

---

## Création de composants et intégration de nouveau « bitmap » dans la palette de composants DELPHI

### Création du composant

Première étape préalable, fermez tous les fichiers en cours d'édition dans DELPHI. Utilisez la commande *FICHER / TOUT FERMER*.

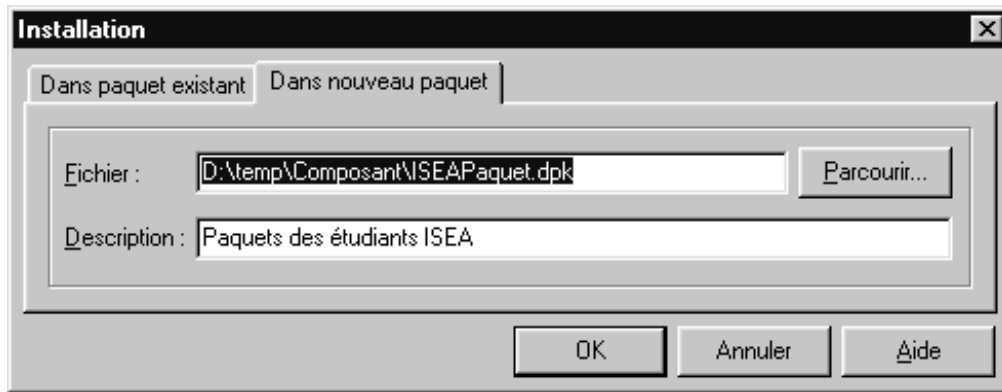
La meilleure manière de créer un composant VISUEL Delphi est d'utiliser le menu *COMPOSANT/NOUVEAU COMPOSANT*. Une fenêtre apparaît, elle permet le paramétrage du nouveau composant à construire.



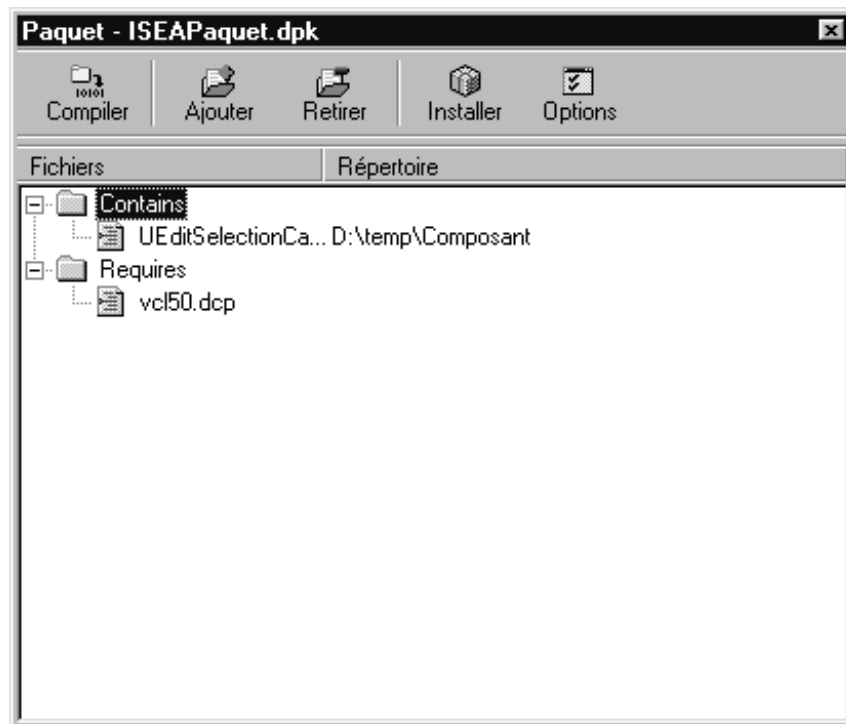
Dans cette boîte figurent successivement :

- ✓ le type ancêtre
- ✓ le nom de la nouvelle classe construite (le composant visuel)
- ✓ la page de la palette dans laquelle elle doit s'inscrire
- ✓ le nom complet (chemin + nom de fichier) de l'unité qui la contient.

