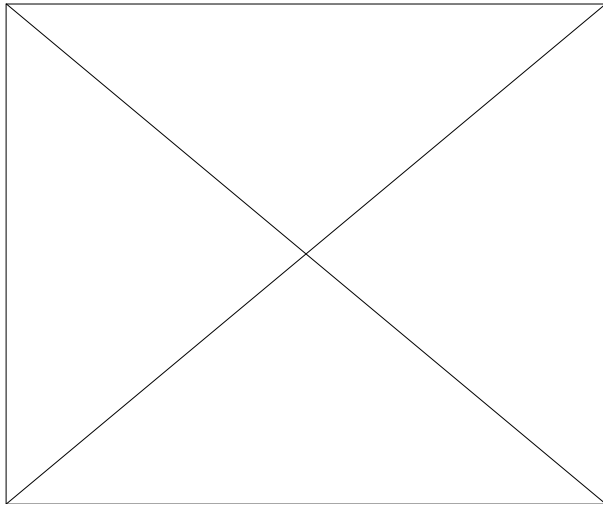


Java, Une application graphique complète sans efforts avec NetBeans 6.1

IDE



L'interface graphique est un élément important d'une application, elle rend son utilisation plus facile et plus intuitive.

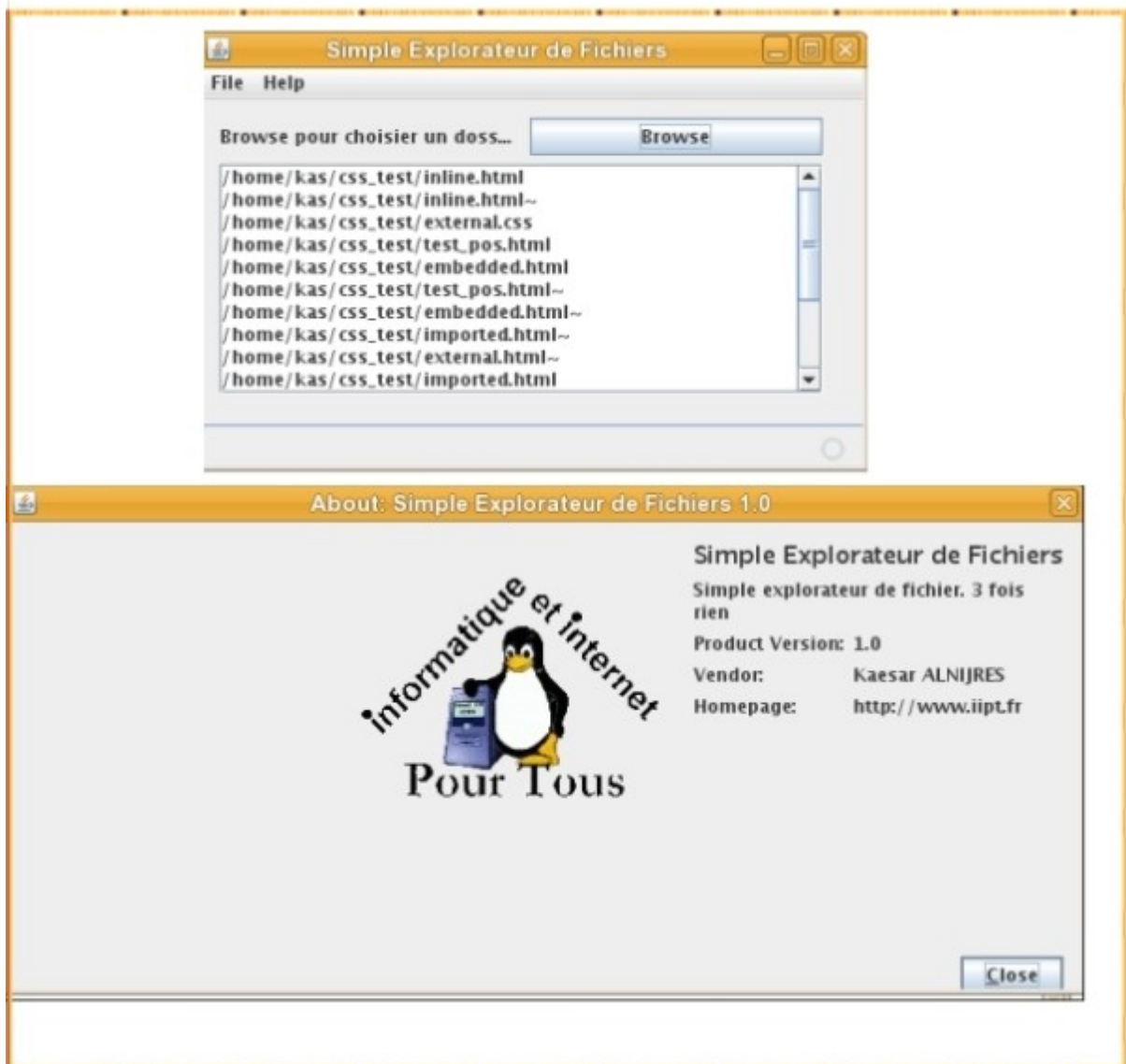
Malgré les avantages non négligeables de l'interface graphique, il arrive souvent qu'un développeur Java, hésite avant de construire de telle interface car l'avantage majeur de Java de pouvoir s'exécuter sur de multiples plateformes se voit contrebalancer par le problème de rendu visuel sur les différents environnements graphiques des systèmes d'exploitation et le problème de positionner des composants graphiques sur des layouts.

NetBeans 6.1 apporte une bonne solution au problème de l'interface graphique en adoptant le "GroupLayout" comme un layout par défaut. GroupLayout est disponible avec Java SE 6.

NetBeans va encore plus loin en fournissant des bibliothèques nécessaires pour l'exécution des programmes graphiques sous Java SE 5 ou JDK 1.5

1. Un Exemple

Pour démontrer la capacité graphique de NetBeans nous allons dans ce petit tutoriel programmer un mini explorateur de fichiers (vraiment mini, l'objectif étant ici l'interface graphique).



2. Logiciels nécessaires :

Java SE 6 (de préférence) ou Java 5 SE
NetBeans 6.1

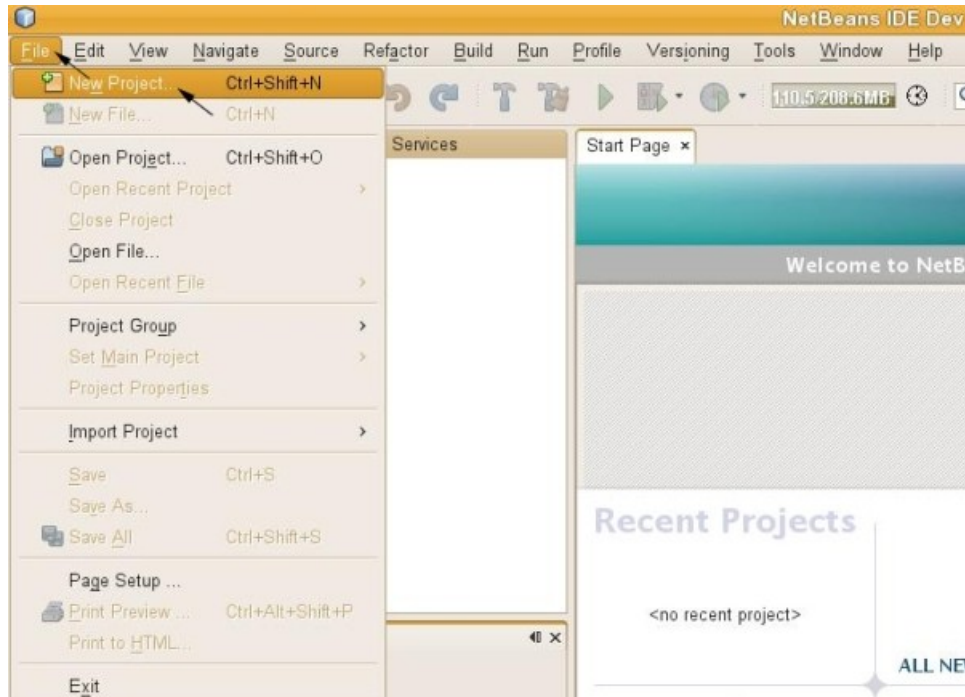
3. Les étapes de création de l'application :

- 1.Création d'un projet dans NetBeans
- 2.Placer des composants graphiques nécessaires sur une fenêtre (Form)
- 3.Ajouter quelques lignes de code pour le fonctionnement
- 4.Tester l'application
- 5.Préparer l'application pour une diffusion

4. Création d'un projet NetBeans :

Pour NetBeans une application doit être construite dans un projet. Plusieurs modèles de projet sont disponibles permettant de construire une partie importante de la structure d'une application dans un domaine précis.

File -> New Project (ou un clic sur l'icône New Project)



5. Java Desktop Application :

Sélectionner Java -> Java Desktop Application -> un clic sur Next

Note :

Java Desktop Application utilise Swing Application Framework.

Note :

L'application produite avec Java Desktop Application a une structure complète. Elle comporte une barre de menus, une barre de progression, une fenêtre de présentation dans About etc.

Note :

Les fichiers nécessaires (y compris les jar sont ajoutés automatiquement dans le projet)
Dans le projet sont ajoutés des fichiers sources Java, fichiers Meta, fichiers de propriétés, script Ant etc



6. Choisir un nom et une location pour le projet

1. Entrer un nom pour le projet dans "Project Name" ici "ExplorateurDeFichiers"

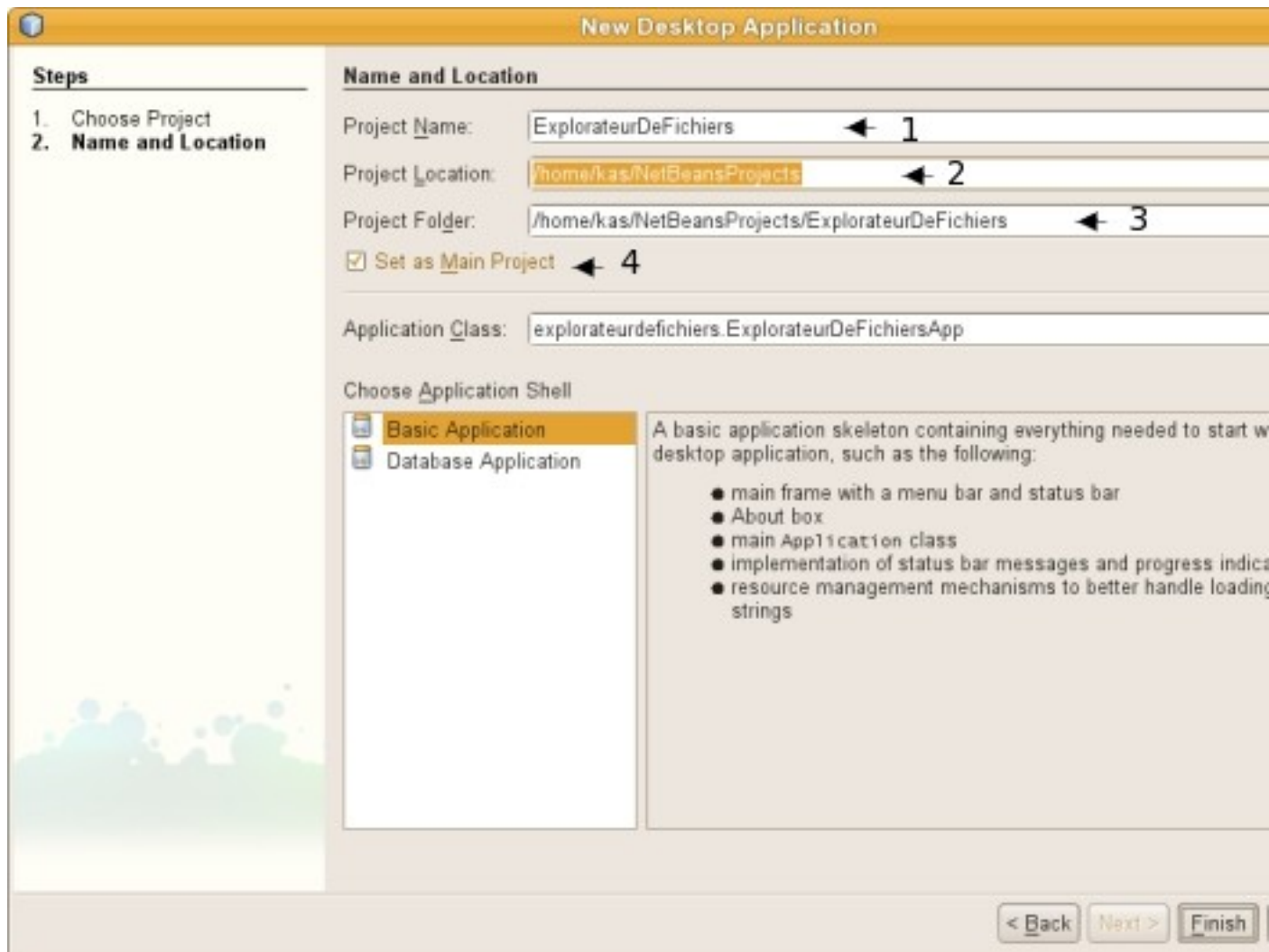
2. NetBeans propose par défaut : dossier_de_l'utilisateur/NetBeansProjects comme une place pour ses projets. Il est possible de modifier ce choix.

3. Entrer un dossier pour le projet dans "Project Folder". Normalement le dossier aura par défaut le nom du projet

4. Laisser la case "Set as Main Project" cochée

5. Laisser sélectionnée "Basic Application"

Un clic sur Finish



7. Le projet est créé :

1.ExplorateurDeFichiers est dans la fenêtre Project

2.La fenêtre "Inspector" montre les composants de l'application (Graphiques et non graphiques)

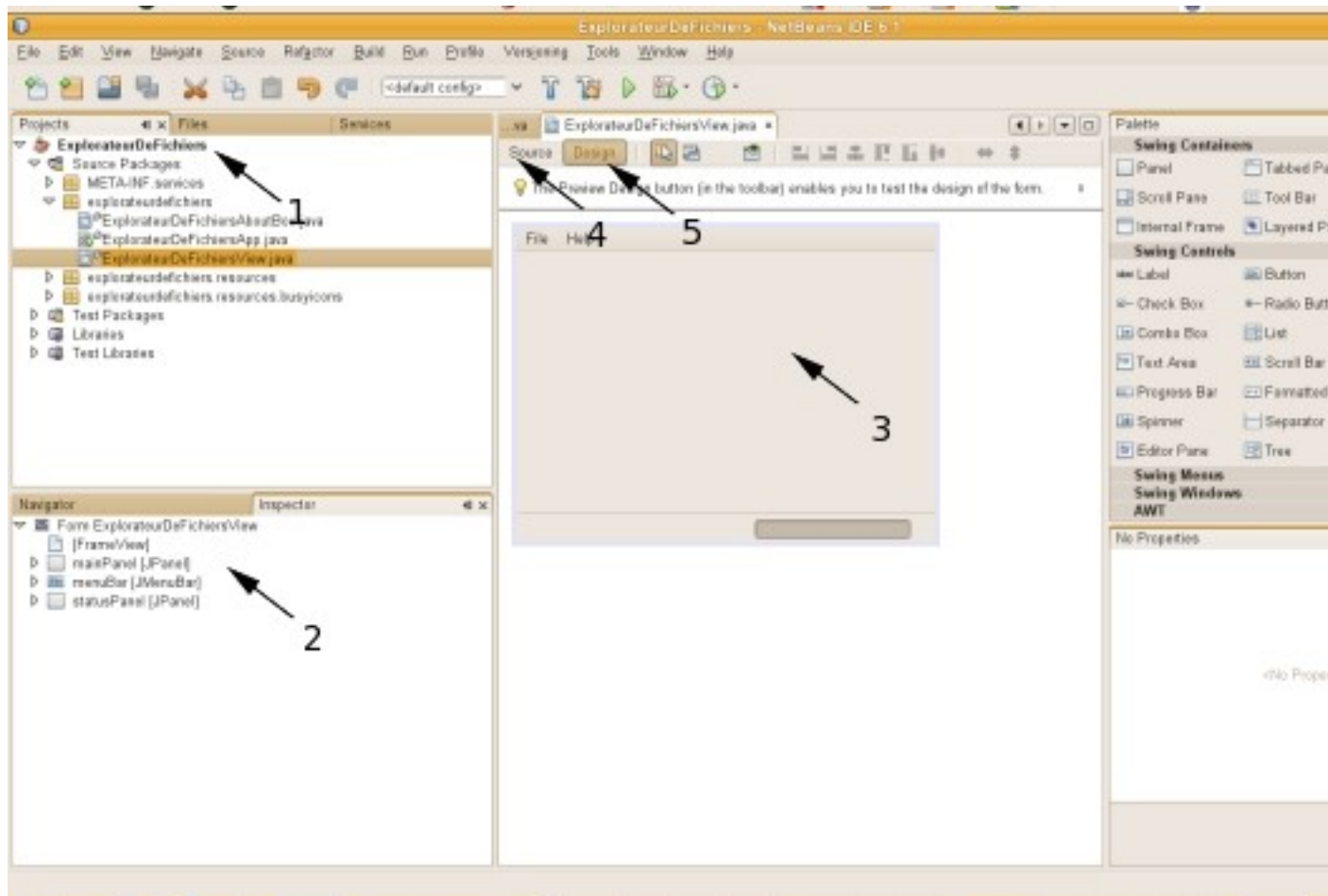
3.Un espace pour le travail graphique (FrameView)

4.Le bouton "Source" permet de voir le fichier source produit par NetBeans et d'y ajouter des lignes de code

5.Le bouton "Design" permet de revenir sur l'espace graphique à partir du fichier source

6.La palette des composants graphiques

7.L'ensemble est dans une fenêtre (Form)



8. L'espace graphique (FrameView)

C'est la fenêtre principale de l'application et c'est aussi la place pour ajouter des composants. Ses dimensions sont modifiables et s'adaptent automatiquement selon ses composants.

