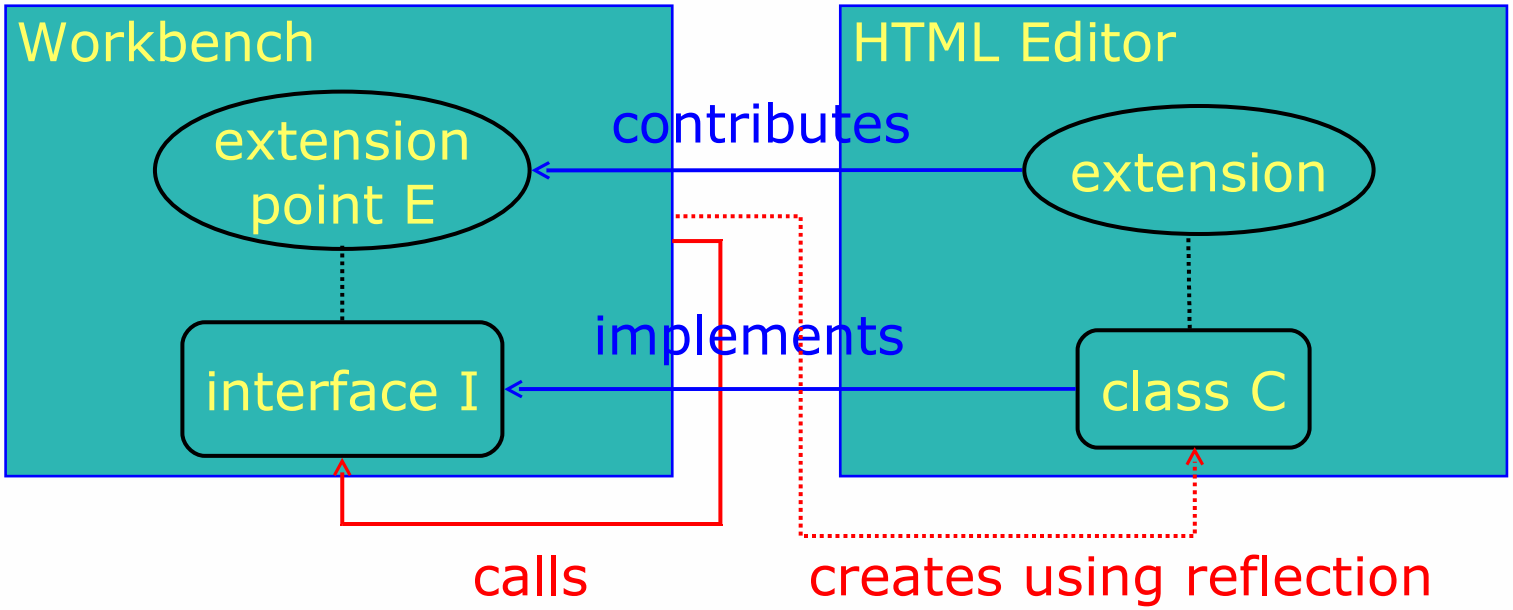


Smaller Example

- Workbench providing a file system model and views to browser the model
- No editors are provided by the workbench
- Several additional plug-ins to edit different kinds of files



Contributing to an Extension



- **Workbench**
  - Declares extension point E (org.eclipse.ui.editors)
  - Declares interface I (IEditorPart) for E
- **HTML-Editor**
  - Implements interface I with its own class C (HTMLEditor)
  - Contributes class C to extension point E
- Workbench instantiates HTMLEditor and calls its methods via interface IEditorPart

Developing Plug-ins: PDE

- PDE = Plug-in development environment
- Extenders use PDE to implement plug-ins
- Specialized tools for developing Eclipse plug-ins
- Built atop Eclipse Platform and JDT
  - Implemented as Eclipse plug-ins
  - Using Eclipse Platform and JDT APIs and extension points
- Features
  - Specialized PDE editor for plug-in manifest files
  - Templates for new plug-ins
  - PDE runs and debugs another Eclipse application

PDE Basics

- Setting up a plug-in for development
  - Plug-in project wizard
  - Templates
  - Plug-in editor
- Management of classpath
    - PDE Container
    - Launcher
    - Export wizard
- Developing a plug-in
  - “Monkey see, monkey do”
  - Search for an example within the Eclipse code base
  - Self hosting → launch the Eclipse application
    - Pause main thread
  - Testing: JUnit plug-in tests

