

Développement d'un client riche avec SWT et Eclipse RCP

Référence : SWT

Face aux critiques visant les bibliothèques traditionnelles de création d'IHM Java (AWT ou Swing), de nouvelles API (SWT / JFace) Open Source ont vu le jour. Plus rapides et moins consommatrices en ressources, ces nouvelles bibliothèques s'imposent peu à peu auprès de la communauté des développeurs Java.

L'environnement de développement Eclipse est l'application la plus en vue qui utilise ces API, mais au-delà d'une simple utilisation, Eclipse permet d'intégrer vos propres développements d'interfaces graphiques et surtout d'en automatiser le déploiement.

Ce cours vous permettra, non seulement de créer vos IHM Java en utilisant les puissantes bibliothèques SWT et JFace, mais aussi d'automatiser le déploiement et les mises à jour de vos applications. Ce cours apportera des réponses pratiques à tous ceux qui se demandent : "Quelle API graphique utiliser en Java ?", "Qu'est-ce qu'Eclipse RCP ?", "Quelle différence entre Eclipse RCP et un plug-in Eclipse ?".

Vous allez apprendre à :

- Connaître différents modes de déploiement automatique d'applications Java
- Être capable de développer une IHM avec SWT
- Identifier les apports de l'API JFace dans l'élaboration d'interfaces graphiques
- Comprendre ce qu'est un client riche et l'apport d'Eclipse RCP (Rich Client Platform)
- Savoir personnaliser Eclipse en lui ajoutant des fonctionnalités
- Créer ses propres plug-ins et les internationaliser
- Comprendre le modèle de programmation d'Eclipse

Durée : 4.0 jours - 28.0 heures

Audience :

Développeurs, architectes

Pré-requis :

Connaissances fondamentales de Java ou avoir suivi le cours JOD ou IJOP

Expérience pratique de la plate-forme Eclipse

Méthode pédagogique : 50% de travaux pratiques

Programme détaillé :

Technologies et architectures sur le poste client

- Du client / serveur au client riche
- Standards d'implémentation et de communication
- Problématiques de déploiement

Fondamentaux SWT

- Containers de base : Display et Shell
- Principaux composants SWT
- Le conteneur Composite et ses Layouts associés
- Gestion des événements : Listener, Adapter et Event

Composants évolués

- Structures complexes : tableaux et listes déroulantes ou arborescentes
- Composants spéciaux vs. composants natifs
- Composants et édition graphique

API JFace

- JFace et SWT
- Gestion des ressources
- Actions, menus et assistants (Wizards)
- Gestion des préférences
- Mise en place des viewers et mapping avec un modèle métier

Déployer une application

- Portabilité des applications SWT
- L'approche Java WebStart
- L'approche socle applicatif : Eclipse

L'architecture Eclipse

- Présentation du modèle de programmation d'Eclipse
- Contribuer en étendant les plug-ins de base

- Créer ses propres points d'extension

Développer un plug-in Eclipse

- Cycle de développement d'un plug-in
- Architecture Eclipse et structure détaillée du plug-in
- Manifest.mf, la carte d'identité du plug-in
- Définition des extensions et points d'extension via plug-in.xml
- Cycle de vie d'un plug-in et classe Activator

Internationaliser un plug-in

- Identifier et remplacer les libellés à internationaliser
- Automatisation de l'internationalisation
- Stratégie de déploiement via les fragments

Industrialiser la création des plug-ins

- PDE, l'environnement de développement
- Mises en place de tests unitaires avec SWTBot
- L'alternative ANT pour les phases de build et l'intégration continue

Eclipse, un client riche

- Faire d'Eclipse un client RCP
- Personnaliser le client RCP en fonction de la plate-forme
- Automatiser les mises à jour du client