Pour une création immédiate, il faut cliquer sur le bouton *INSTALLER*. La boîte suivante permet de spécifier le paquet auquel le composant appartient. Un paquet, dont l'extension de nom de fichier est DPK, peut contenir plusieurs composants, on l'utilise souvent pour regrouper les composants de même nature, ou du même fournisseur. Le mieux est d'associer au paquet une page de la palette de composants.

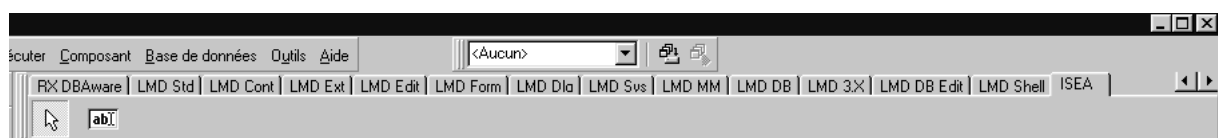


En cliquant sur OK, une boîte vous demande d'installer immédiatement votre composant, cliquez NON et le gestionnaire de composant apparaît. Il est destiné à gérer les composants (1 unité = 1 composant généralement) qui s'y trouvent. Il maintient également, de manière automatique, la liste des paquets externes nécessaires à la compilation de votre composant.



Le bon réflexe est de sauvegarder le paquet et son contenu (les unités) en faisant *FICHER / TOUT ENREGISTER*.

En cliquant sur *INSTALLER*, vous assurez à la fois que toute la compilation est faite, et que le composant va s'intégrer dans la palette. Ainsi, un nouvel onglet ISEA est présent dans la palette.