PDE Advanced

- Extension point schema editor
- Configuring target
  - Specifying the eclipse version to target against
  - Specifying a subset of plug-ins
- Features
  - PDE provides means for feature based self-hosting
    - Feature project
    - Feature editor
- PDE and RCP
  - RCP product configuration editor
  - Build product in a single click
    - Definition to branding to executable

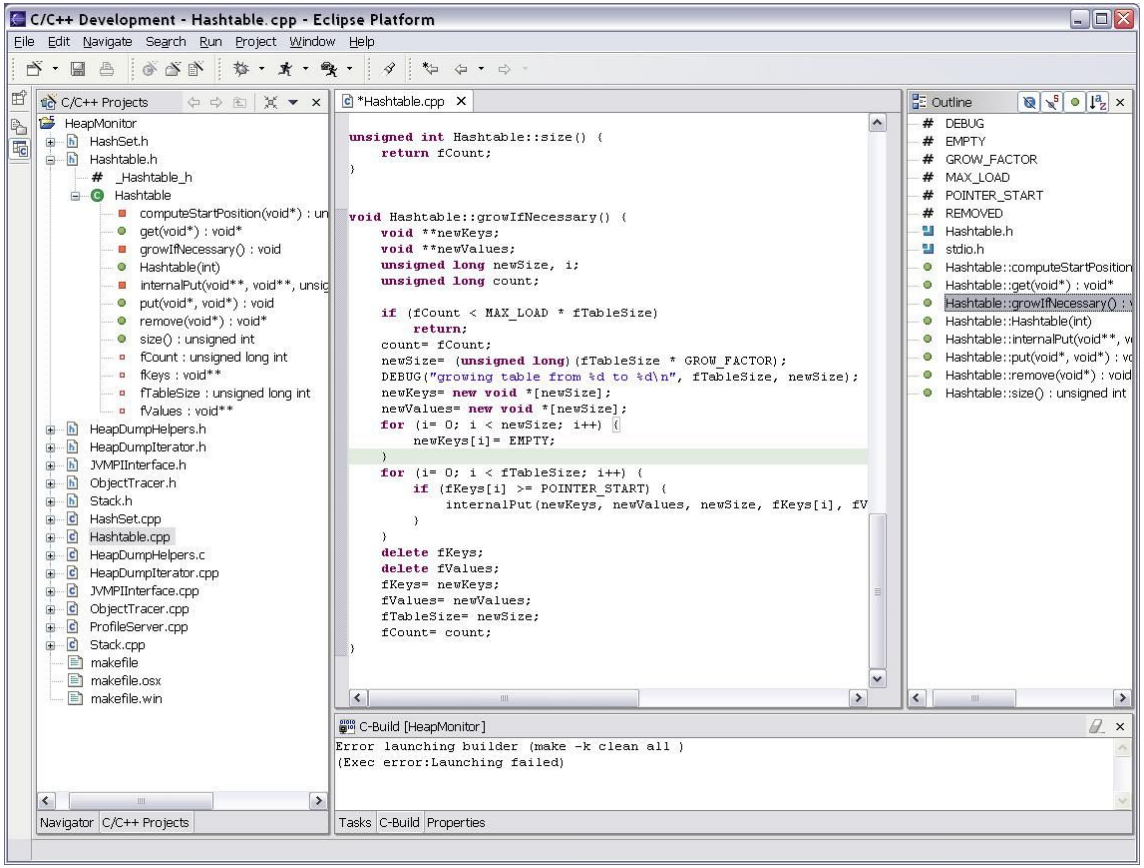
PDE DEMO

Full Eclipse spectrum

- eclipse project
  - Platform
  - JDT
  - PDE
- eclipse tools project
  - **VE**
  - UML2
  - Hyades
  - **CDT**
  - GEF
  - COBOL
  - …
- eclipse technology project
  - CME
  - ECESIS
  - WSVT
  - **AJDT**
  - AspectJ
  - Equinox
  - GMT
  - Stellation
  - XSD
  - …
- eclipse web tools project

