Eclipse Modeling Framework (EMF)

Basic course

EMF is an Eclipse-based modeling framework that provides a solid base for application development through the use of pragmatic modeling and code generation facilities.

It is used for building Code generators, Graphical diagramming frameworks, Model Transformation, Validation, and Search. Eclipse E4 and projects such as Xtext and Sirius are entirely based on EMF.

EMF has transformed the modeling tools industry. Leading modeling tool vendors such as Borland and IBM have based their products on EMF. Almost every Eclipse-based modeling project is based on EMF.

Workshop

This workshop is an introduction to EMF's Modeling and Code generation capabilities. Participants will have an opportunity to build models based on the EMF ECore model, Generate Java code and customize it.

Key concepts namely Persistence framework, Proxy resolution, and Dynamic EMF are covered in-depth with appropriate hands-on exercises.

Prerequisite

Knowledge of Java is essential
Experience of using Eclipse IDE for Java development
Prior experience of UML is a plus.

Modalities

Duration: 8 hours
Structure: Instructor-led with 100% hands-on labs
Participants: Maximum 8 per-workshop
Equipment: Participants supply their own computer with the latest Eclipse with Modeling tools installed.

Agenda

Introduction

- Introduction to the Eclipse Modeling Framework (EMF)
- EMF Workflow - Meta-modeling, Java Code generation, Testing using the Reflective ECore editor
- Code Generation
    - Factory, Package, Adapter Factory and Switch classes
    - Customizing Generated code.

ECore Kernel

- EClass and EObject
- Attributes - Single and Multi-Valued
- References - Non-Containment, Containment, Bidirectional, and Map
- DataTypes, Operations, and Annotations.

Persistence framework

- Persistence API - ResourceSet, Resource, and URI
- EMF Package Registry.

Proxy resolution

- Proxy resolution in EMF
- Influencing Proxy resolution - Resolve Proxies, and Containment Proxies fields.

Dynamic EMF

- EMF Dynamic API.