Note :

Pour modifier les dimensions de l'application

Sélectionner -> Cliquer (garder cliquer) -> déplacer

Note :

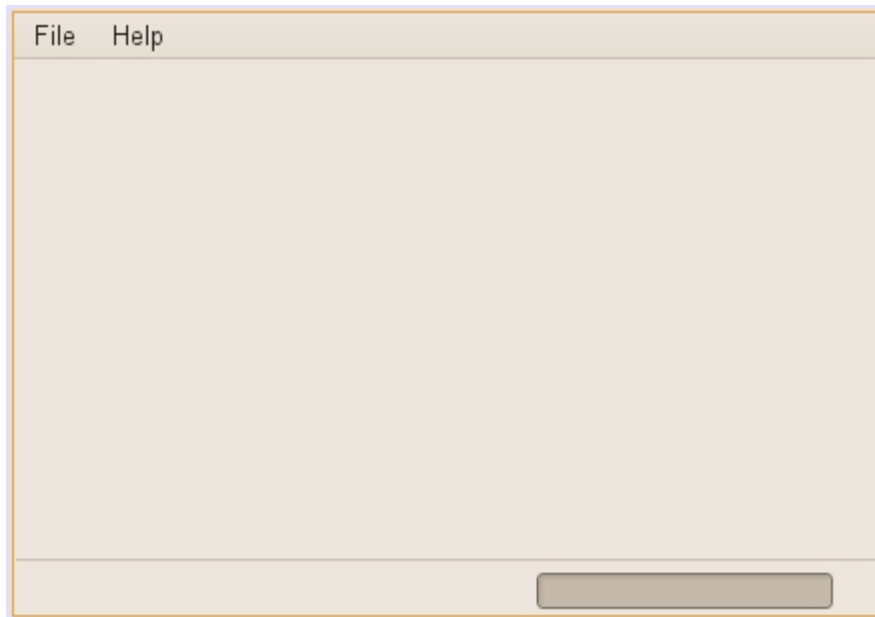
Il est possible de supprimer un (ou plusieurs) des composants ajoutés par défaut

Un clic sur un composant -> Delete (ou Suppr) sur le clavier

Note :

Il est possible de personnaliser tous les composants

Un clic sur un composant et modifier ses propriétés dans la fenêtre "Properties"



9. Ajouter une étiquette (JLabel) :

Avant d'ajouter notre premier composant, un petit mot concernant le positionnement automatique (Snapping). Des règles magnétiques de positionnement sont affichées pour aider au positionnement des composants. NetBeans affiche des coordonnées horizontales et verticales avec des suggestions pour placer un composant par rapport aux autres composants (déjà placés) et aux marges de la page

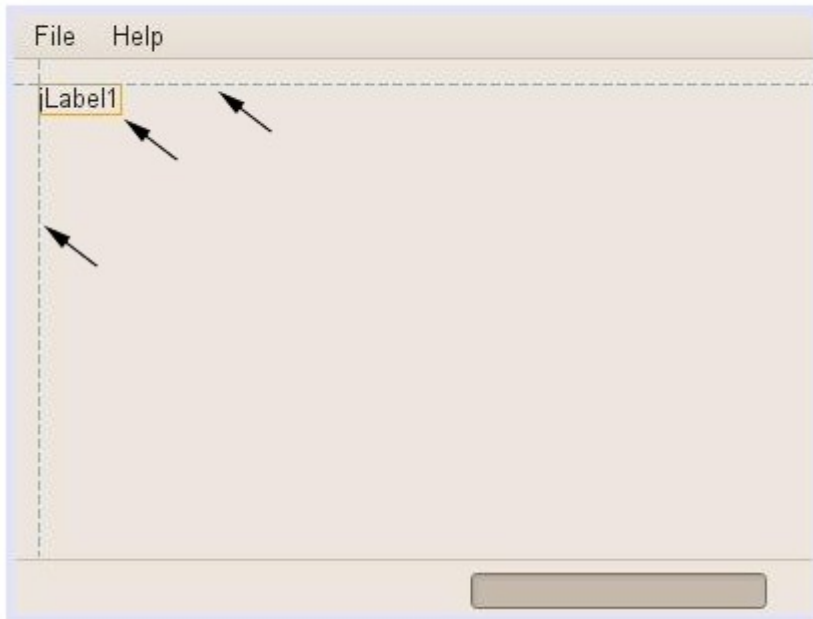
Le positionnement de GroupLayout est similaire au positionnement absolu dans le sens où on place un composant là où on désire le voir WYSIWYG.

Un clic dans la palette sur Label -> Sélectionner une place sur l'espace de travail -> un clic

Note :

Pour annuler un composant "ESC" ou Échappe

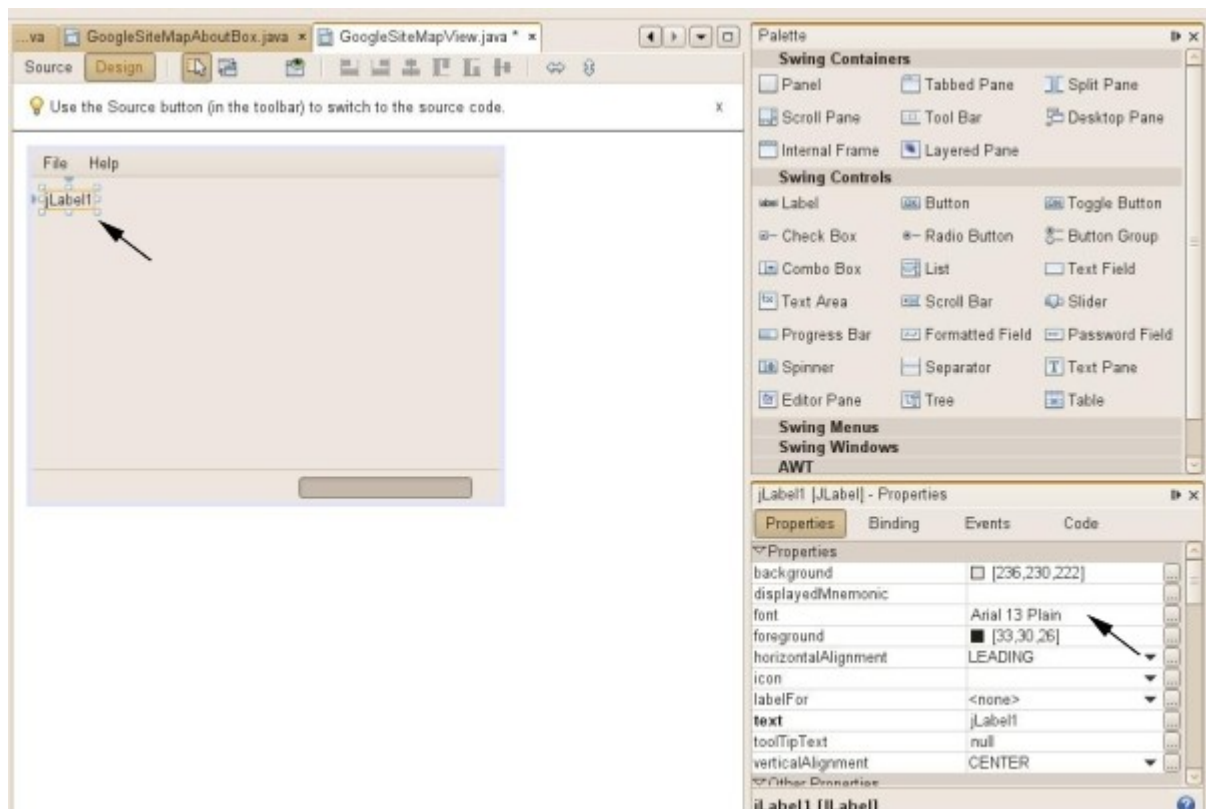
- 💡 Snap at the preferred distance from the container's left border.
- 💡 Snap at the preferred distance from the container's upper border.



10. JLabel est ajouté :

Quand on clique sur l'espace de travail JLabel est placé à l'endroit désiré.
Il est possible maintenant de modifier ses dimensions, son titre, couleur etc.

La fenêtre de (Properties) est maintenant affichée permettant de personnaliser des composants et des conteneurs.



11. JLabel est ajouté dans le fichier source

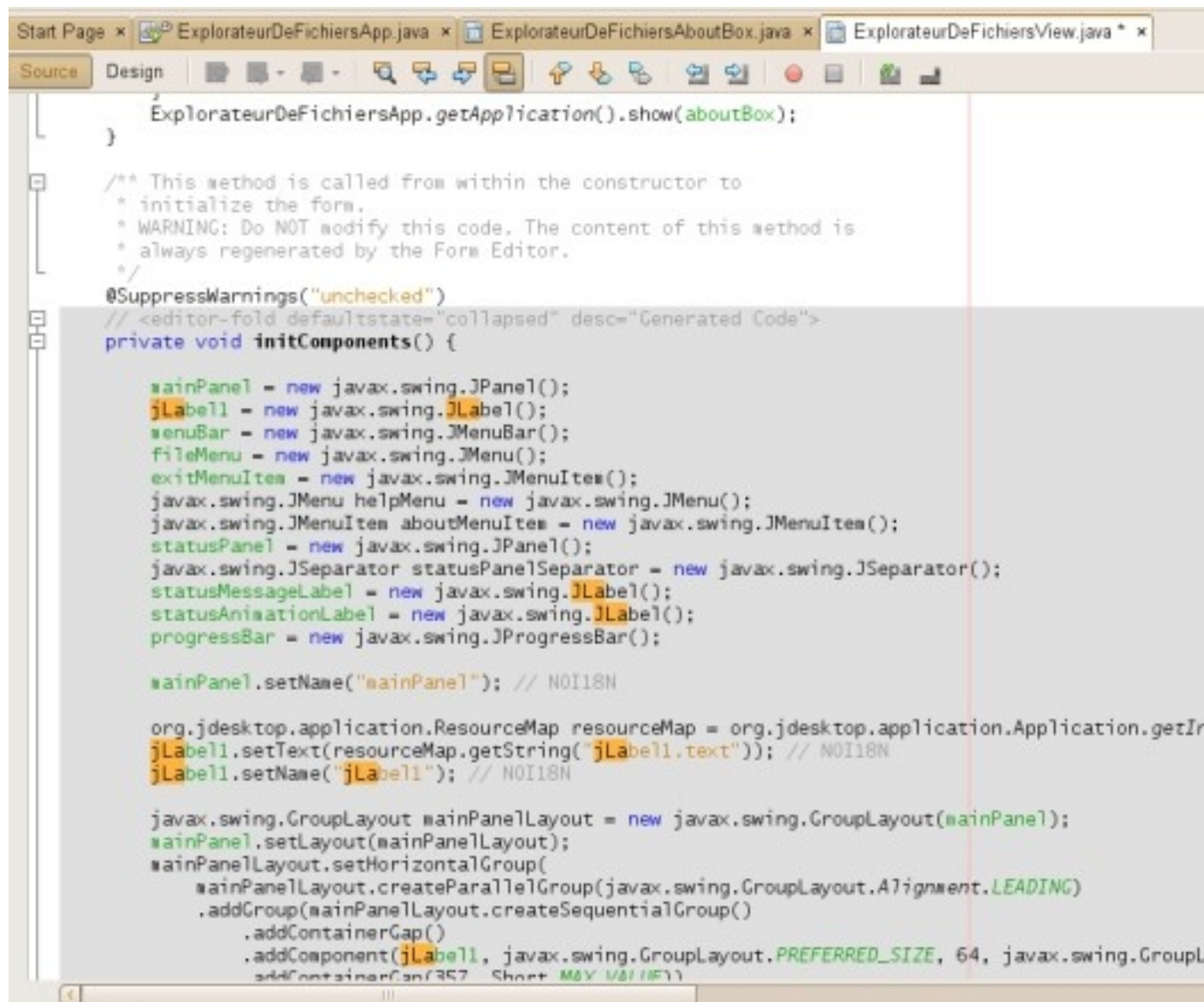
Un clic sur "Source"

Note :

Dans la fenêtre source

Les parties de code en grisées ne sont pas modifiable et générées automatiquement

Un clic droit sur l'éditeur -> Insert Code permet de lancer un wizard pour générer un constructor, des setter et getter etc.



```
ExplorateurDeFichiersApp.getApplicati...
ExplorateurDeFichiersAboutBox.java
ExplorateurDeFichiersView.java *
Source Design
ExplorateurDeFichiersApp.getApplicati...
}
/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    mainPanel = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    menuBar = new javax.swing.JMenuBar();
    fileMenu = new javax.swing.JMenu();
    exitMenuItem = new javax.swing.JMenuItem();
    javax.swing.JMenu helpMenu = new javax.swing.JMenu();
    javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
    statusPanel = new javax.swing.JPanel();
    javax.swing.JSeparator statusPanelSeparator = new javax.swing.JSeparator();
    statusMessageLabel = new javax.swing.JLabel();
    statusAnimationLabel = new javax.swing.JLabel();
    progressBar = new javax.swing.JProgressBar();

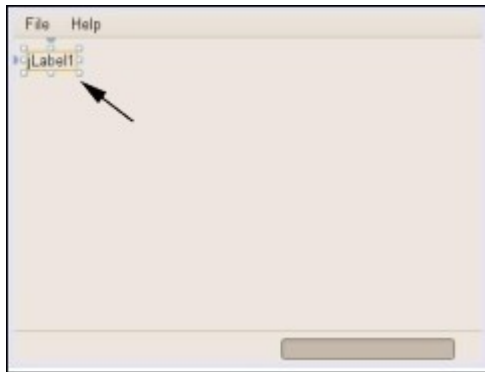
    mainPanel.setName("mainPanel"); // NOI18N

    org.jdesktop.application.ResourceMap resourceMap = org.jdesktop.application.Application.getI...
    jLabel1.setText(resourceMap.getString("jLabel1.text")); // NOI18N
    jLabel1.setName("jLabel1"); // NOI18N

    javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
    mainPanel.setLayout(mainPanelLayout);
    mainPanelLayout.setHorizontalGroup(
        mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 64, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
    );
    mainPanelLayout.setVerticalGroup(
        mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 64, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
    );
}
```

12. Modifier les dimensions d'un composant

Un clic sur le composant -> des carrés sont affichés autour du composant -> un clic sur un carré -> garder le bouton pressé -> déplacer selon les dimensions désirées.



13. Changement des autres propriétés de JLabel

JLabel au moment de sa création a jLabel1 comme nom de variable et jLabel1 comme texte affiché.

Il est possible dans la fenêtre de propriétés de modifier tous les propriétés de jLabel1 ; son nom, son texte, sa couleur etc.

Nous allons de suite modifier le texte de jLabel1 et au lieu de "jLabel1" avoir "Browse pour choisir un dossier"

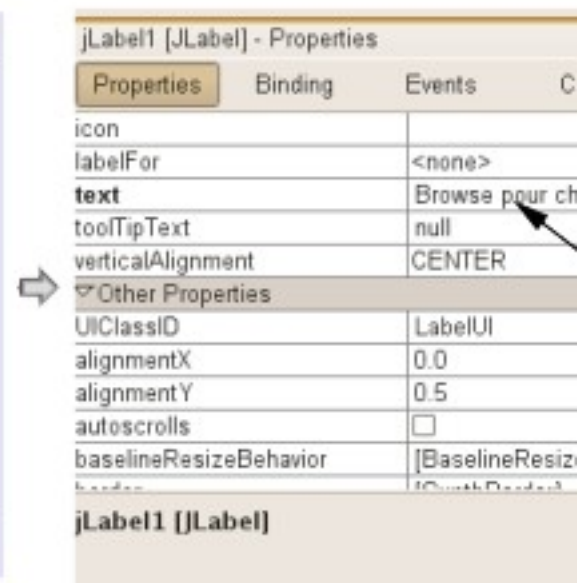
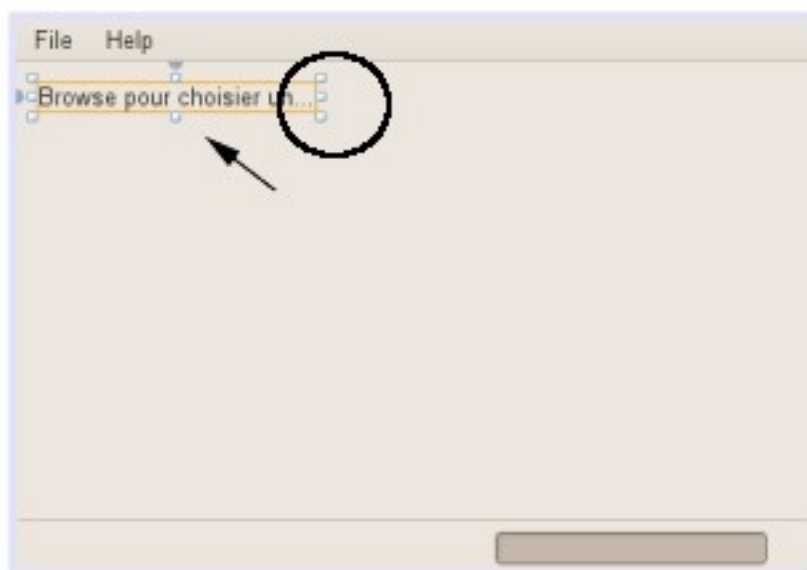
Un clic sur jLabel1

Dans la fenêtre de propriétés (Properties) -> Un clic sur le champ -> text -> entrer "Browse pour choisir un dossier" -> un clic sur ENTER

Penser ici à agrandir "jLabel1" en tirant sur les petits carrés

Note :

Pour modifier le texte d'un label -> Double clic (au lieu de passer par la fenêtre de propriétés) ou un clic droit -> Edit Text



14. Penser à sauvegarder

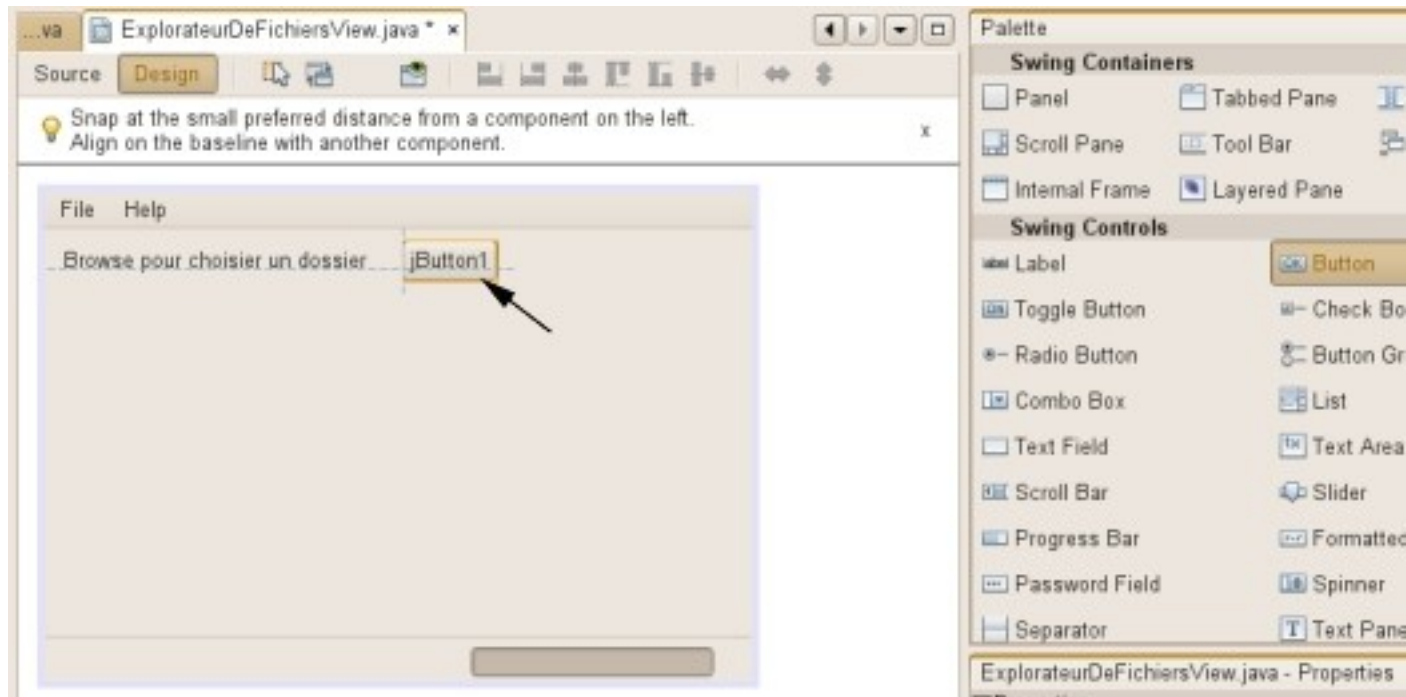
Un clic sur le menu File -> Save All

15. Ajouter un bouton

Un clic dans la palette sur "Button" et un clic par la suite sur l'espace de travail

Note :

Les règles sont affichées pour le positionnement automatique par rapport aux marges, à JLabel1 pour aligner la base de textes dans les deux composants etc.