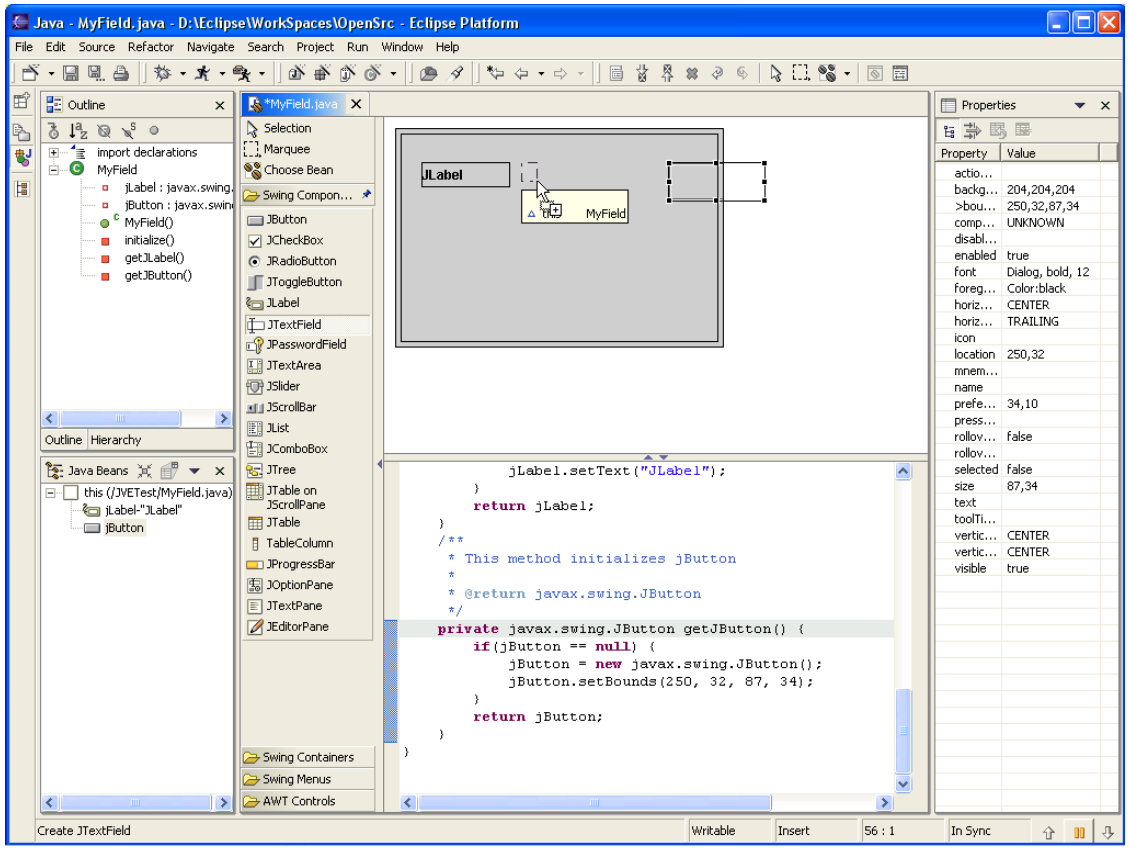
C/C++ Development Tooling

- Building
- Launching
- Debugging
- Code editing
  - content assist
  - content outline
- Extension points to hook into build process



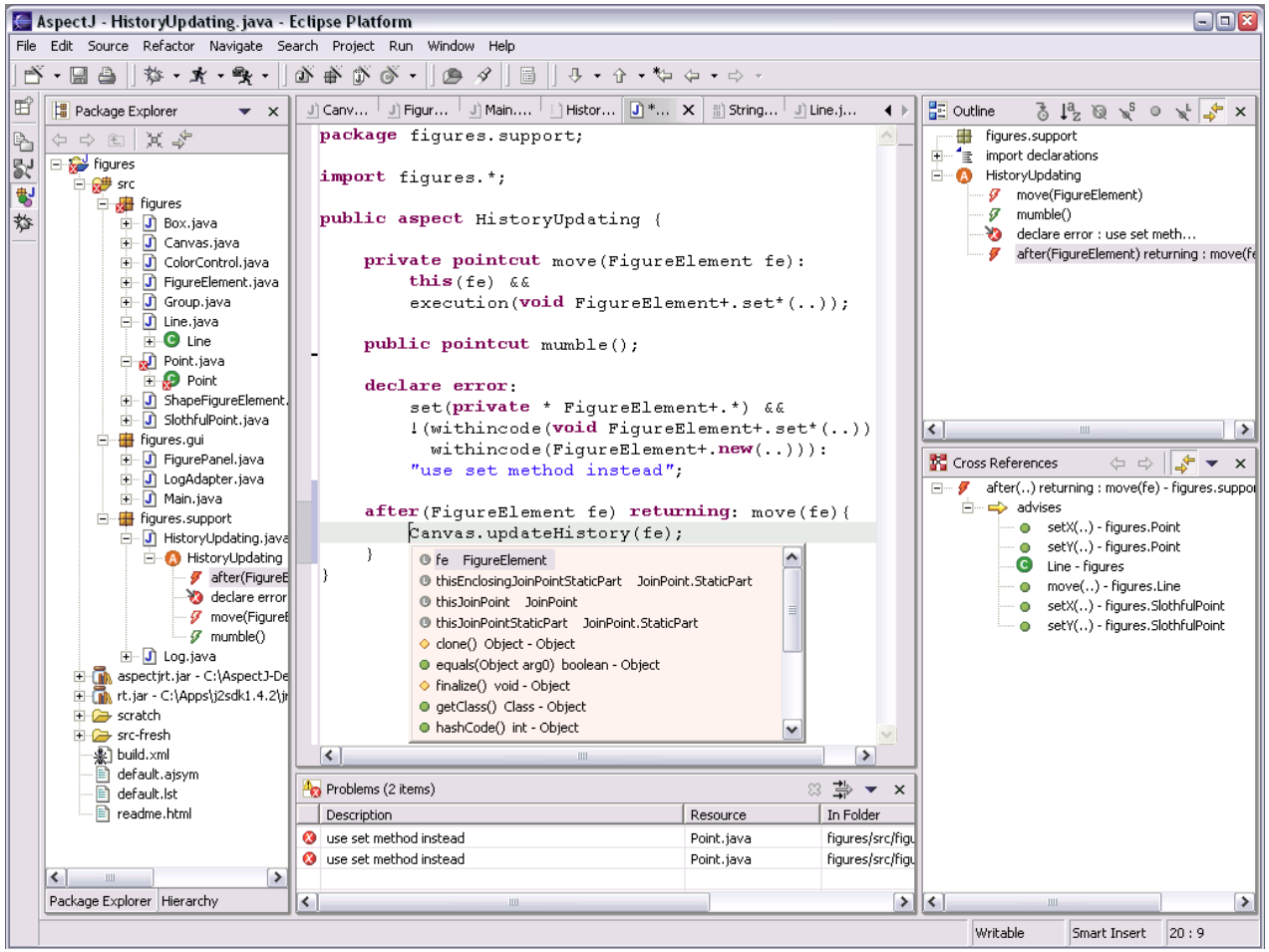
Visual Editor (VE)