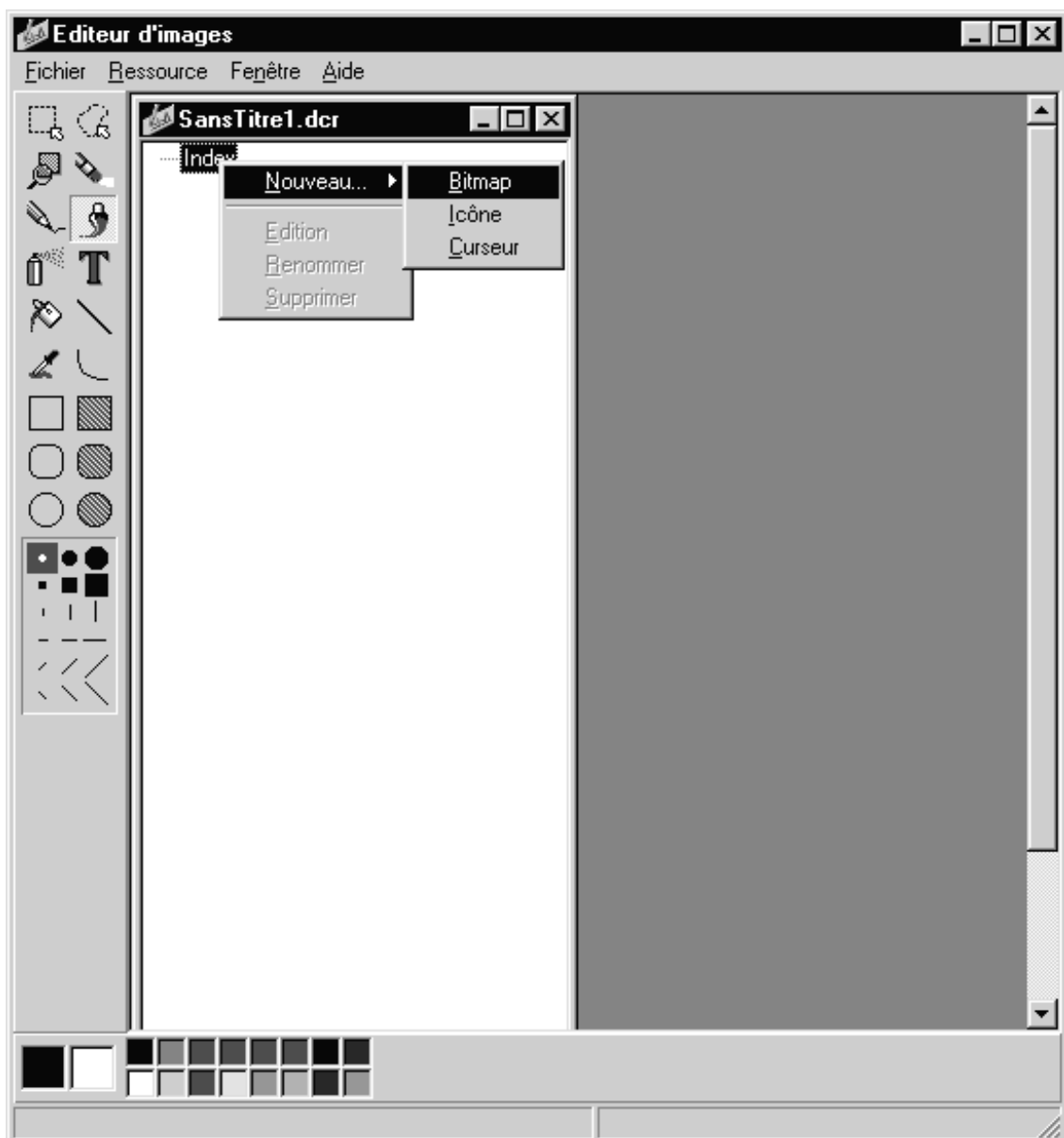
---

## Intégration d'une nouvelle image dans la palette

Notre composant, étant un héritier direct de TEDIT, « hérite » du même bitmap dans la palette de composant. Notre second objectif est de le modifier de manière à personnaliser notre présentation.

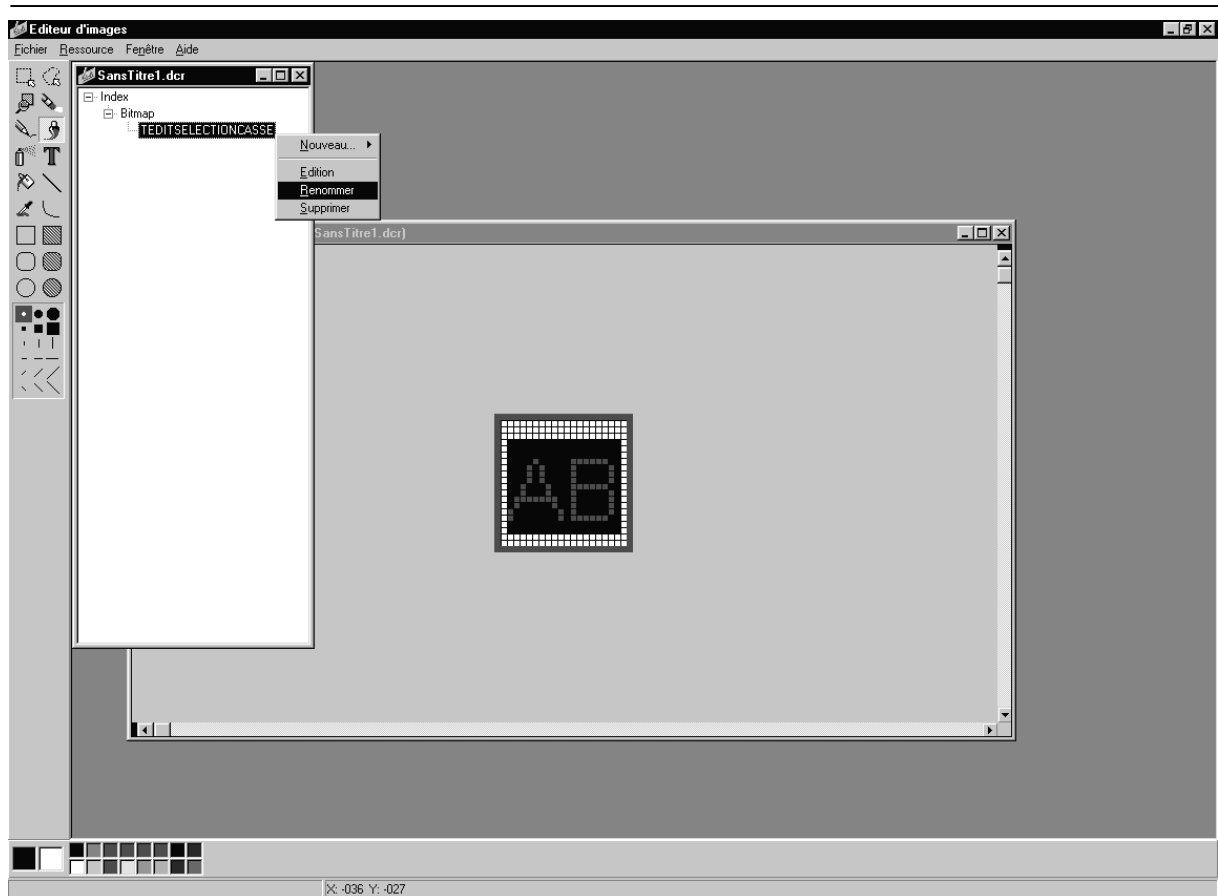
L'outil correspondant se trouve dans *OUTILS / EDITEUR D'IMAGE*. Le logiciel correspondant apparaît, vous devez créer un nouveau fichier ressource de composant en allant dans *FICHIER / NOUVELLE / RESSOURCE COMPOSANTS (.DCR)*.