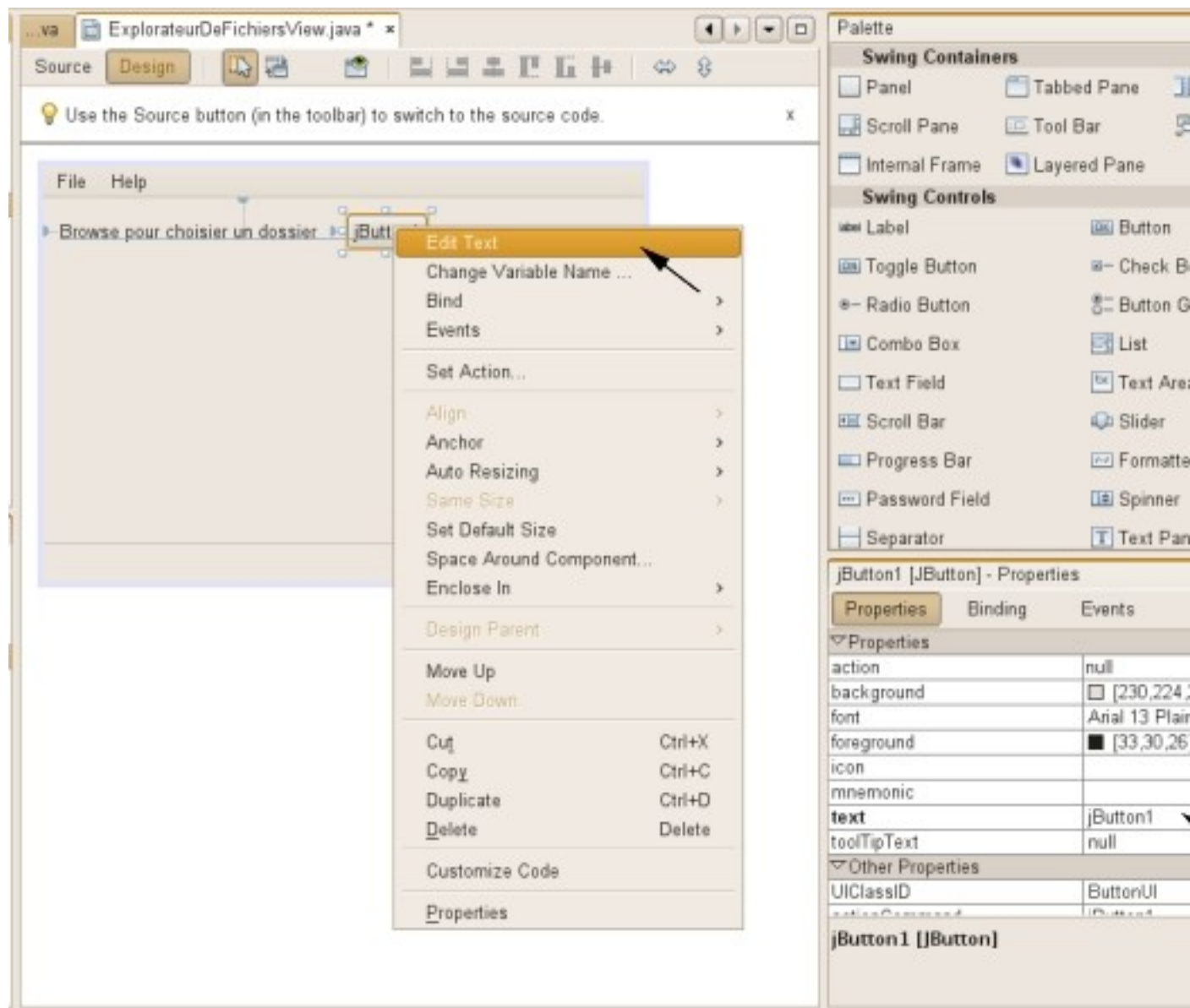
16. Modifier le titre du bouton

Pour modifier le texte affiché sur le bouton

Un clic sur le bouton -> Properties -> Text -> entrer "Browse"

ou bien un clic droit sur le bouton -> Edit Text

Pour modifier le nom de la variable jButton1 voir plus loin

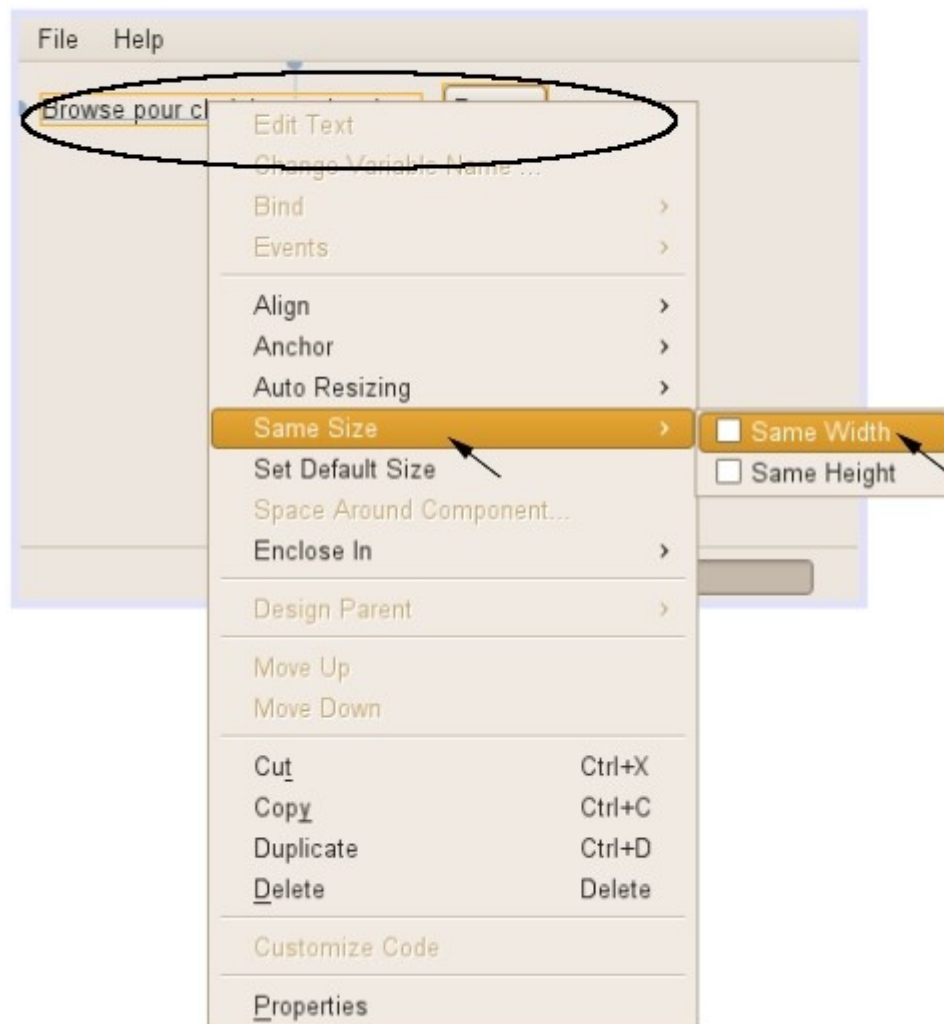
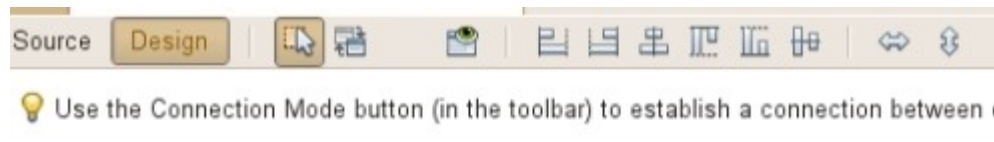


17. Donner aux deux composants la même largeur

Il est possible bien sûr de tirer sur les carrés pour modifier les dimensions du bouton
Mais nous allons allonger le bouton pour qu'il prenne la même largeur que JLabel1

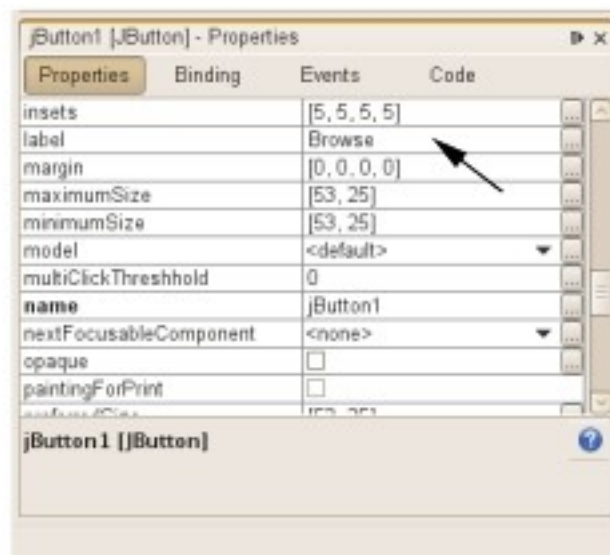
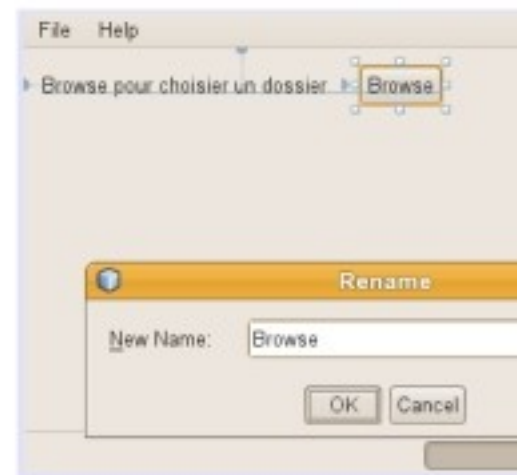
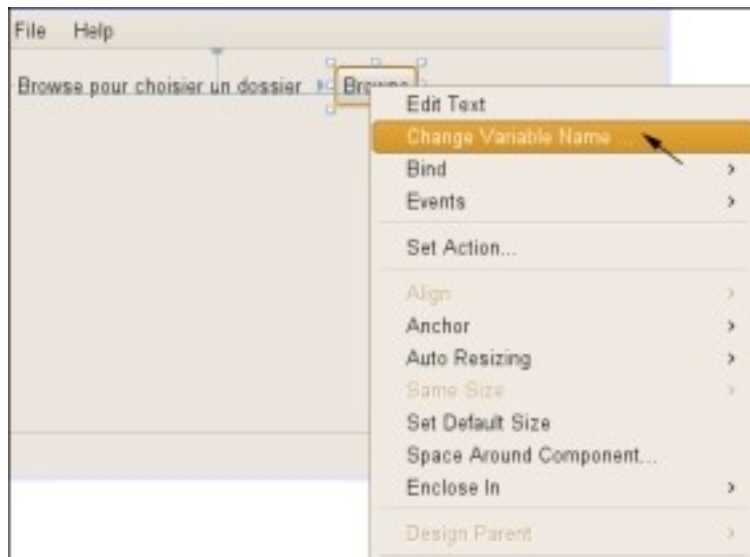
Un clic sur JLabel1 et CTRL+un clic sur le bouton :

Les deux étant sélectionnés un clic droit -> Same Size -> Same Width



18. Modifier le nom de jButton1

Un clic droit sur le bouton -> Edit Variable Name -> dans la fenêtre Rename -> entrer Browse ou bien un clic sur le bouton -> dans la fenêtre de propriétés -> sélectionner "Label" et entrer dans le champ valeur "Browse"

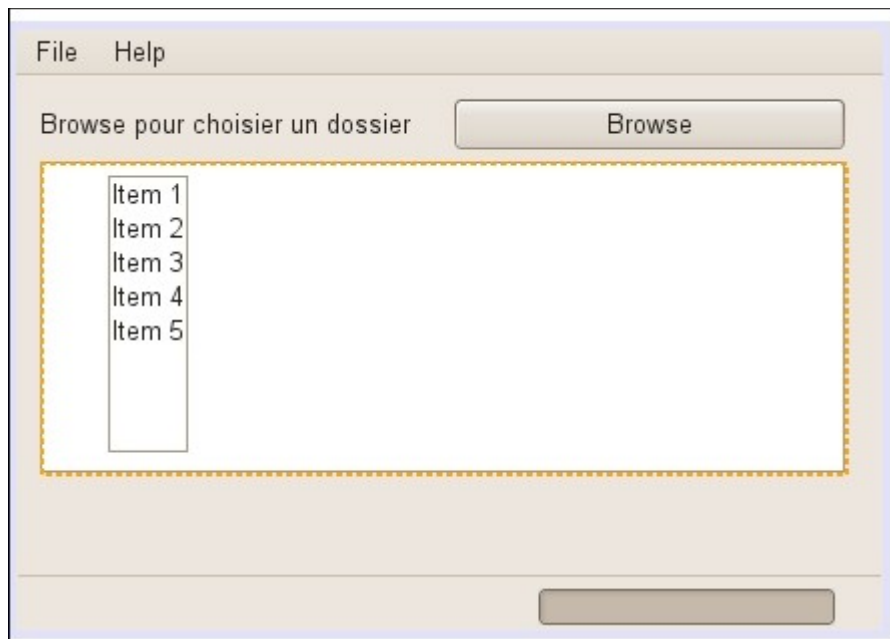


19. Ajouter une liste

Nous allons ajouter une liste sur l'espace de travail. Ajouter une liste ne pose aucun problème mais avant de le faire nous allons ajouter un composant JScrollPane. Ce composant va ajouter des ascenseurs si le contenu de la liste dépasse sa capacité d'affichage

Un clic sur Scroll Pane dans la palette et un autre pour le positionnement sur l'espace de travail. Les règles sont affichées et le changement de dimensions du composant se fait en tirant sur les petits carrés.

Ensuite un clic sur List dans la palette pour ajouter un composant JList. Un autre clic à l'intérieur de "Scroll Pane".



