



Tutoriel développement Java - JFace

Introduction

- **Introduction**

Dans ce tutoriel, vous découvrirez comment mettre en œuvre une interface graphique Java à l'aide de JFace.

Pour cela, nous utiliserons l'outil **WindowBuilder** sous Eclipse.

1

Présentation

- **Présentation**

Les développeurs Eclipse ont donc conçu **SWT** (Standard Widget Toolkit) comme une alternative à **AWT** et **Swing**, considérés comme trop lents pour une application Java qui requiert des performances optimales.

JFace est à **SWT** ce que **Swing** est à **AWT** : un jeu de composants graphiques construisant sur les fondations d'une bibliothèque graphique proche du système.

1

Présentation

- **Présentation**

JFace est basé sur **SWT**, et donc comme ce dernier dépend d'Eclipse, ce qui se retrouve dans ses paquets de **type** **org.eclipse.jface.***

JFace apporte à **SWT** certaines fonctionnalités de **Swing**, à commencer par une architecture MVC (Modèle-Vue-Contrôleur).

JFace étend donc SWT, sans empêcher l'application de se servir directement de SWT.

1

Présentation

- **Présentation**

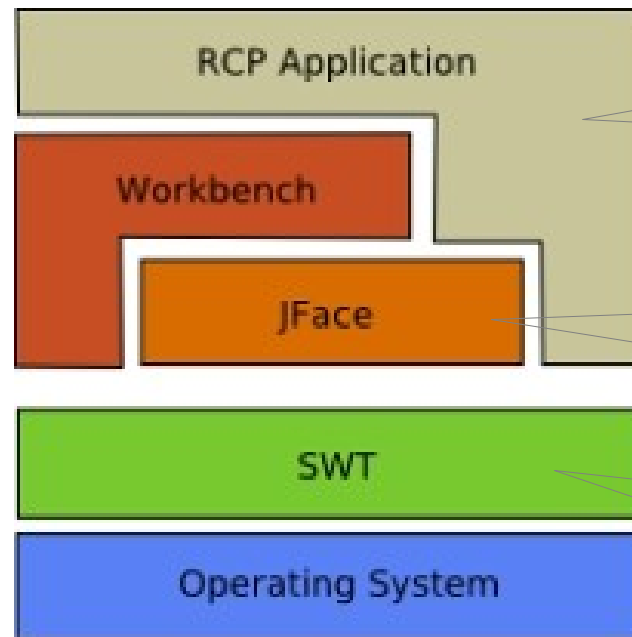
JFace se charge d'encapsuler certains comportements et traitements fondamentaux pour que le développeur se dégage du code. Par exemple, pour créer une boîte d'information :

- avec **SWT**, il faut faire appel à plusieurs classes
- avec **JFace** cela se fait avec une méthode et quelques arguments.

1

Présentation

• Présentation



L'application s'appuie sur JFace et SWT

Extension et encapsulation de SWT

La boîte à outils connectée aux composants natifs

1

Présentation

- **Présentation**

JFace propose également un framework de liaison de données (l'API **Data Binding Framework**), qui cherche à améliorer la gestion des données et leur mise en place au sein d'interfaces JFace/SWT.

L'idée est d'automatiser les flux de données entre la couche métier et la couche de présentation.

1

Présentation

- **Présentation**

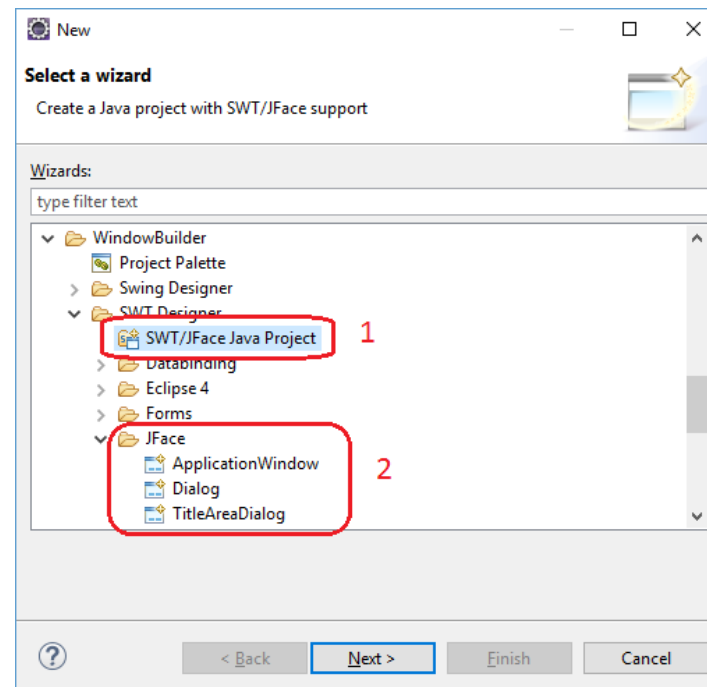
Ce framework dépend du paquetage **org.eclipse.equinox.common**, et non plus de SWT, car il a été conçu pour ne pas être attaché à une seule bibliothèque graphique (même si l'utilisation avec SWT est facilitée).