Dans la boîte qui surgit dans l'éditeur d'image, demandez une nouvelle image BITMAP, de taille 20 x 20 en 16 couleurs.

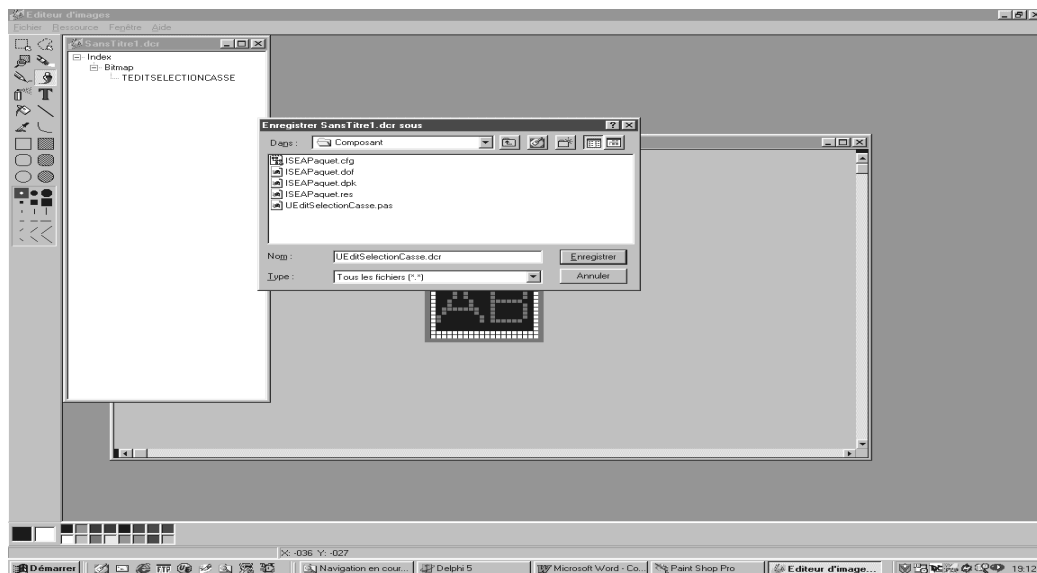


Deux opérations très importantes doivent suivre :