20. Alignement des composants

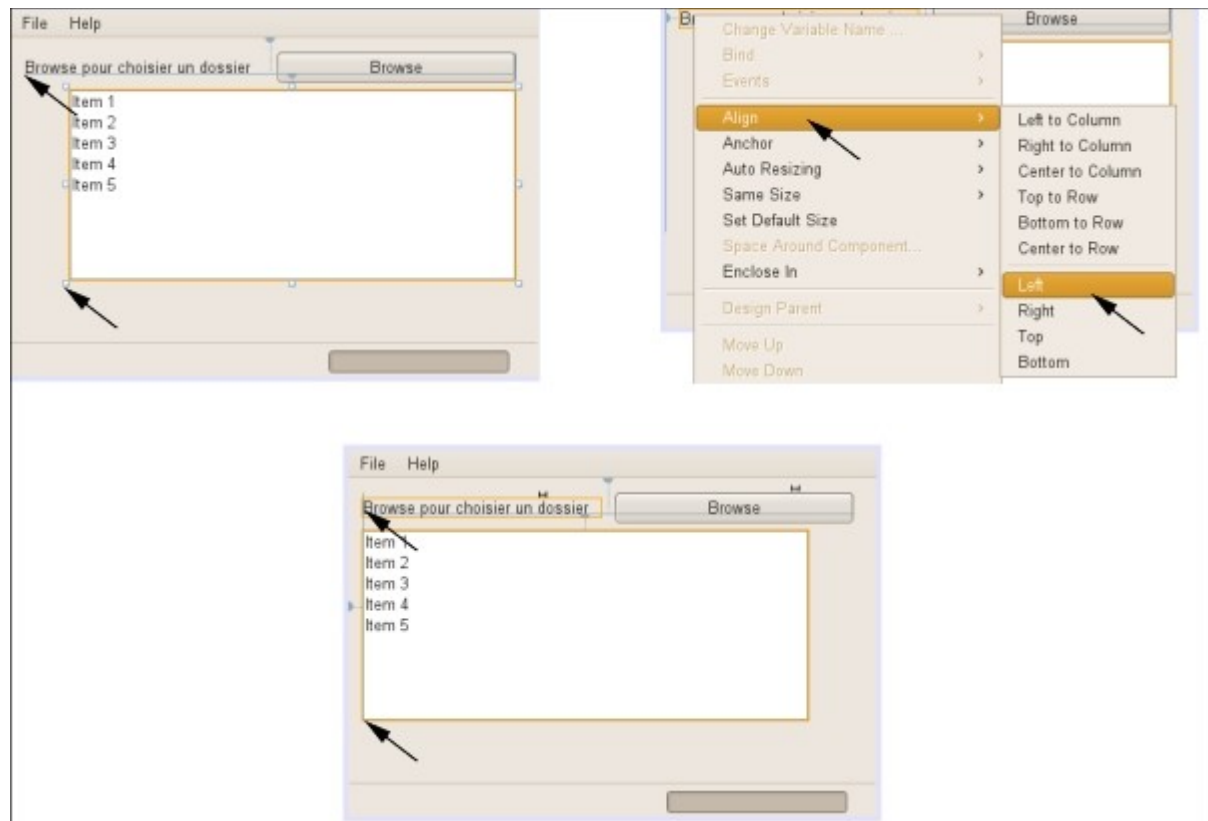
Nous allons aligner JLabel1 et JList1

Un clic sur JLabel1 -> CTRL+clic sur JList1

Les deux sont sélectionnés -> clic droit -> align -> Left

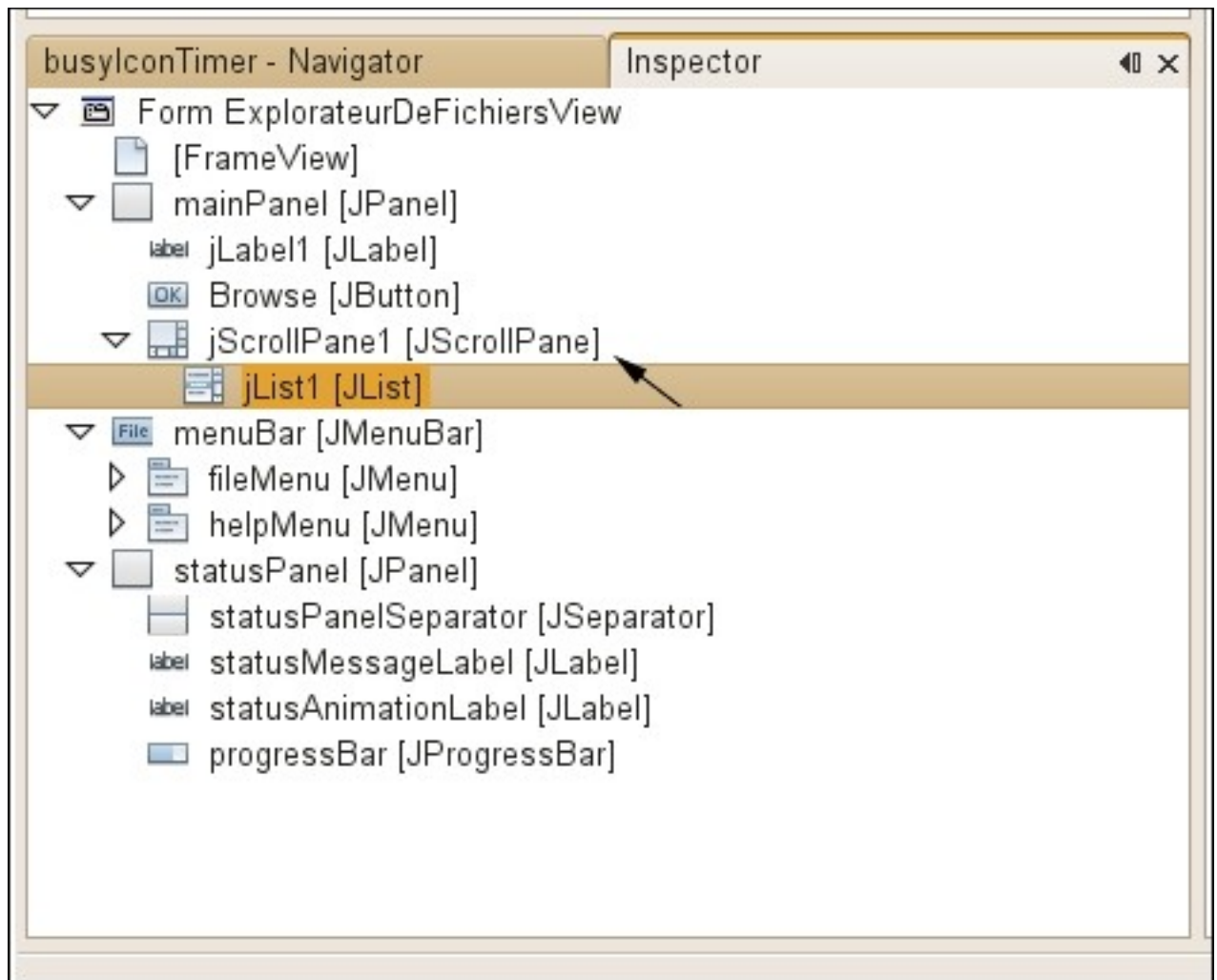
Les deux composants sont maintenant alignés à gauche

Tirer sur les carrés pour élargir la liste



21. L'inspector

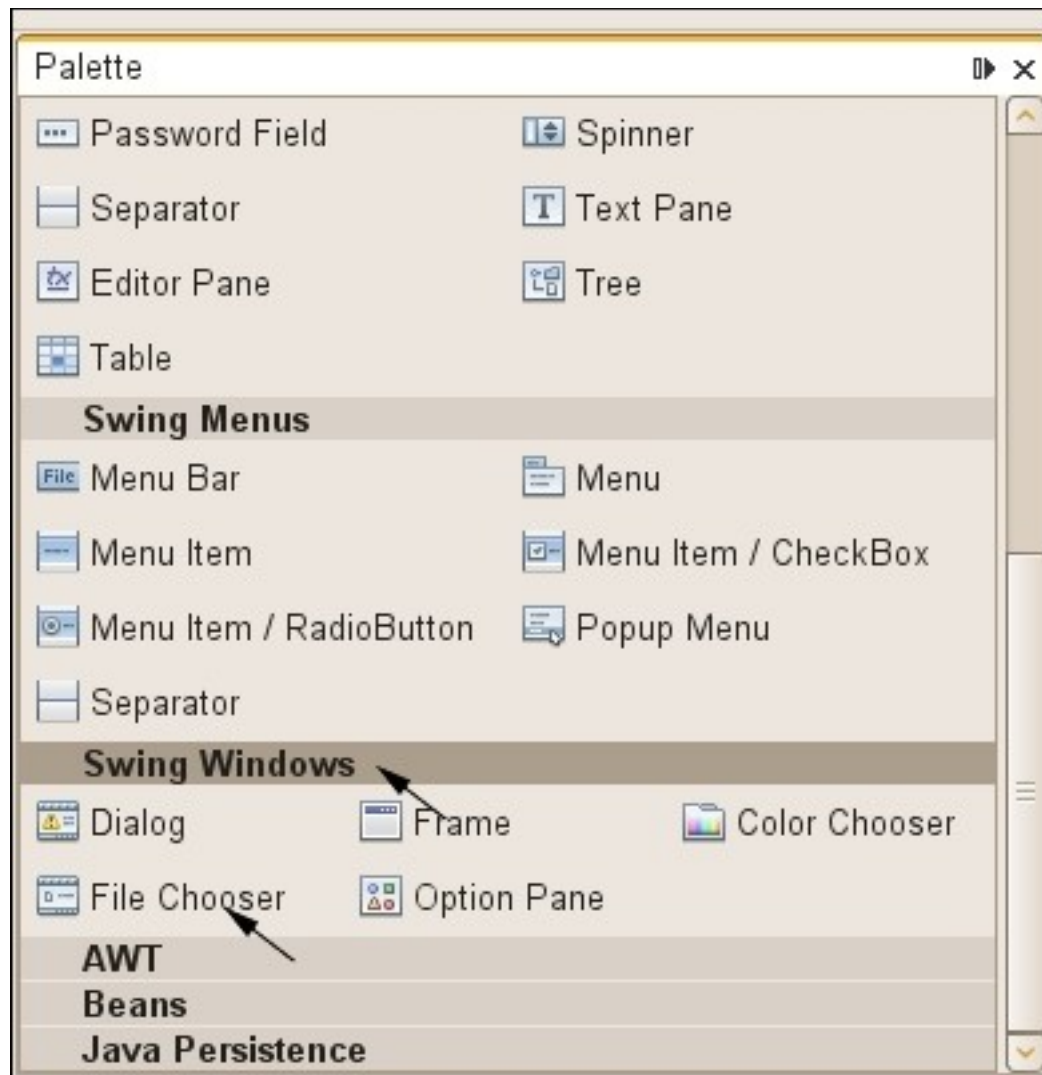
L'inspecteur de NetBeans permet de voir les composants de notre application visuel ou non
 JScrollPane est un composant non-visuel



22. Ajouter un composant File Chooser

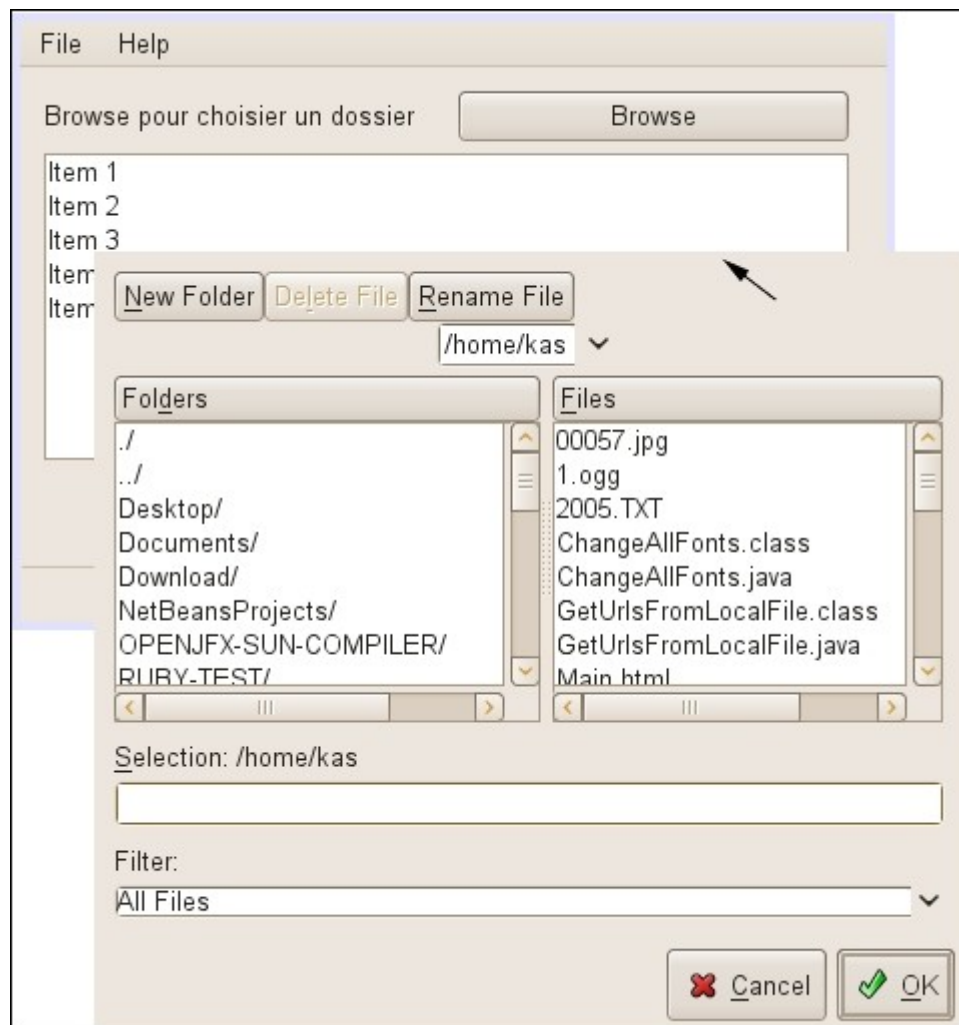
JFileChooser est un composant non-visuel permettant d'ouvrir un explorateur de fichiers pour sélectionner des fichiers et des dossiers sur le système d'exploitation.

Un clic sur File Chooser -> (dans la partie Swing Windows de la palette) -> un autre clic à n'importe quel endroit sur l'espace de travail, si on souhaite le positionner sur l'espace de travail à côté des autres composants ou bien en dehors de l'espace de travail. Il va s'ouvrir sous forme de Pop-Up.



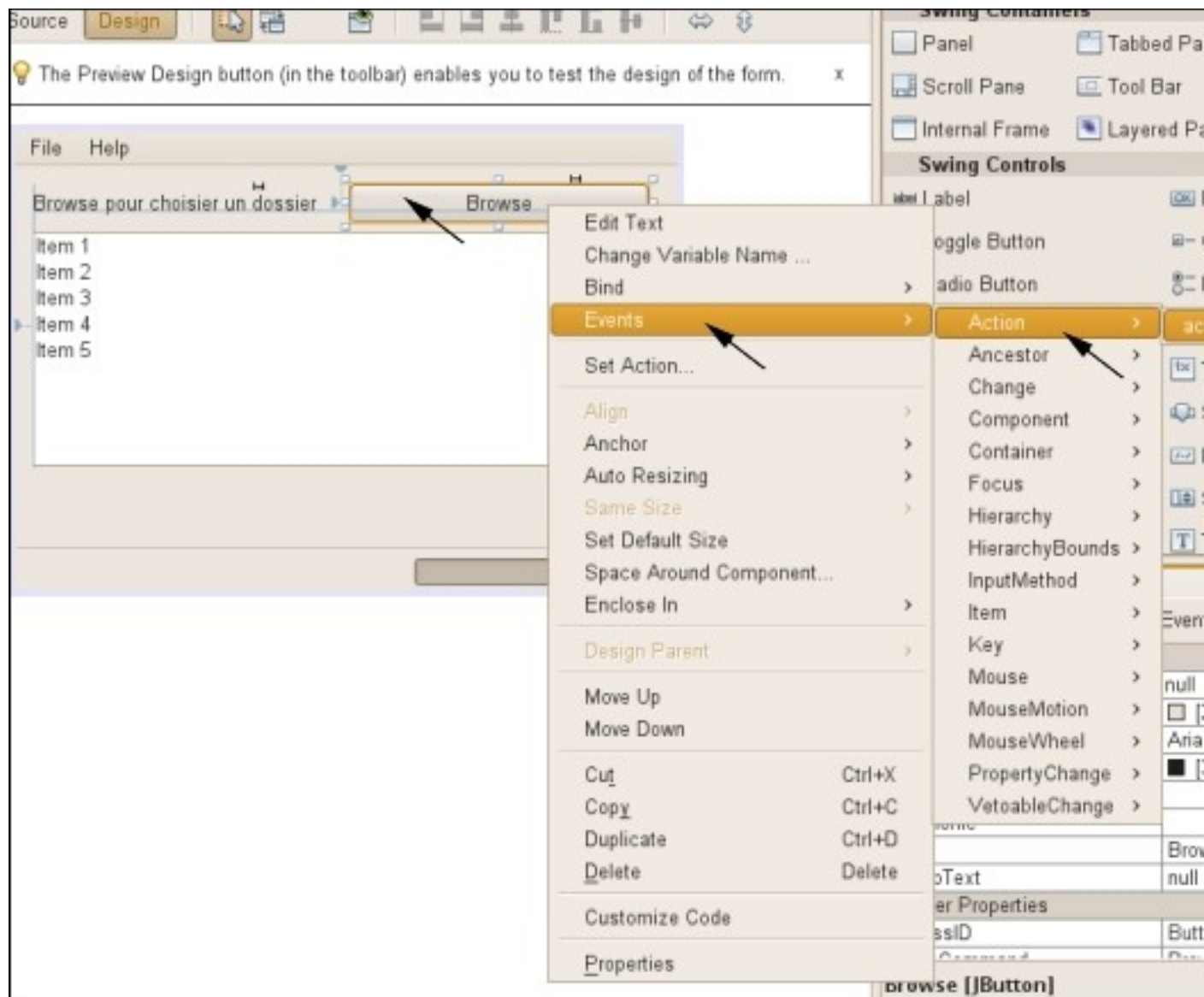
23. File Chooser

Suite



24. Associer une action au bouton

Nous allons ajouter un événement associé au bouton -> un clic droit sur le bouton "Browse"
-> Events -> Action -> actionPerformed



25. L'éditeur Source

L'éditeur du fichier source s'ouvre automatiquement lors du clic sur actionPerformed permettant d'ajouter du code pour personnaliser l'action

```

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(statusMessageLabel)
.addComponent(statusAnimationLabel)
.addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.D
.addGap(3, 3, 3))

);

setComponent(mainPanel);
setMenuBar(menuBar);
setStatusBar(statusPanel);
} // </editor-fold>

private void BrowseActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}

// Variables declaration - do not modify
private javax.swing.JButton Browse;
private javax.swing.JMenuItem exitMenuItem;
private javax.swing.JMenu fileMenu;
private javax.swing.JLabel jLabel1;
private javax.swing.JList jList1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JPanel mainPanel;
private javax.swing.JMenuBar menuBar;
private javax.swing.JProgressBar progressBar;
private javax.swing.JLabel statusAnimationLabel;
private javax.swing.JLabel statusMessageLabel;
private javax.swing.JPanel statusPanel;
// End of variables declaration

private final Timer messageTimer;
private final Timer busyIconTimer;
private final Icon idleIcon;
private final Icon[] busyIcons = new Icon[15];
private int busyIconIndex = 0;

private JDialog aboutBox;
}

```

26. Quelques lignes de code

Ajoutons les lignes suivantes pour faire apparaître `JFileChooser1` quand on clic sur le bouton "Browse" et retourner dans "dirName" le dossier sélectionné

```

jFileChooser1=new JFileChooser();

jFileChooser1.setSelectionMode(JFileChooser.DIRECTORIES_ONLY);

jFileChooser1.setMultiSelectionEnabled(false);

jFileChooser1.setDialogTitle("Select a Directory");

int result=jFileChooser1.showOpenDialog(null);

if(result==JFileChooser.APPROVE_OPTION )

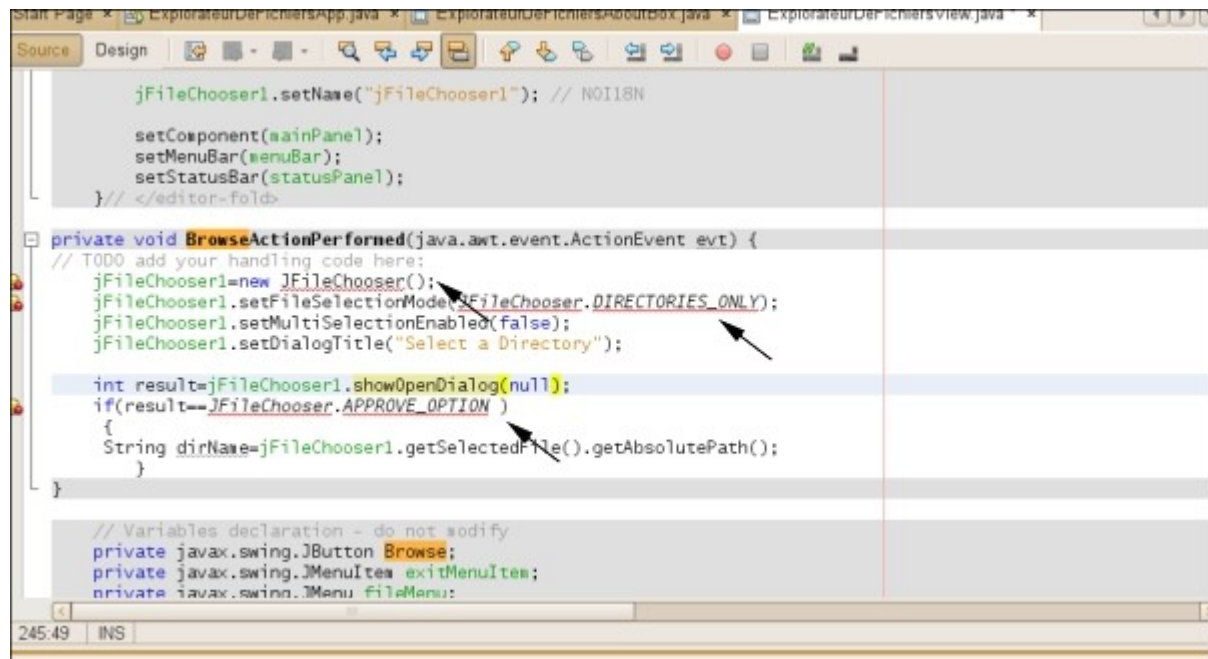
{

String dirName=jFileChooser1.getSelectedFile().getAbsolutePath();

}