- A framework for GUI Builders
- Reference implementations:
  - Swing
  - SWT



AJDT – AspectJ Development tool

- Navigate AspectJ code
- Custom views shows static and dynamic cross cutting effects
- AspectJ aware code assist
- AJDT extends JDT compiler



Eclipse 3.1:

Targeted for 2Q 2005

- **Built to last** - Eclipse has always been a platform for delivering integrated software tools. With a large and growing base of both free and commercial offerings based on Eclipse, it's critical for continued success to maintain API stability and ensure that the platform scales well. This theme includes work to measure and improve the performance of key operations under various loads (number of installed plug-ins, number of resources in the workspace, etc.). This theme also includes consolidation activities where groundwork was laid in 3.0 but needs to be completed and brought into full use.
- **Simple to use** - The Eclipse platform needs to not only provide the features that advanced users demand, but also be something that most users find simple to use. This theme includes ease-of-use reviews of existing features, and work that helps make Eclipse-based products simple to use for users with widely-varying backgrounds and skill sets.
- **Rich client platform** - The Eclipse RCP is a Java-based application framework for the desktop. Building on the Eclipse runtime and the modular plug-in story, it is possible to build applications ranging from command line tools to feature-rich applications that take full advantage of SWT's native platform integration and the many other reusable components that the Eclipse platform provides. This theme includes work to enhance the RCP, and to provide PDE support for developing and deploying RCP-based applications.
- **J2SE 5 support** - This theme covers work to add full support for J2SE 5 to JDT.
- **Large-scale development** - Large software projects are long-term collaborations involving large teams of developers playing a variety of roles. In order to be effective for large projects, software tools and processes must fit well into this reality. This theme includes laying the groundwork in the Eclipse Platform that will enable large teams to make effective use of Eclipse-based products.
- **Broadening the community** - This theme includes work that grows deeper roots into the various OS-specific communities, spreads Eclipse to additional operating environments, and builds bridges to other open source communities.

How to Get Even More

Contribute your expertise

- Report, annotate defects
  - Provide test cases
- Request enhancements
- Build plug-ins to address your priorities
  - Grow the ecosystem: commercial, research, education, open source
- Answer newsgroup questions
- Write articles
- Give presentations
- Enhance the open source technology
- …

For more information

- eclipse community at http://www.eclipse.org/community/index.html
- third party eclipse sites
  - e.g. eclipse plug-in central at http://www.eclipseplugincentral.com/
- eclipse newsgroups
  - news.eclipse.org
- eclipse mailing lists

  http:// www.eclipse.org/mail/index.html

Q & A

Darin_Swanson@us.ibm.com