- ✓ nommez le bitmap avec le nom de votre composant

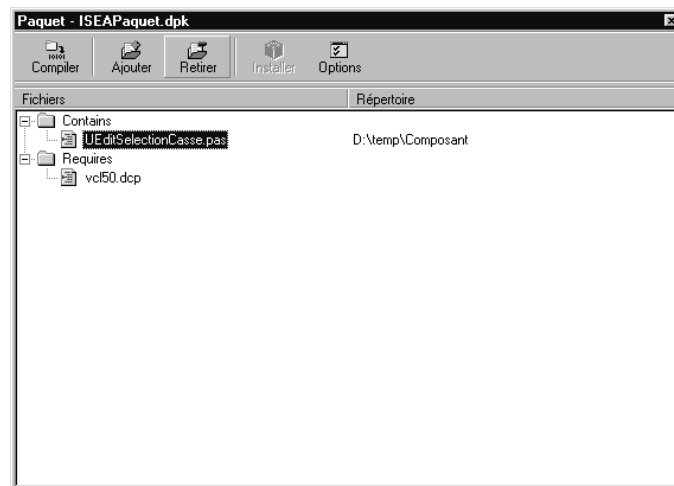


- ✓ nommez votre fichier de composant avec le nom de l'unité dans lequel est défini le composant

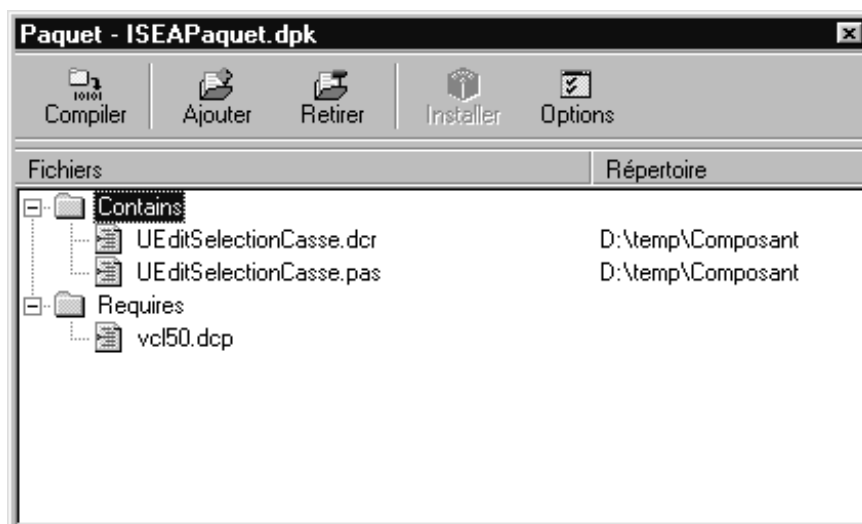


Pour que Delphi prenne en compte ces modifications, il faut par la suite se livrer à une suite d'opérations assez ésothériques :

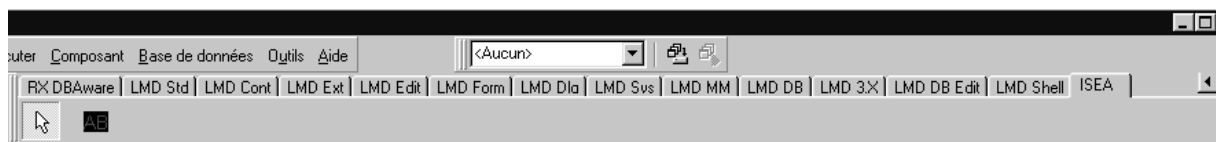
- ✓ retirez l'unité du paquet en le sélectionnant dans la liste, puis en cliquant sur *RETIRER* ;



- ✓ rajoutez-le de nouveau en cliquant sur AJOUTER, puis en sélectionnant le fichier dans la boîte de sélection de fichier qui suit. Cette seconde opération permet à DELPHI de charger automatiquement le fichier DCR associé à l'unité. Ce n'est pas très élégant mais ça marche.



Il ne vous reste plus alors qu'à relancer la compilation et le bon icône s'affiche dans la palette de composants.



Il ne vous reste plus alors qu'à fermer le gestionnaire de paquets et à utiliser le composant dans vos programmes.