```

Des messages d'erreurs dues au problème d'import !!!



```
Start Page * ExplorerDeFichiersApp.java * ExplorerDeFichiersAboutBox.java * ExplorerDeFichiersView.java *
Source Design
jFileChooser1.setName("jFileChooser1"); // NOI18N
setComponent(mainPanel);
setMenuBar(menuBar);
setStatusbar(statusPanel);
} // </editor-fold>

private void BrowseActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
jFileChooser1=new JFileChooser();
jFileChooser1.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
jFileChooser1.setMultiSelectionEnabled(false);
jFileChooser1.setDialogTitle("Select a Directory");

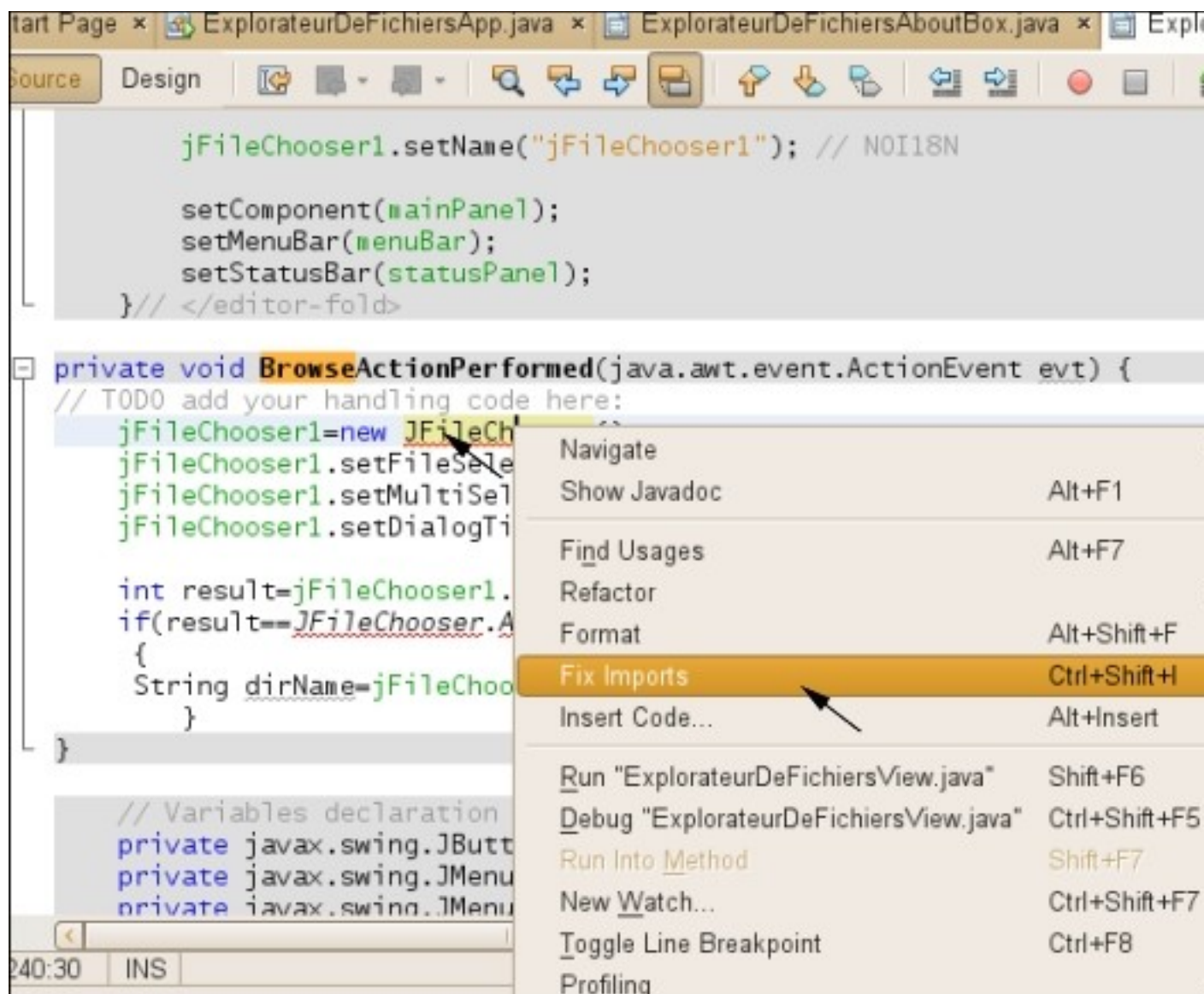
int result=jFileChooser1.showOpenDialog(null);
if(result==JFileChooser.APPROVE_OPTION)
{
String dirName=jFileChooser1.getSelectedFile().getAbsolutePath();
}

// Variables declaration - do not modify
private javax.swing.JButton Browse;
private javax.swing.JMenuItem exitMenuItem;
private javax.swing.JMenu fileMenu;
```

245.49 INS

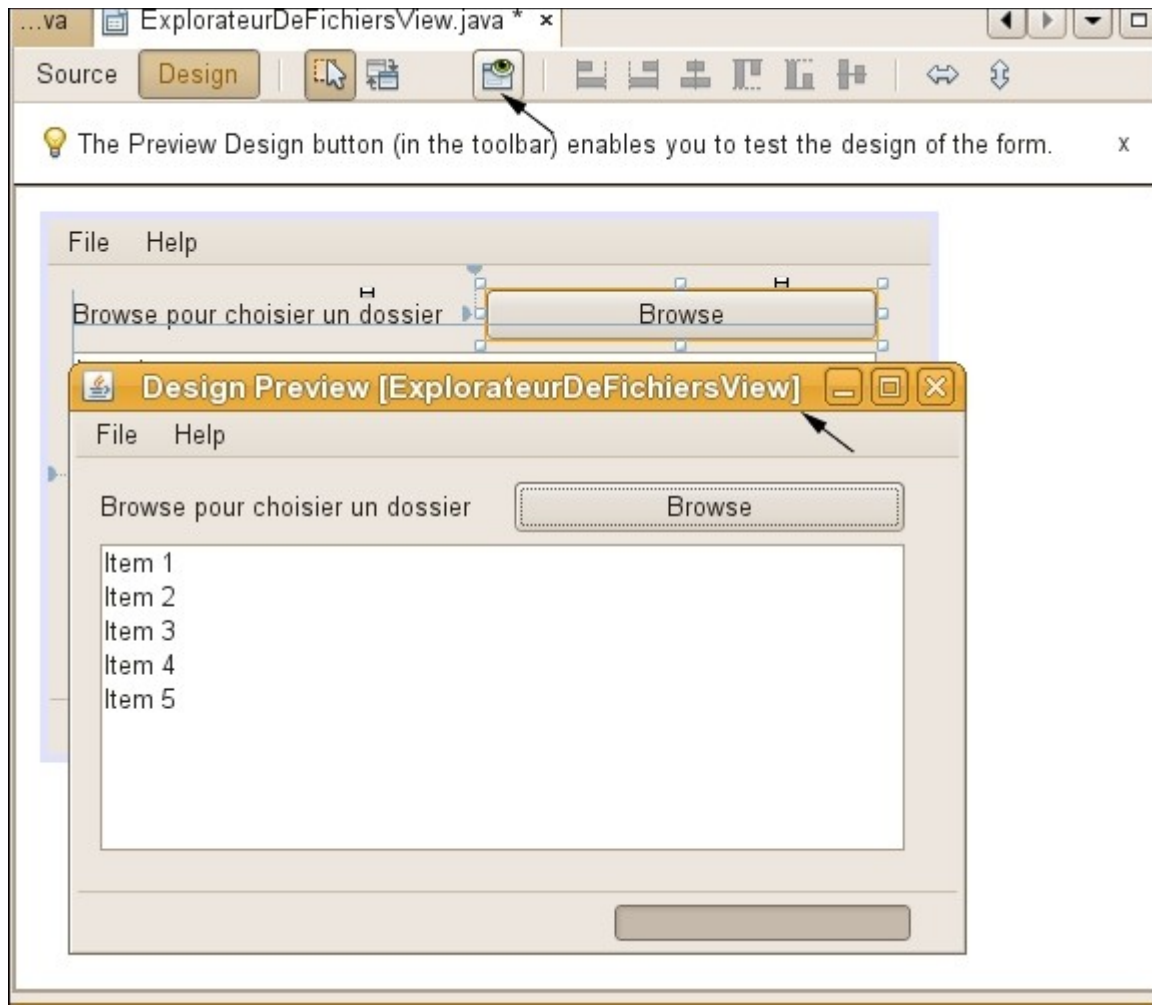
27. Solutionner le problème d'import

Un clic droit sur JFileChooser -> Fix Imports (CTRL+SHIFT+I) et le problème est réglé



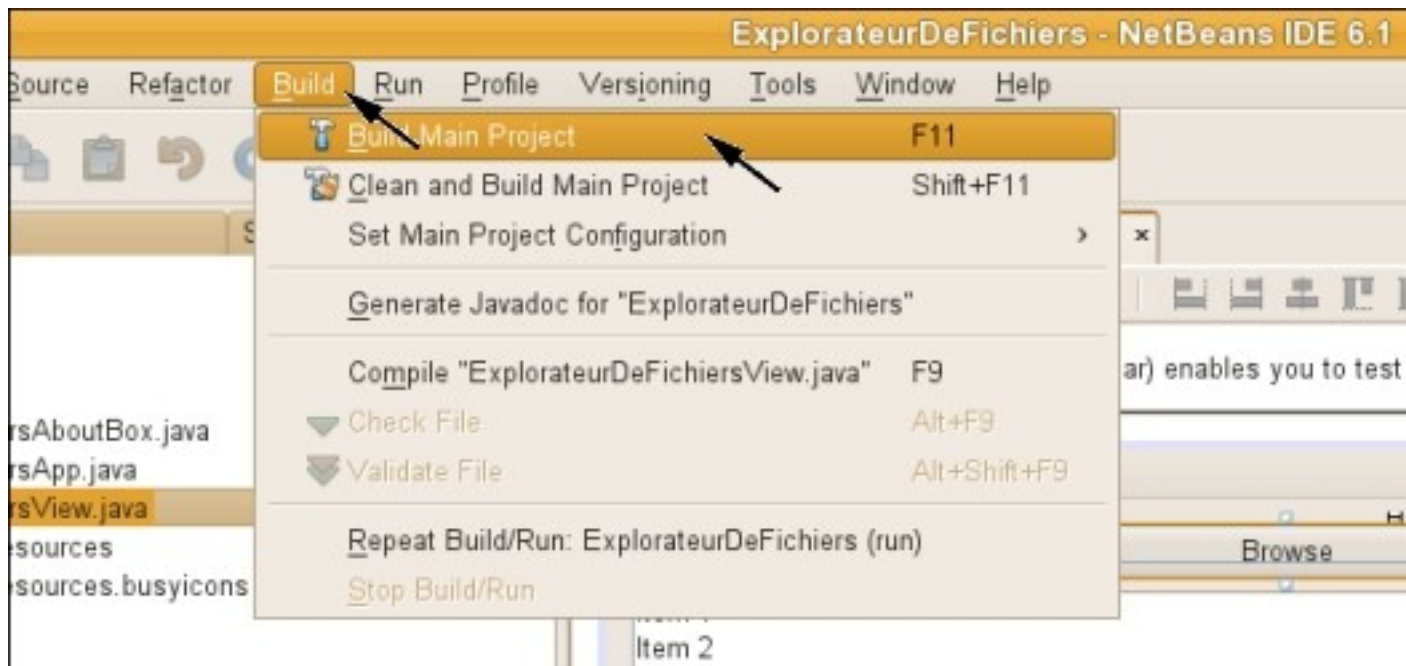
28. L'aperçu du designer :

Un clic sur le bouton Preview Desing -> à partir de la fenêtre de design (l'inverse de la fenêtre de l'édition du fichier source) -> ouvre un aperçu de l'application



29. Compiler notre projet

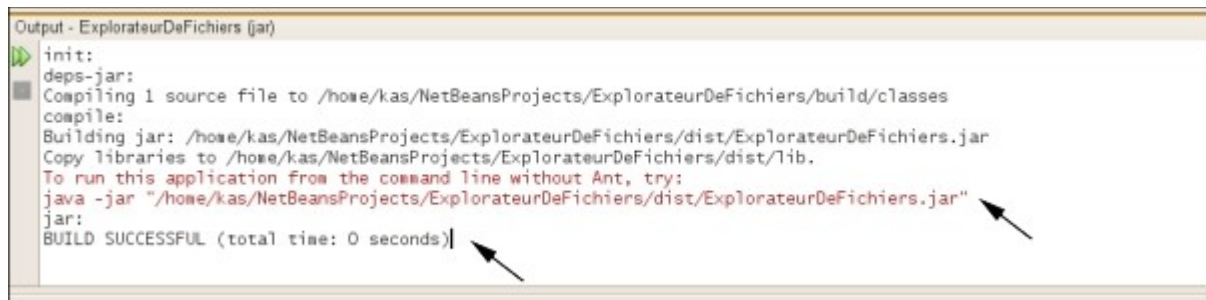
Un clic sur -> le menu Build > Build Main Project (ou F11)



30. Une compilation réussite :

Un message indiquant la fin réussite de la compilation est affiché et une indication pour exécuter l'application en dehors de l'IDE