2

WindowBuilder

- **WindowBuilder**

Comme pour Swing et SWT, il est possible de dessiner une application **JFace** grâce à **WindowBuilder**.

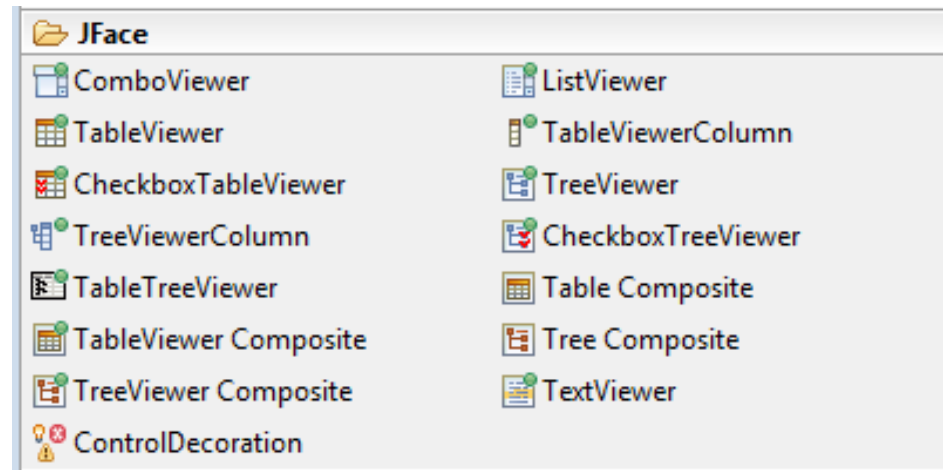


2

WindowBuilder

- **Les composants de JFace**

Voici les composants de JFace

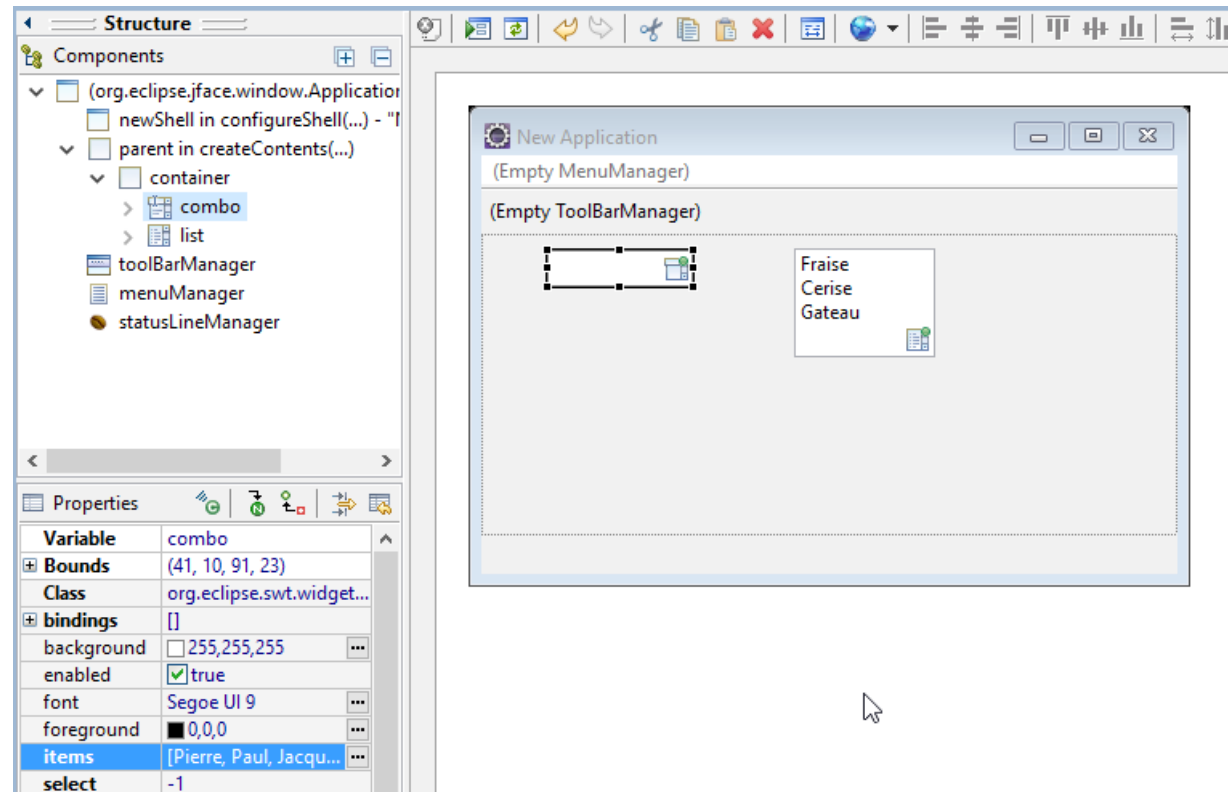


2

WindowBuilder

• Les composants de JFace

Exemple de composants JFace :





Fin du tutoriel Java