## Surcharger une des méthodes (procédure) d'un composant

On veut que dans notre nouveau composant, la procédure KEYPRESS soit surchargée de manière à modifier les entrées :

- ✓ soit les transformer toutes en majuscules
- ✓ soit les transformer toutes en minuscules

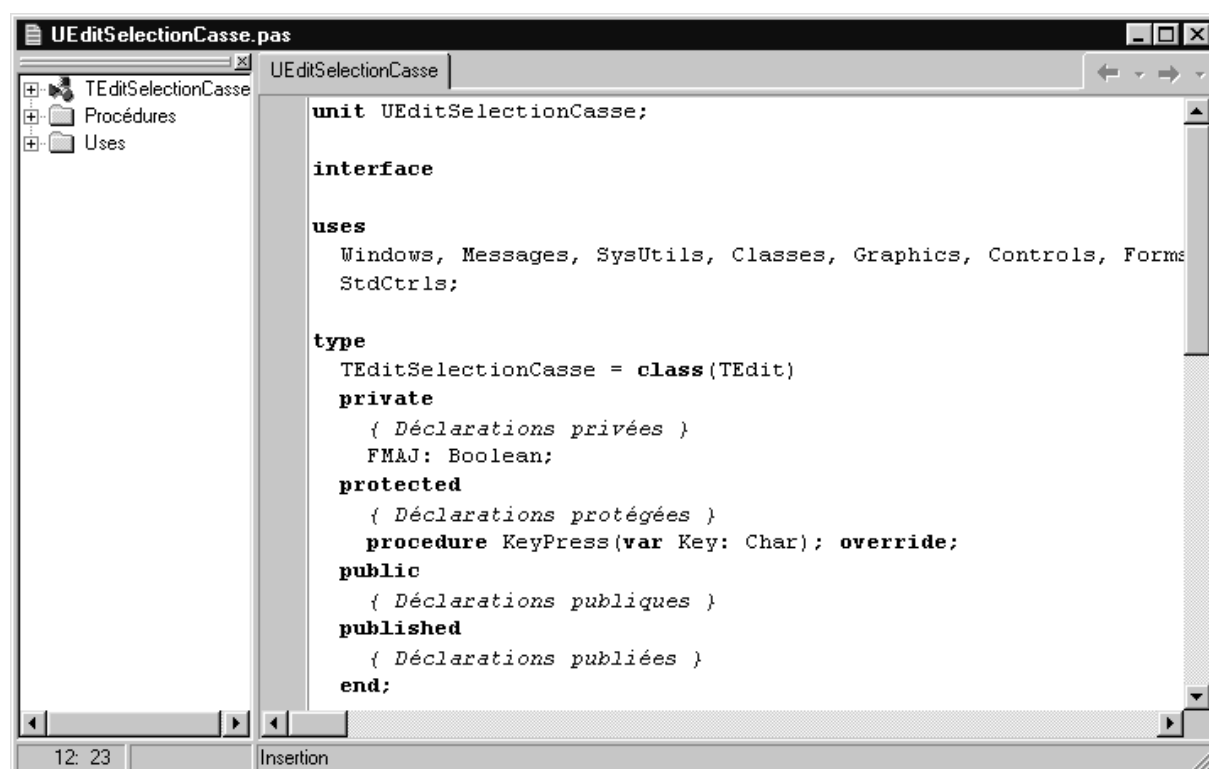
---

On veut utiliser pour cela une propriété interne FMAJ, de type booléen, qui, si elle est VRAIE (TRUE), opte pour la première action, si elle est FAUSSE (FALSE), opte pour la seconde action.

Pour éditer l'unité du composant, la solution la plus sage est d'ouvrir à nouveau le gestionnaire de paquet avec *FICHER / OUVRI*R (l'extension est DPK rappelons-le), puis d'ouvrir la bonne unité en double-cliquant dessus.

La méthode que l'on veut surcharger est KEYPRESS, veillez à respecter scrupuleusement l'en-tête de la procédure et sa visibilité (PROTECTED), utilisez au possible le fichier d'aide pour éviter les erreurs. Comme il s'agit d'une méthode virtuelle (dixit le fichier d'aide DELPHI), vous utilisez le mot clé OVERRIDE lors de la déclaration de l'en-tête.