En cas d'erreurs, un clic sur une erreur elle identifiée dans l'éditeur



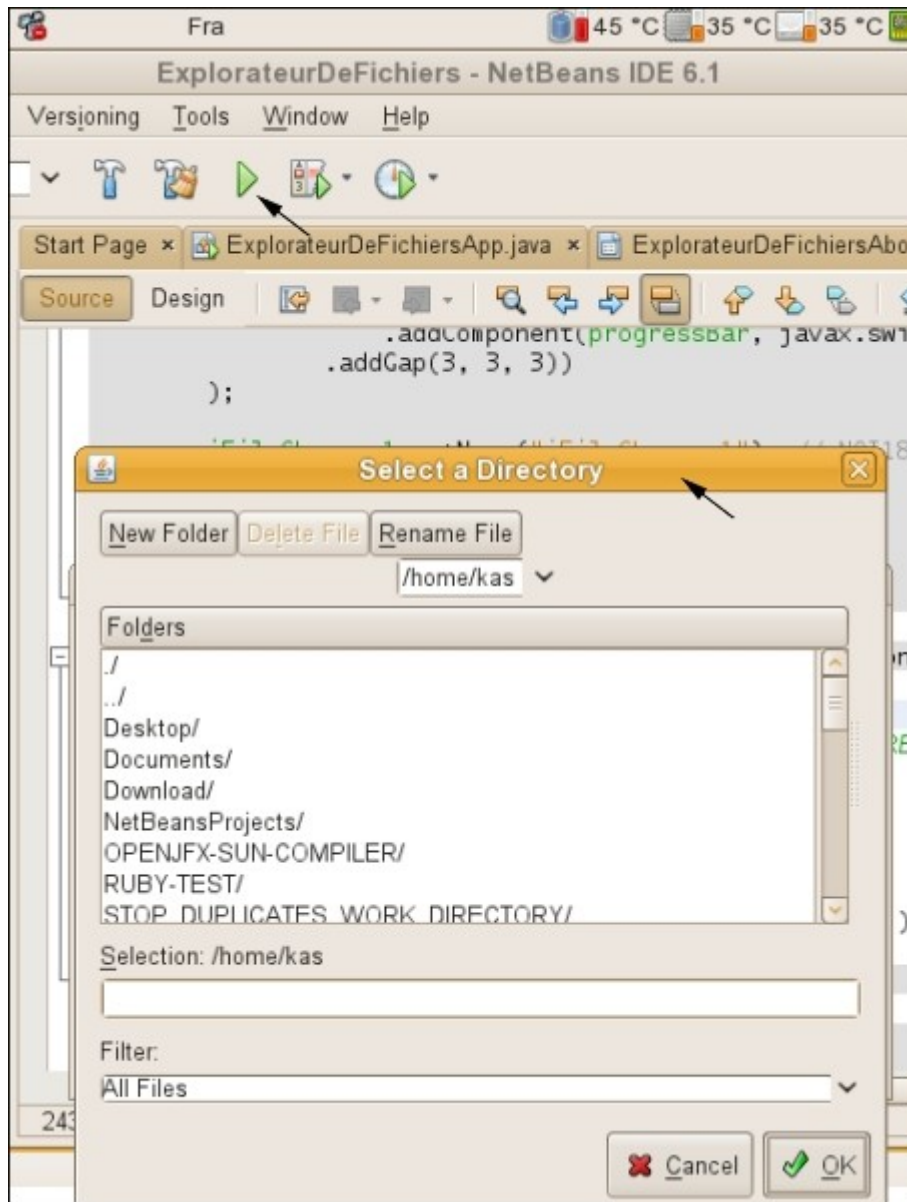
```
Output - ExplateurDeFichiers (jar)
init:
deps-jar:
Compiling 1 source file to /home/kas/NetBeansProjects/ExplateurDeFichiers/build/classes
compile:
Building jar: /home/kas/NetBeansProjects/ExplateurDeFichiers/dist/ExplateurDeFichiers.jar
Copy libraries to /home/kas/NetBeansProjects/ExplateurDeFichiers/dist/lib.
To run this application from the command line without Ant, try:
java -jar "/home/kas/NetBeansProjects/ExplateurDeFichiers/dist/ExplateurDeFichiers.jar"
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```

31. Tester l'application :

Ou l'exécuter à l'intérieur de l'IDE :

Un clic sur le menu Run > Run Main Project

ou l'icône "Play" ou F6



32. Editer jList1

Il est possible d'éditer graphiquement les JLists graphiquement pour modifier les éléments affichés.

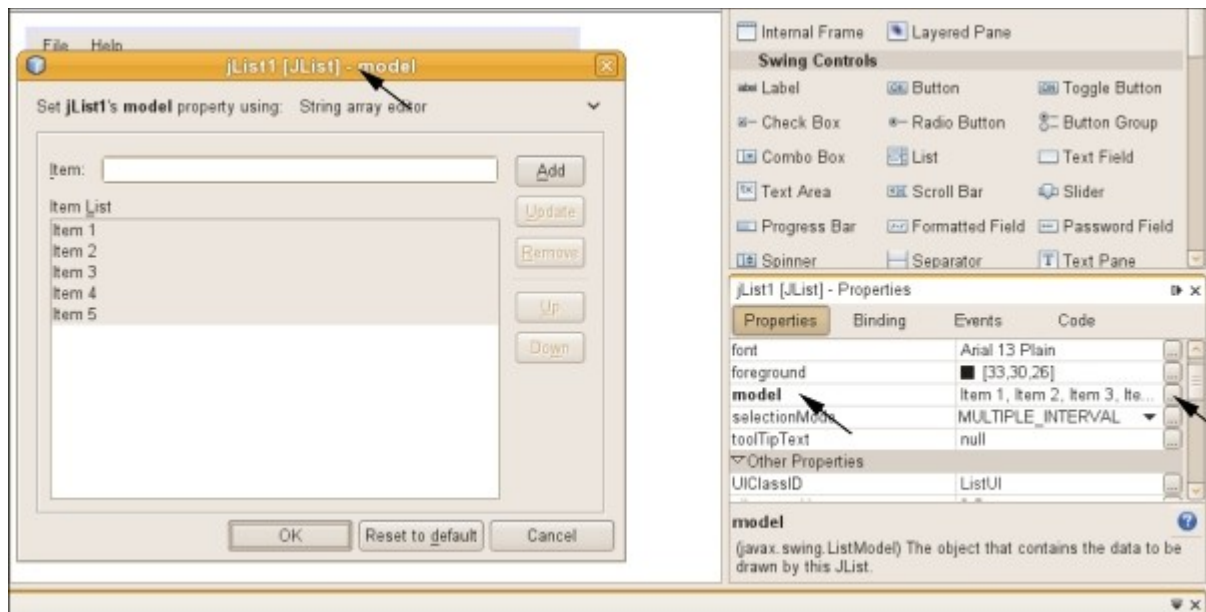
Ici par exemple nous allons vider jList1 de son contenu.

Un clic sur jList1 -> Propriétés -> model -> un clic sur le bouton avec trois point (...)

Une fenêtre de dialogue s'ouvre

Un clic sur "Item 1" -> un clic sur "Remove"

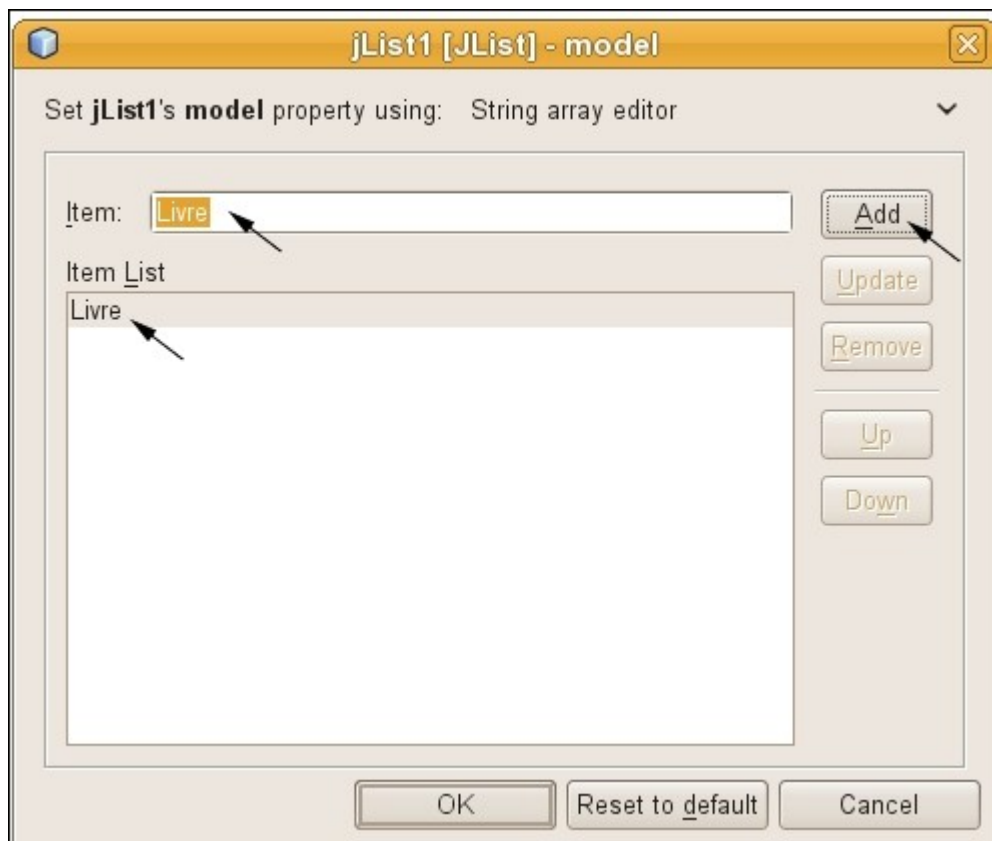
Répéter l'opération pour les Item2 à l'Item5



33. JList suite :

Note :

Pour ajouter des éléments dans jList1 -> entrer un article dans "Item" -> puis un clic sur Add



34. Remplir la liste avec des fichiers

Ajoutons ces quelques lignes de code à la suite des précédentes

L'assistant de NetBeans est là pour aider à la saisie

```

//Modifier Model pour jList1

DefaultListModel jListFilesModel = new DefaultListModel();

jList1.setModel(jListFilesModel);

//Remplir la list avec les fichiers dans le dossier sélectionné avec File
Chooser

File fDir=new File(dirName);

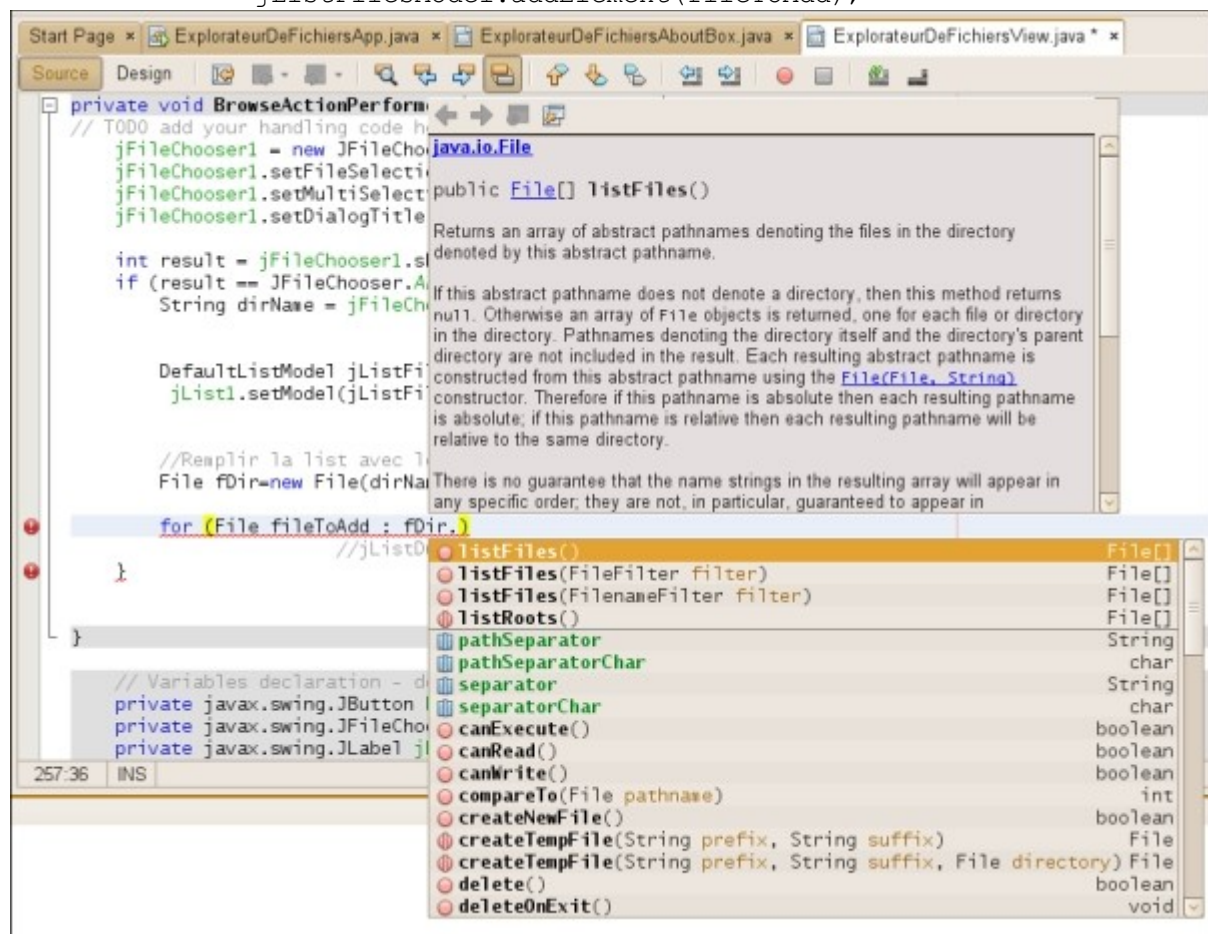
//Remplir la liste avec uniquement des fichiers

for (File fileToAdd : fDir.listFiles())

    if(fileToAdd.isFile())

        jListFilesModel.addElement(fileToAdd);

