

Eclipse



Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

Caractéristiques de l'IDE Eclipse

Presque tout dans Eclipse est un plugin. Nous pouvons étendre les fonctionnalités d'Eclipse IDE en ajoutant des plugins à l'EDI, peut-être pour un langage de programmation supplémentaire ou un système de contrôle de version ou UML.

- Prend en charge divers outils de connaissance source tels que le pliage et la navigation par hyperliens, le classement, le navigateur de définition de macro, l'édition de code avec coloration syntaxique.
- Fournit un excellent outil de débogage de code visuel pour déboguer le code.

Eclipse a une interface utilisateur avec une fonction de glisser-déposer pour la conception d'interface utilisateur.

- Prend en charge le développement de projets et le cadre administré pour différentes chaînes d'outils, le cadre de création classique et la navigation dans les sources.

Java Eclipse IDE dispose d'une fonction JavaDoc à l'aide de laquelle nous pouvons créer automatiquement une documentation pour les classes de notre application.

Installer et configurer Eclipse Java IDE

Afin d'installer et de configurer Eclipse IDE pour le développement Java, nous devons d'abord nous assurer que nous avons une version JDK appropriée sur notre machine.

Voici les étapes pour installer et configurer Eclipse Ide pour le développement Java.

Étape 1: Installez JDK

Pour qu'Eclipse fonctionne correctement, notre système doit avoir JDK installé. Eclipse utilise JDK pour le développement Java.

Une fois que les applications que nous construisons à l'aide d'Eclipse IDE sont construites et exécutées avec succès, nous devrions avoir des valeurs définies pour deux variables d'environnement sur notre système, à savoir «PATH» et «CLASSPATH», qui sont définies à la suite de l'installation de JDK.

Étape 2: Télécharger Eclipse

Téléchargez l'installable Eclipse depuis [ce site](#).

Étape 3: décompressez

Il n'y a pas de séquence d'installation en tant que telle pour Eclipse, il vous suffit de décompresser le contenu du package téléchargé, puis d'exécuter Eclipse et vous êtes prêt pour le développement Java en utilisant Eclipse IDE.

L'espace de travail

WorkBench

La plateforme Eclipse est composée de deux couches logicielles principales : la couche modèle et la couche d'interface utilisateur (IHM). Le modèle est également appelé Workspace. Il contient une collection de ressources identifiées par des projets, des répertoires ou des fichiers.

L'interface utilisateur ou Workbench définit la présentation visuelle des ressources Eclipse. A l'intérieur d'un Workbench, les perspectives sont utilisées pour contrôler la visibilité de chaque élément du modèle et de l'interface utilisateur via le concept d'actions et de vues. Ces contrôles permettent de naviguer ou de modifier le Workspace courant tout en personnalisant l'environnement. Le développeur a également la possibilité d'implémenter de nouvelles perspectives et de nouvelles vues. A ce titre, il doit être en mesure d'identifier la nature exacte des composants qu'il souhaite développer ou étendre afin d'utiliser les API adéquates.

Wizard

Les wizards Eclipse sont souvent utilisés dans les plugins Eclipse pour générer le code d'un composant. Un wizard est constitué de plusieurs wizard pages. Par exemple, le wizard "New java class" accessible par le menu File/New/Class est constitué d'un wizard page qui permet de générer le composant "classe Java ».

Perspective

L'une des principales particularités d'Eclipse est l'utilisation de "perspectives", qui permettent d'adapter l'interface à la manière dont l'utilisateur veut voir le code source. Les perspectives servent à lier les deux couches autour desquels Eclipse est architecturé : l'interface utilisateur (l'atelier, ou workbench, avec ces éditeurs et visualiseurs) et le modèle (l'espace de travail, ou workspace, qui présentent les ressources disponibles : fichiers, projets, répertoires...).

Ainsi, selon la perspective, voire le langage utilisé, l'interface d'Eclipse, tant au niveau des fenêtres que des menus, peut être entièrement modifiée selon les tâches en cours.

Il existe principalement quatre perspectives, accessibles directement via le bouton à l'extrémité gauche de l'interface : Resource, Java, Java Browsing (navigation Java) et CVS Repository Exploring (exploration du référentiel CVS). D'autres sont accessibles via le bouton Open a Perspective : Java Type Hierarchy, Plug-in Development... Elles sont toutes axées autour de Java. Lors du premier démarrage d'Eclipse, on se trouve en perspective Resource. Il suffit de sélectionner la perspective Java Browsing pour se rendre compte des changements côté fenêtres et menus. Bien entendu, chaque perspective peut-être modifiée à souhait - on peut même créer ses propres perspectives...

La combinaison de vues et d'éditeurs rend Eclipse extrêmement flexible : on peut facilement créer son propre environnement de programmation, notamment grâce à une foule de plug-ins...

Les Vues

Les vues représentent l'élément de base permettant de naviguer dans une hiérarchie d'informations. Elles sont typiquement utilisées pour ouvrir un éditeur de texte, afficher des propriétés sur la fenêtre active ou récupérer des informations spécifiques sur l'application. Par défaut, l'IDE propose un certain nombre de vues telles que l'explorateur de package, la fenêtre des propriétés, l'explorateur de classes, etc ...