La propriété FMAJ qui est interne à votre classe peut être déclarée dans la partie PRIVATE.



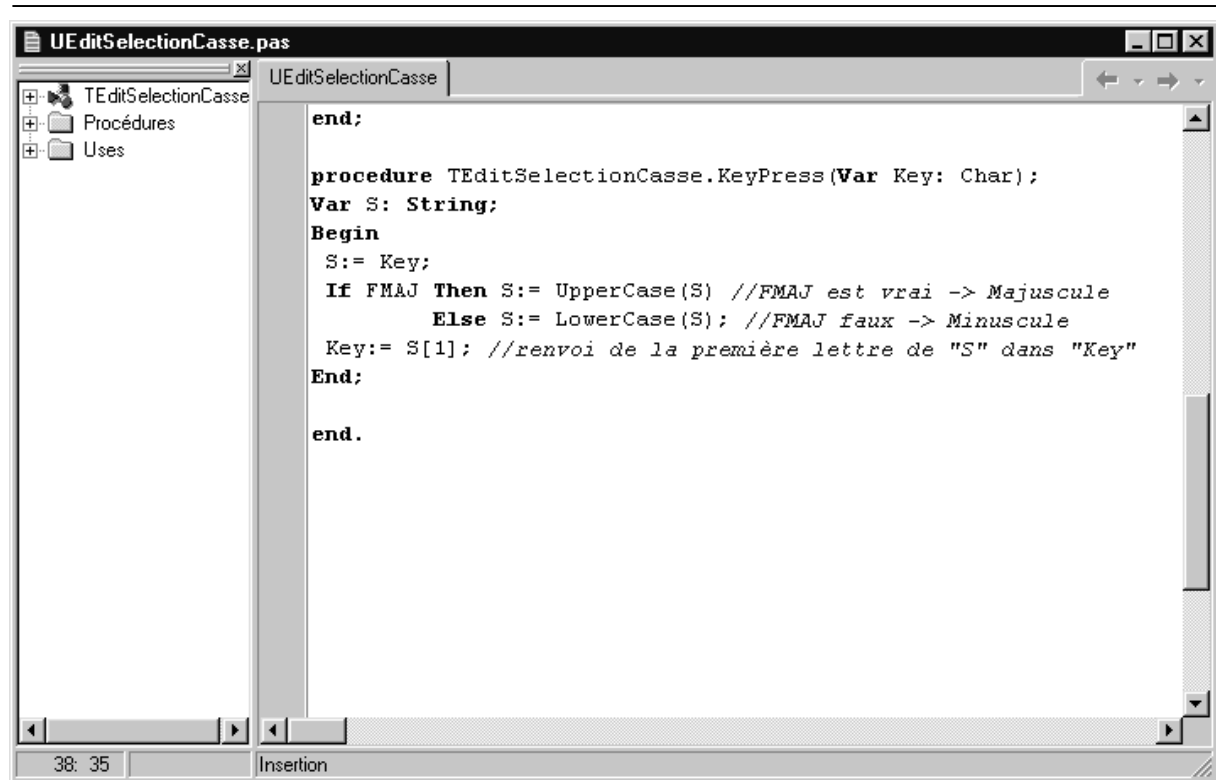
```
unit UEditSelectionCasse;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  StdCtrls;

type
  TEditSelectionCasse = class(TEdit)
  private
    { Déclarations privées }
    FMAJ: Boolean;
  protected
    { Déclarations protégées }
    procedure KeyPress(var Key: Char); override;
  public
    { Déclarations publiques }
  published
    { Déclarations publiées }
  end;
end;
```

Il vous reste alors à implémenter la méthode dans la partie implémentation. Ici, la déclaration de l'en-tête de la procédure doit être complète, c'est-à-dire intégrant le nom de la classe. La déclaration OVERRIDE n'est en revanche, pas nécessaire, ni souhaitable.