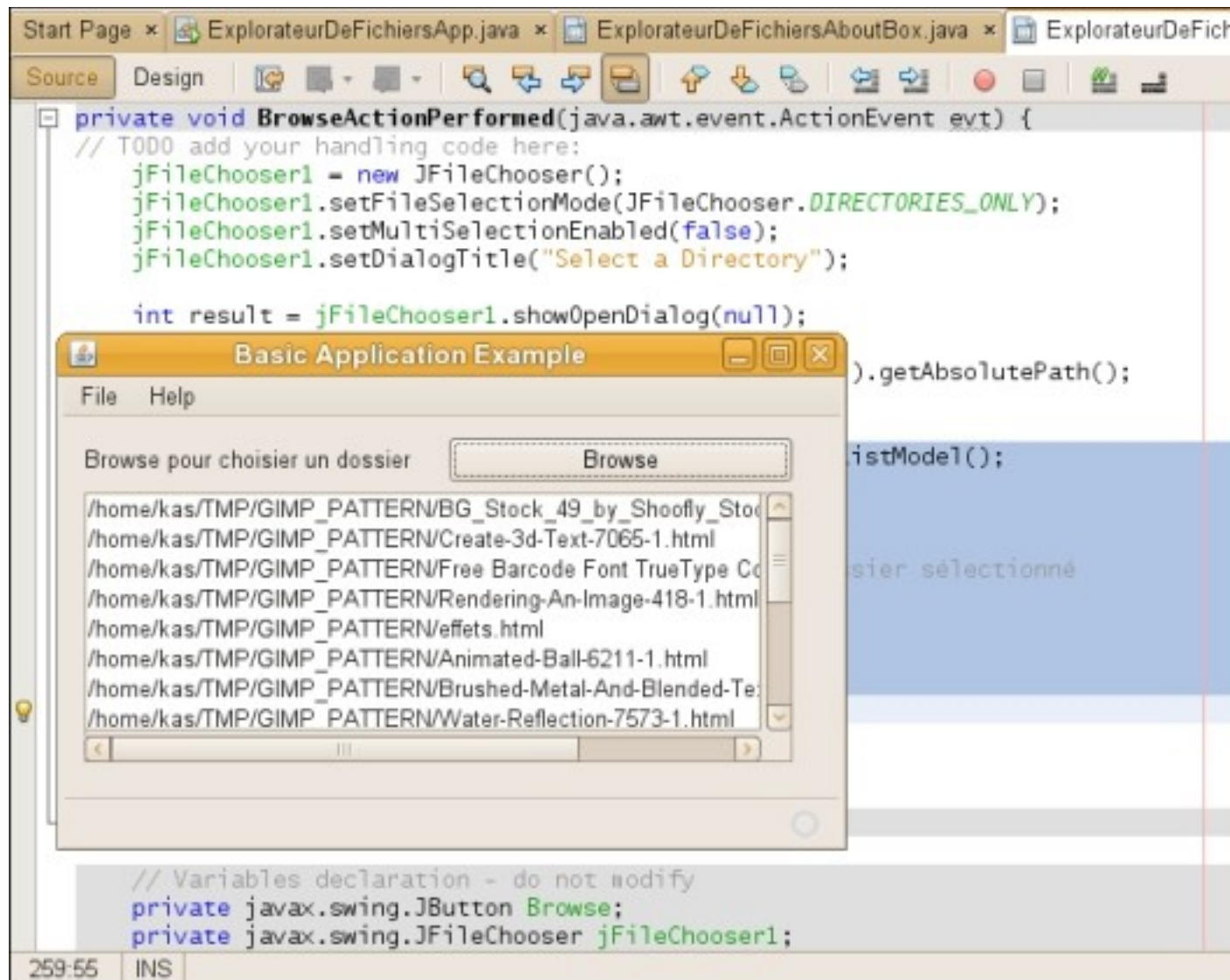
```



35. Tester l'application :

Un clic sur le menu Run -> Run Main Project va procéder à la compilation et l'exécution de notre application.

Un clic sur le bouton "Browse" ouvre l'explorateur de fichiers sur le système -> quand on clique sur un dossier -> jList1 est remplie avec des fichiers du dossier sélectionné

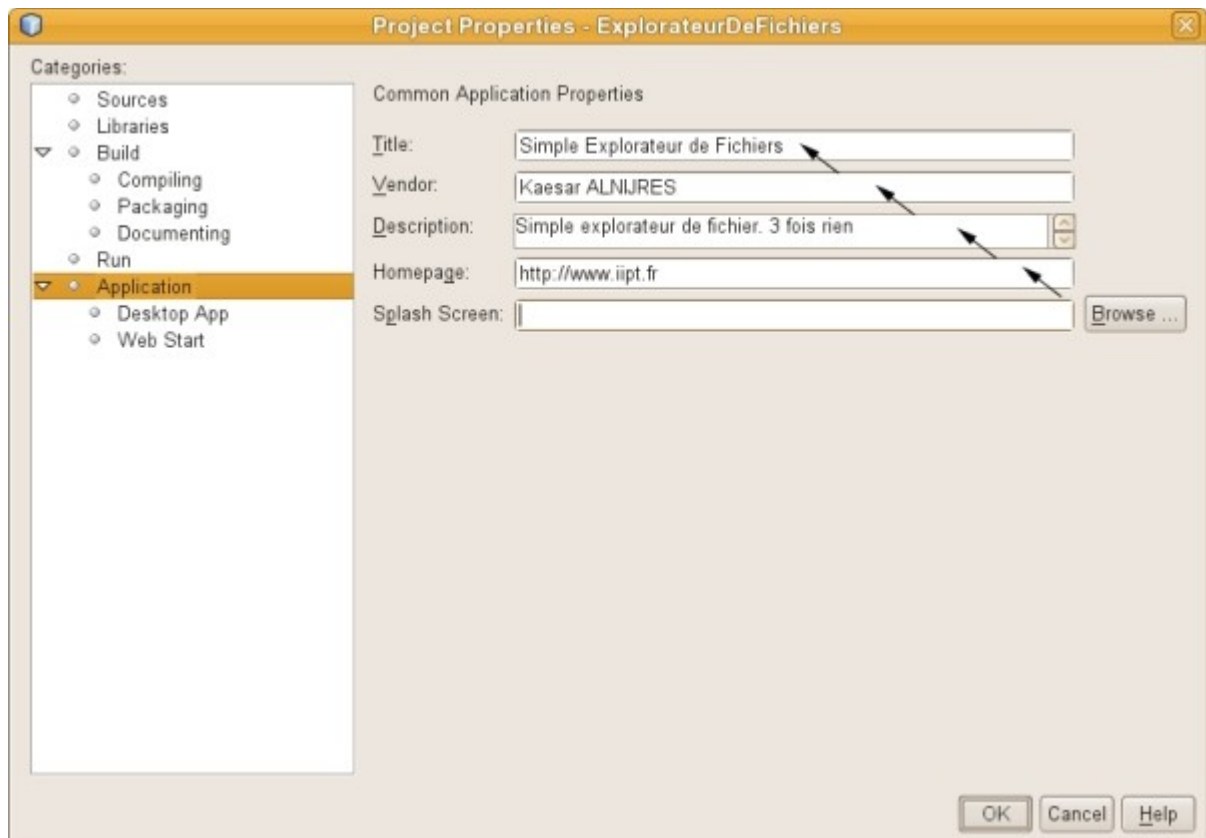


36. Les touches de la fin

Sélectionner le projet -> clic droit -> Properties (propriétés) -> Application

A personnaliser le titre de l'application, son auteur ou vendeur, description, et HomePage.

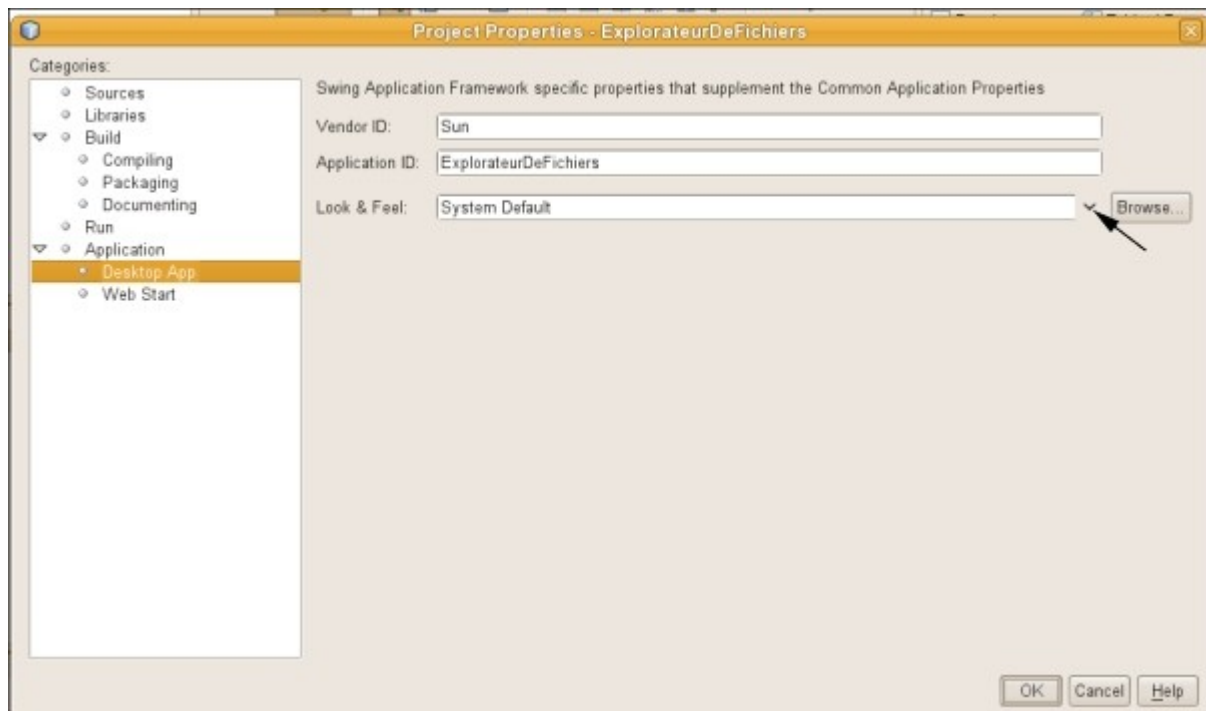
Nous reviendrons sur le SplashScreen plus loin



37. Sélectionner le Look & Feel

Toujours dans propriétés -> un clic sur Desktop App

Sélectionner dans Look & Feel -> Java Default

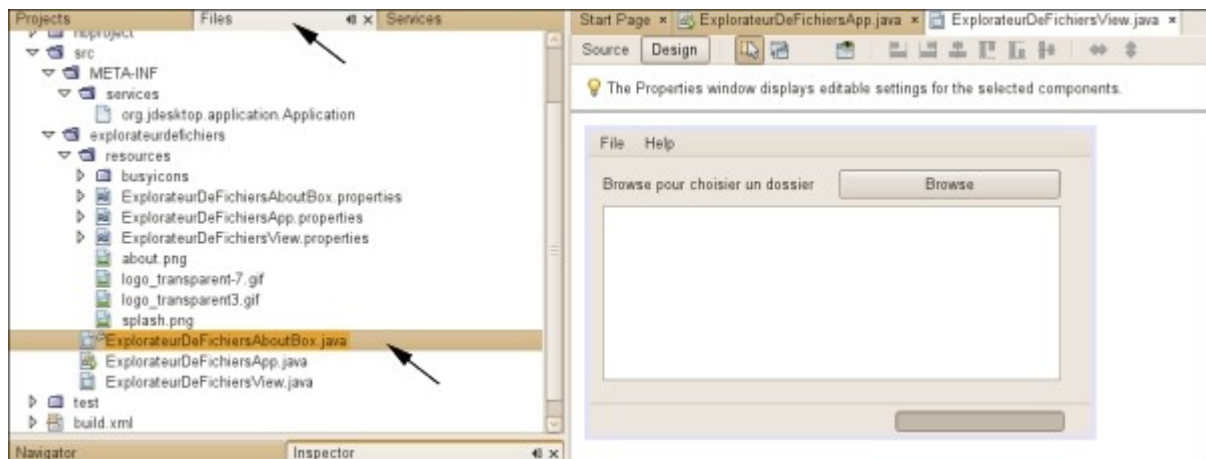


38. About Dialog

Avec Java Desktop Application il est possible de personnaliser le menu "About" sous le menu "Help"

Un double clic dans l'explorateur de fichier de NetBeans sur "ExplorateurDeFichierAboutBox.java" ouvre le fichier dans le "Designer" (Files -> ExplorateurDeFichiers->src -> explorateurdefichiers-> ExplorateurDeFichierAboutBox.java)

On peut modifier le design à volonté



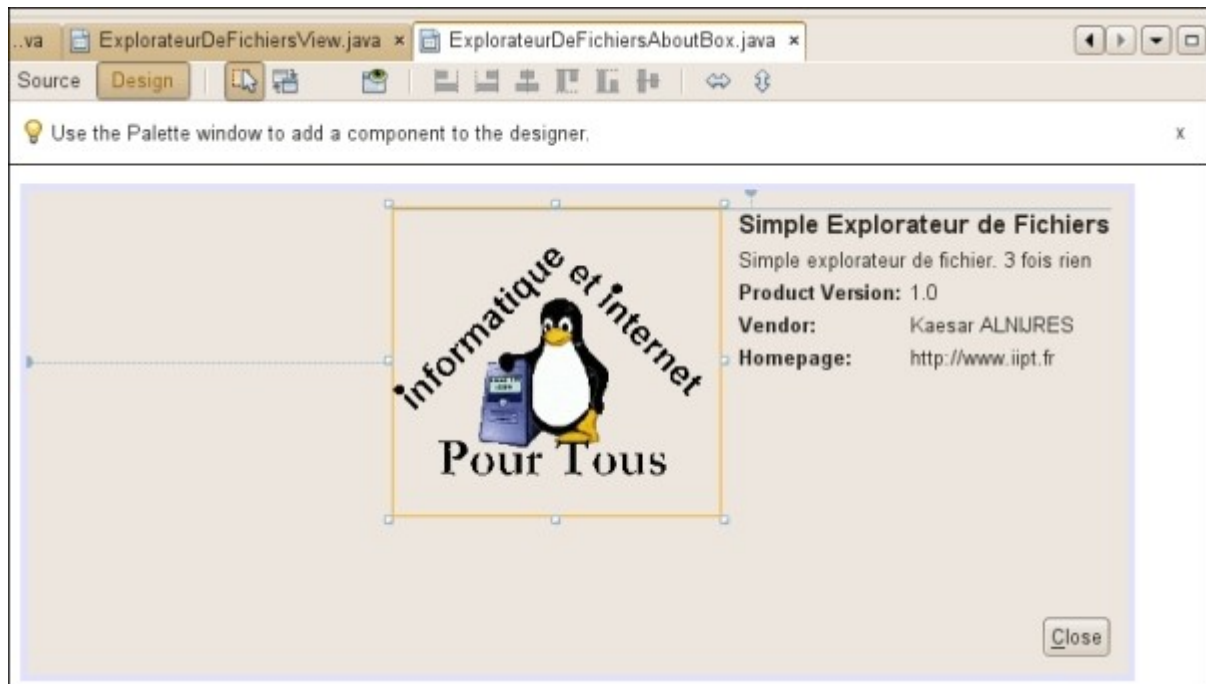
39. La fenêtre de design de About

La fenêtre d'About s'ouvre dans le designer permettant des modifications à volonté, l'image les textes (JLabel), ajouter des composants à partir de la palette etc.

A titre d'exemple un clic sur l'image -> propriétés -> Icon -> ouvre une fenêtre pour sélectionner une image dans le projet, importer une image et l'utiliser comme icône etc.

Note :

Plus de détails concernant l'importation des images plus loin



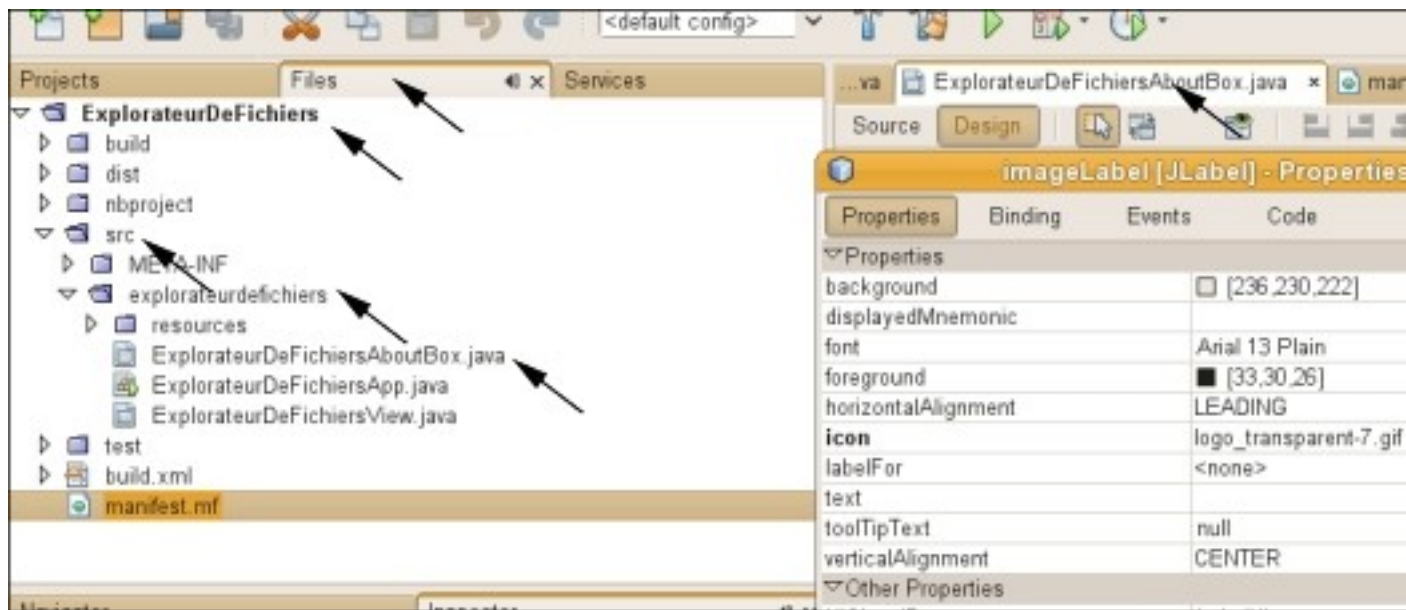
40. Le Splash Screen :

Selon les experts de NetBeans un bug s'est glissé dans le template, l'empêchant de modifier le fichier manifest.mf. Nous allons donc utiliser une des nouveautés de Java SE 6 pour afficher le Splash Screen dans l'application finale

On commence par copier une image .gif dans le dossier resource du projet NetBeans soit par une opération de copie directe ou en ouvrant le fichier "ExplorateurDeFichiersAboutBox.java" en double cliquant sur son nom dans le l'onglet Files -> ExplorateurDeFichiers->src -> explorateurdefichiers.

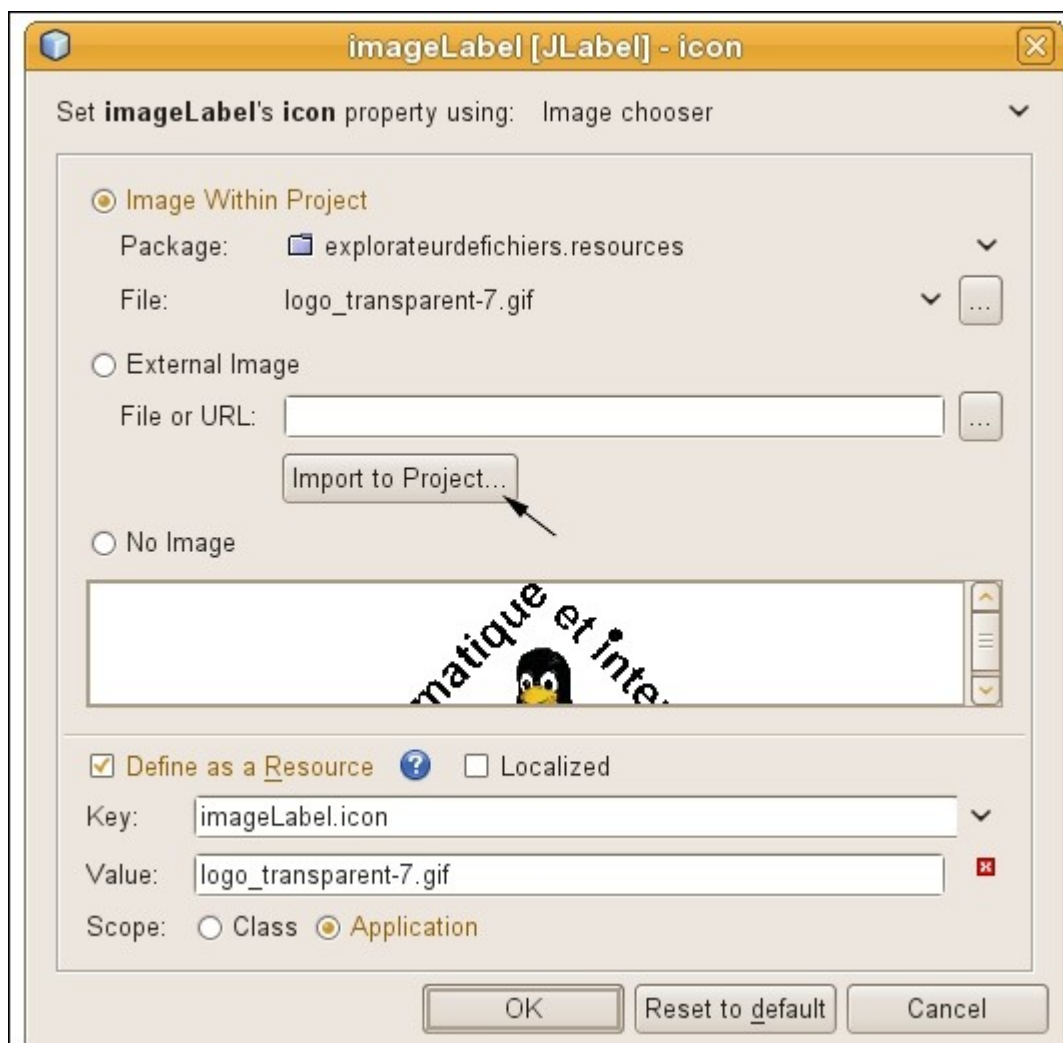
Un clic droit sur l'image de "About" -> Properties -> ouvre une boîte de dialogue.

Sur la ligne icon un clic sur le bouton pourtant trois point (...) ouvre un wizard permettant de trouver des images sur le disque dur et de les importer dans le projet



