

ECLIPSE PLUGIN DEVELOPMENT (5 DAYS)

By Dr. Vishwanath Rao

Prerequisites:

Understanding of basic eclipse concepts

Ability to create and deploy simple plug-ins

Prerequisites can be satisfied by taking Eclipse University course: Developing Eclipse Plugins

Day 1

Contents:

1 Eclipse Plug-in Architecture

- Eclipse Overview
- Plug-in architecture
 - Plug-in, Extension Point, and Extension
 - Plug-in manifest
 - Eclipse overview
 - Plug-in features and fragments
- Platform architecture
 - Workspace Component
 - Workspace and Resource API
 - Incremental Project Builders
 - SWT
 - JFace
 - Workbench, Editors, Views, Perspectives
 - Debug component
- PDE Introduction

2 Plug-in Development Environment (PDE) & Eclipse Plug-in Development (Include lab)

- Why PDE?
- Concepts
- Plug-in and Life cycle
- Plug-in structure
- Work with PDE
 - Configure PDE
 - Create a new Plug-in project
 - Plug-in manifest file and editor
 - Dependencies
 - Runtime
 - Extensions and Extension points
 - Running the plug-in
- Plug-in communication
- Plug-in fragments

3. Workbench

- Eclipse's UI-metaphor
- Launching : IApplication
- Windowing: window, menu, page, part
- Advisor classes
- Related links and resources

4. Publishing

- Plug-in publication • Products
- Features
- UpdateSite
- UpdateManager
- Branding, themes, and customizations • Related links and resources

Day 2

5. Plug-ins

- Identity
- Configuration (MANIFEST.MF, plugin.xml) • Lifecycle
- Related links and resources

6. Contributing to Workbench

- Extension points
- Extensions
- Contributions to the Workbench
- First examples of Contributions
- Perspectives
- Views
- Using ActionBarAdvisor versus extension points • Create your own extension point
- Related links and resources

7. Views

- Introduction / concept of a view
- Views: the general-purpose Workbench Part • How to create a custom view
- Related links and resources

8. Commands and Actions

- Introduction
- Menus, Commands, and Handlers • Key bindings
- Types of Actions
- Lazy loading
- Contributing actions to Views
- Visibility and enablement
- Actions and selections
- Related links and resources

9. Editors

- Introduction/concept of an Editor
- Difference between a View and an Editor • Types of Editors
- Implementing an Editor
- Dirty state management
- Contributing and opening an Editor
- Related links and resources

10. Wizards

- Contributing wizards
 - Defining a wizard extension
 - Implementing a Wizard
 - Implementing a WizardPage
 - Customizing a wizard
 - Controlling wizard page progression • Reusable specialized wizard pages •
- Related links and resources

Day 3

11. Enhancing Views

- Drag and Drop in the Eclipse UI
- Understand Eclipse Drag-n-Drop facility
- Understand SWT Drag-n-Drop API
- Program drag source and drop target to add drag and drop support to Eclipse views
- Transfer data within application using Clipboard

12. Understanding Decorators in Eclipse

- What are Decorators?
- Types of Decorations
- Decorator classes
- Defining decorators in plugin.xml
- Defining and Enabling custom decorators

13. Developing Editors (include lab)

- Introduction
- Understand Eclipse JFace Text framework
- Editor Configuration – Customization Points
- Text Editor classes
- The AbstractTextEditor class
- Class relationships
- Model-View-Controller relationships
- How to create custom text editor
- Create a basic text editor
- Add Content Assistant
- Add syntax-based Color Highlighting
- Add Content Formatting

14. Eclipse Modeling Framework (EMF) – Introduction (include lab)

- Understand Eclipse Modeling Framework (EMF)
- The ecore model
- The Generator Model
- Generating EMF Model from Annotated Java Code
- Generate Model implementation code
- Generate model.edit and model.editor plug-ins

15. Xpand Setup.

- Defining an EMF metamodel
- Creating an EMF project
- Defining the (meta)model
- Generating the EMF tooling
- Setting up the generator project
- Defining an Example Data Model .
- Using Dynamic EMF
- Generating code from the example model

Day 4

16. The workflow definition

- Running the workflow
- Templates
- Running the generator again
- Checking Constraints with the Check Language
- Defining the constraint .
- Integration into the workflow file
- Extensions
- Expression Extensions .
- Java Extensions .

17. EMF – Customize Code Generation (include lab)

- Learn techniques to customize code generation in EMF
- Understand GenModel properties
- Edit GenModel properties to customize code generation
- EMF.Edit framework overview
- Modify presentation behavior

- EMF Reflective API overview
- Create model instance using Reflective API
- Serialize model in XMI format
- Introduction to JET
- Use of JET in EMF

18. XTEXT Grammar Language

- Configuration
- Integration with Java
- Generating Artifacts
- Integration with EMF
- Editor Support

19. Extending the Java Development Tools (include lab)

- Java Development Tooling (JDT) and its uses
- Major components of JDT
- Java Model
- Java Model Elements and relationship
- About JavaCore
- About ICompilationUnit
- Abstract Syntax Tree (AST)
 - AST, ASTParser, and AST Node
 - Handling element change notifications
- Ways of extending the JDT User Interface

Day 5

20. XTEND Programming

- Extension methods
- Lambda Expressions
- ActiveAnnotations - annotation processing on steroids
- Operator overloading
- Powerful switch expressions
- Multiple dispatch
- Template expressions
- No statements

21. Introduction to RCP

- Brief history
- Benefits / examples
- Anatomy of an RCP application
- Internal architecture
- Concept of Plug-in based modularization
- First glance at PDE
- Related links and resources
- Optional: introductory demos (plug-ins/minimal RCP)

22. RCP Application lifecycle

- Physical artifacts
- Runtime artifacts
- Development time artifacts

23. Internal structure

- Runtime and OSGi
- Generic Workbench
- Plug-ins
- Start-up sequence
- Extension points, extensions, contributions • SWT / JFace
- Related links and resources