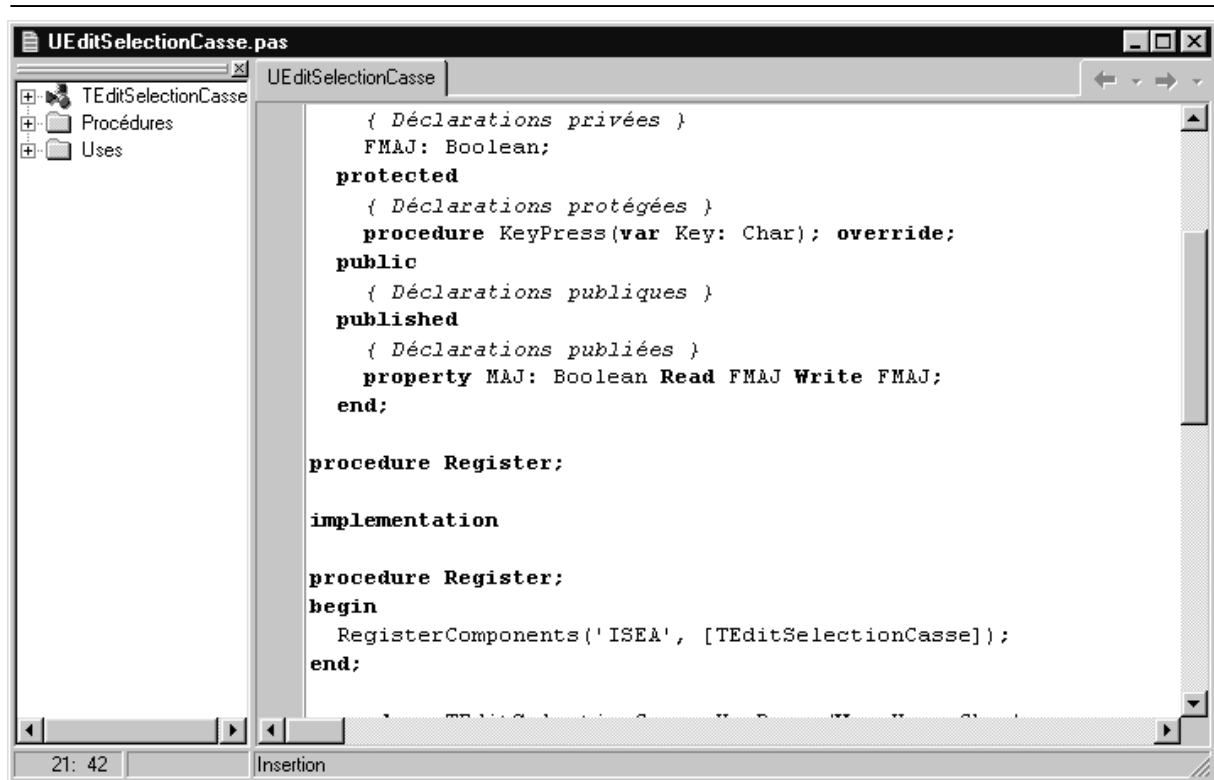
Ceci fait, il faut revenir dans le gestionnaire de paquet et demander à nouveau la compilation (après avoir tout enregistré bien entendu).

## Intégration d'une propriété publiée

Le dernier hic qui ne nous permet pas d'utiliser pleinement notre zone d'édition est l'incapacité de l'utilisateur à accéder à la propriété FMAJ. On aimerait de plus que ceci soit fait de manière interactive dans l'éditeur de propriété de DELPHI.

Pour ce faire, on doit :

- ✓ déclarer une propriété, MAJ par exemple, qui soit reliée en lecture et en écriture à FMAJ (c'est-à-dire lorsque l'on écrit dans MAJ, on place en réalité la valeur dans FMAJ, et lorsque l'on lit MAJ, on lit en réalité la valeur dans FMAJ) ;
- ✓ la placer dans la partie PUBLISHED qui permet de la rendre visible dans l'explorateur des propriétés.



Recompiler de nouveau le paquet, puis fermez-le. Ca y est, votre composant est prêt à l'emploi. Créez une nouvelle application, puis placez votre composant dessus. Non seulement il marchera selon vos prévisions, et vous aurez accès à la propriété MAJ dans l'explorateur de propriété.