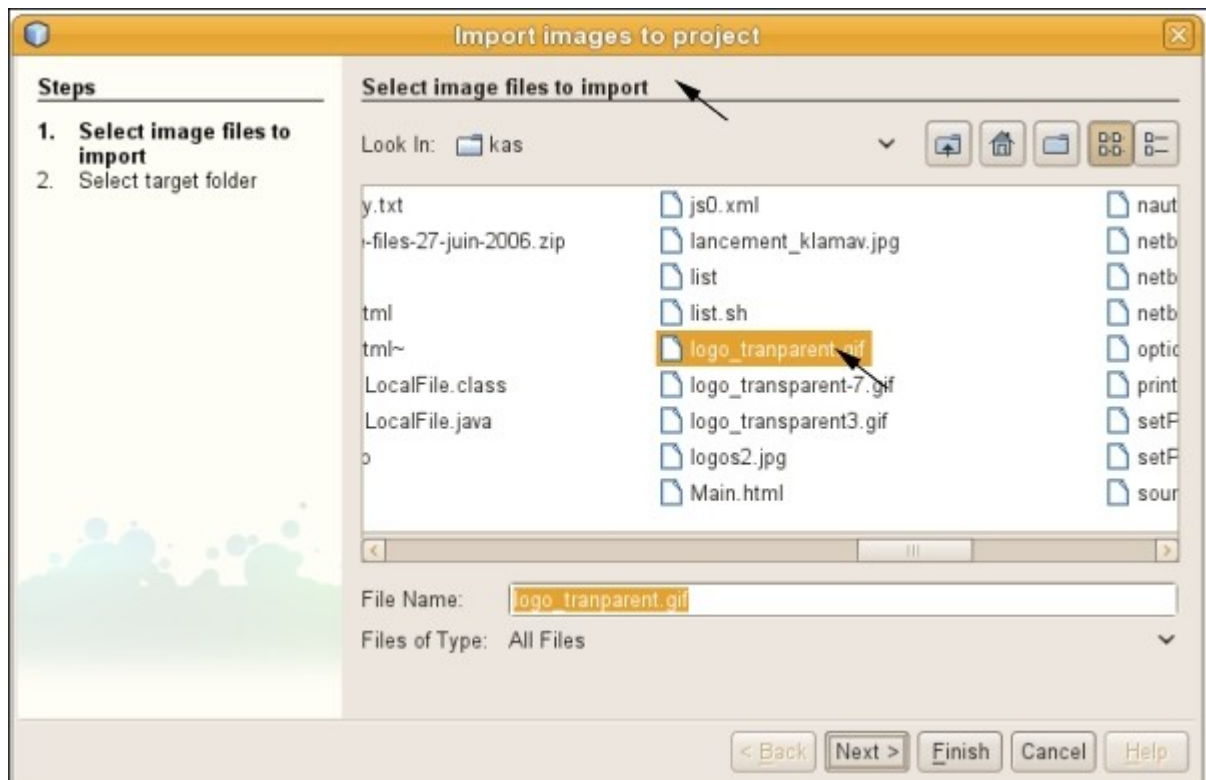
41. Une image pour le Splash Screen

Un clic sur "Import to Project"



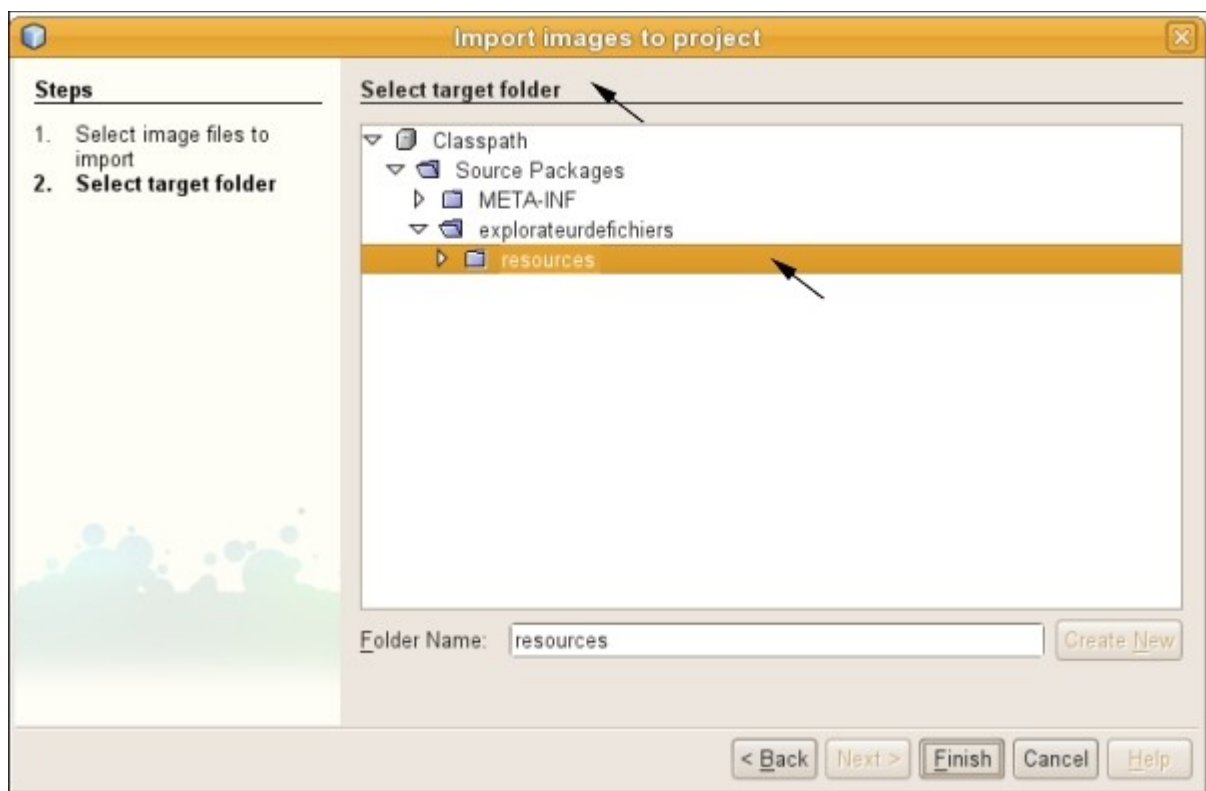
42. Importer une image

Le navigateur de fichiers s'ouvre pour sélectionner une image sur le disque dur



43. Sélectionner le dossier de destination dans le projet

Le dossier de destination doit être "resources" dans le projet



44. Image de Splash Screen suite

L'image est maintenant dans le projet dans le dossier ressources

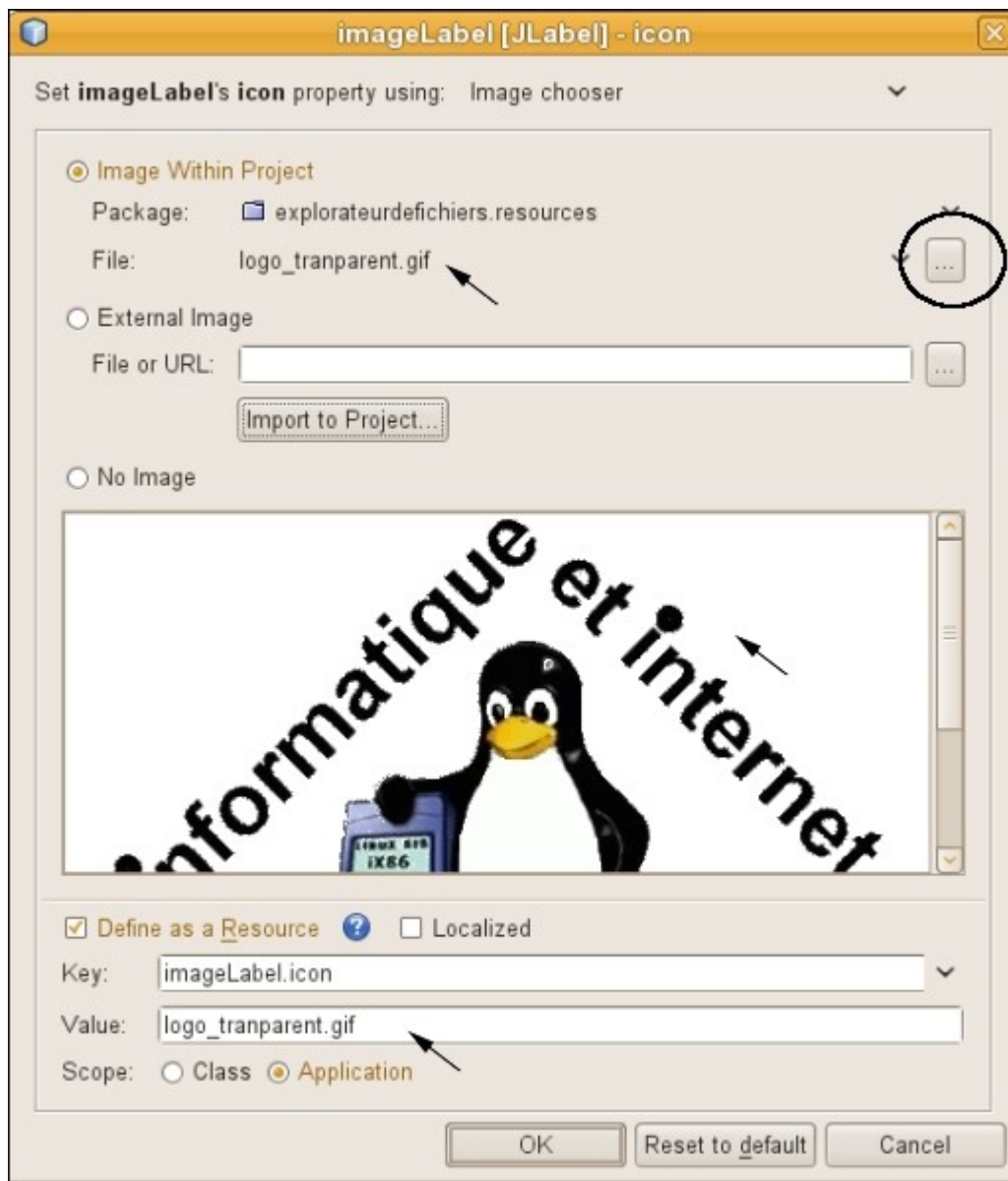
Le nom de l'image est affiché dans aperçu

Sa visibilité dans "Scope" est Application

Maintenant l'image est importée, redonnons à About son image -> un clic sur le bouton avec 3 points (...) -> sélectionner dans "resources" l'image choisie précédemment pour About

Un clic sur "OK"

Un clic sur "close" pour fermer les propriétés



45. Modifier manifest.mf

Un clic sur le navigateur de fichiers dans NetBeans -> double clic sur manifest.mf

Ajouter la ligne suivante dans le fichier :

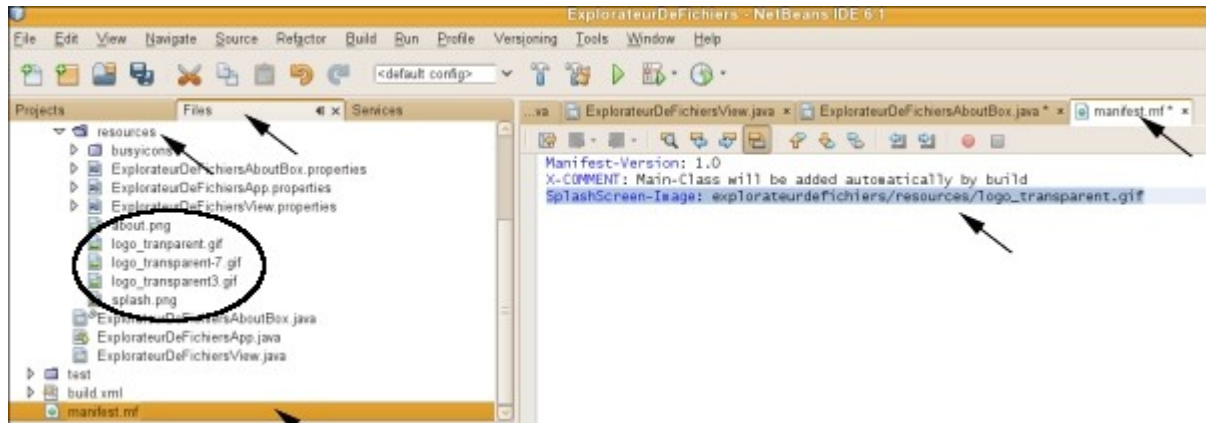
SplashScreen-Image: explorateurdefichiers/resources/logo_transparent.gif

Note :

Le nom de l'image de l'image est choisi par moi et reste bien sûr à personnaliser

Note :

Les images disponibles pour le Splash Screen et la boîte "About" sont dans le dossier "resources" indiqué sur l'image



46. Exécuter l'application en dehors de l'IDE

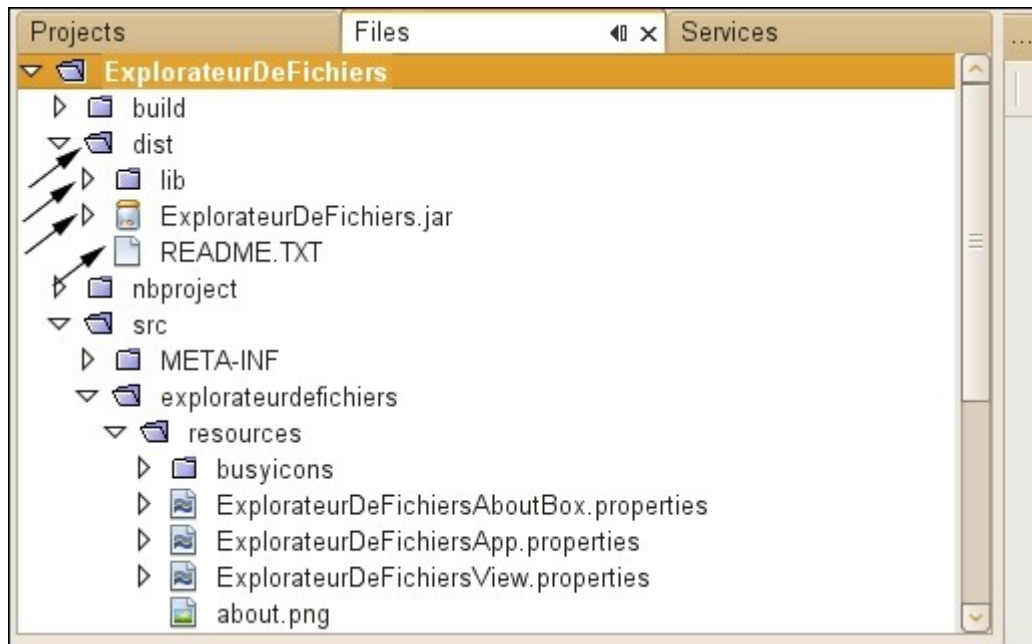
Un clic sur le menu Build -> Build Main Project

A la fin de l'opération le dossier "dist" va contenir :

1. Un dossier "lib" contenant les bibliothèques nécessaires à l'exécution du projet, spécialement sur Java SE 5
2. Le fichier .jar exécutable de projet
3. Un fichier README.TXT décrivant les procédures pour exécuter le .jar et comment préparer le projet pour le distribuer et le déploiement sur un autre système

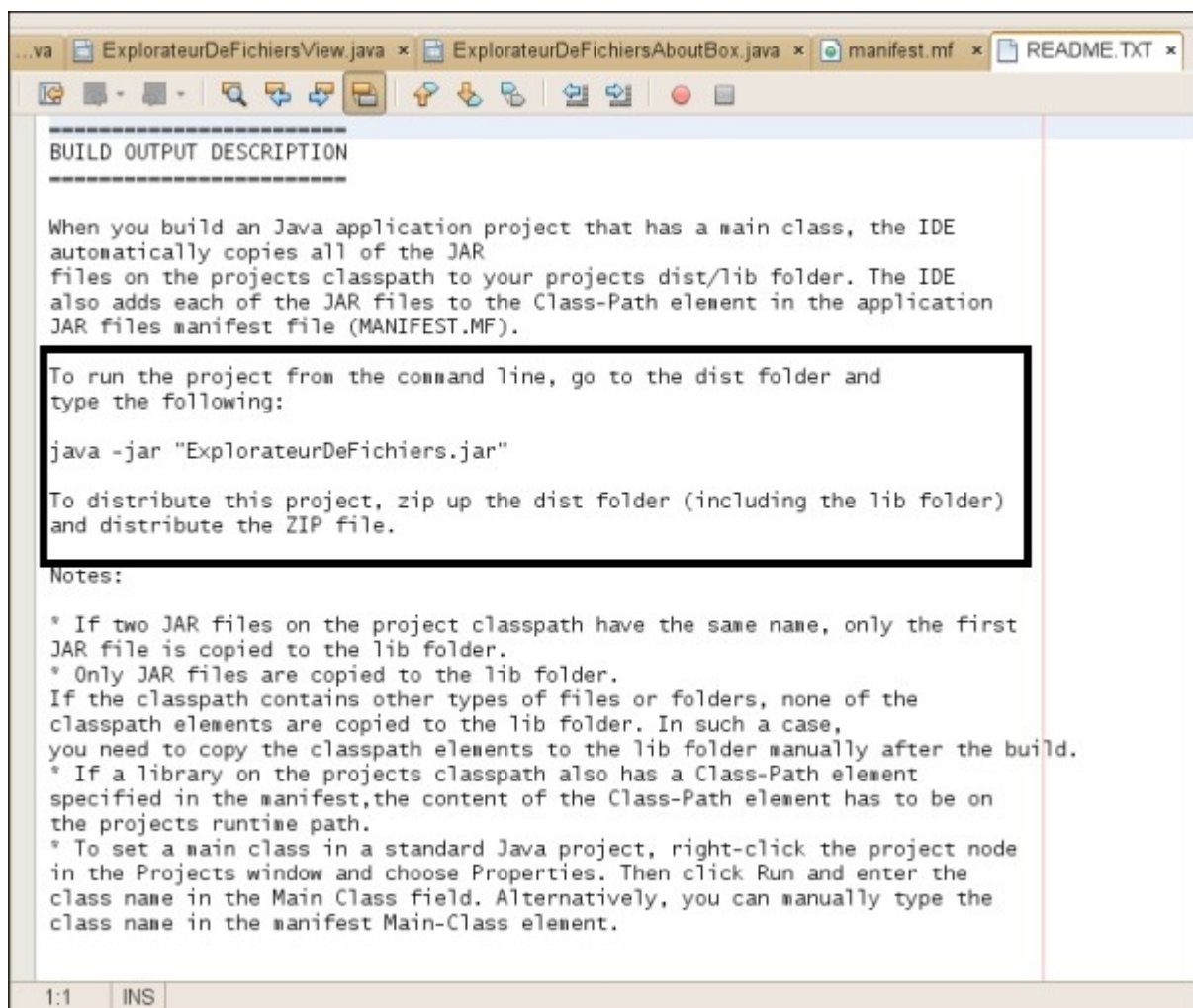
Note :

Pour préparer le projet pour une distribution, il suffit de zipper le contenu du dossier "dist" y compris le dossier "lib"



47. Le fichier README.TXT :

Les informations nécessaires pour exécuter le fichier .jar et pour préparer une distribution sont indiquées dans ce fichier



48. Exécuter le .jar

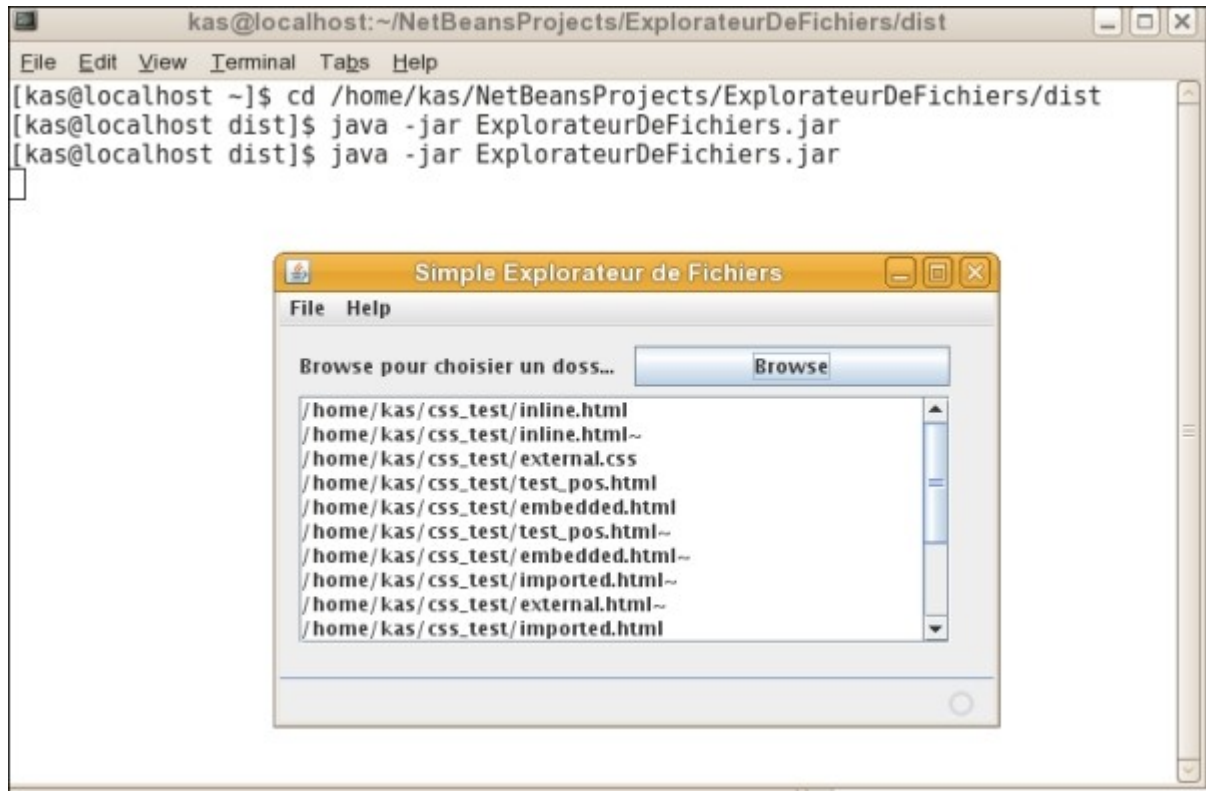
1. Aller dans le dossier NetBeansProjects/ExplorateurDeFichiers/dist

Note :

NetBeansProjects/ExplorateurDeFichiers est le nom du dossier de notre projet

2. A partir de la ligne de commande : `java -jar ExplorateurDeFichiers.jar`

3. L'application marche sans aucuns problèmes



49. Conclusion :

Une application graphique complète avec uniquement quelques de lignes de code. Sans commentaires.

