

TUTORIAL C++BUILDER 5

Calcul de racine carrée

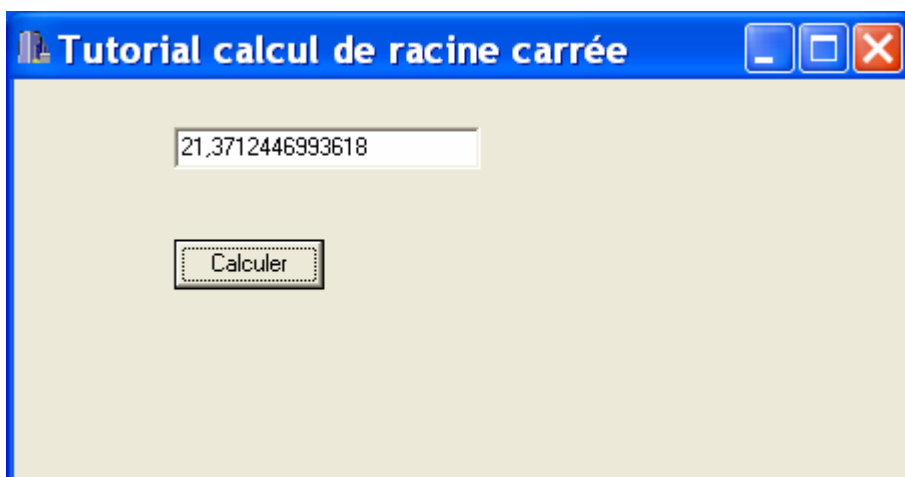
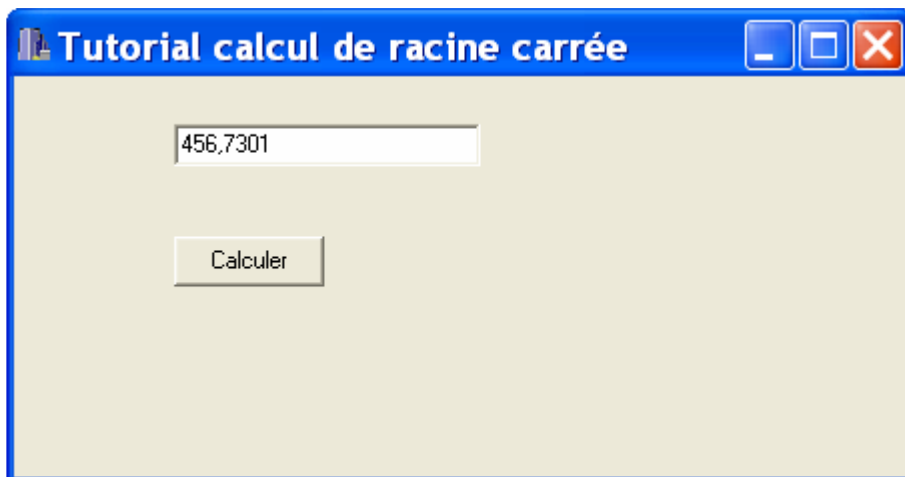
Sommaire

- I- Introduction
- II- Démarrage d'une nouvelle application
- III- Définition des valeurs d'une propriété
- IV- Ajout d'objets à la fiche
- V- Ecriture du code source de la fiche
- VI- Options de projet
- VII - L'application Tutorial.exe

I- Introduction

Ce tutorial s'adresse à ceux qui utilisent pour la première fois l'environnement de développement C++Builder (dans sa version 5).

Il s'agit de créer une application exécutable sous Windows (intitulée **Tutorial.exe**) qui calcule la racine carrée d'un nombre :



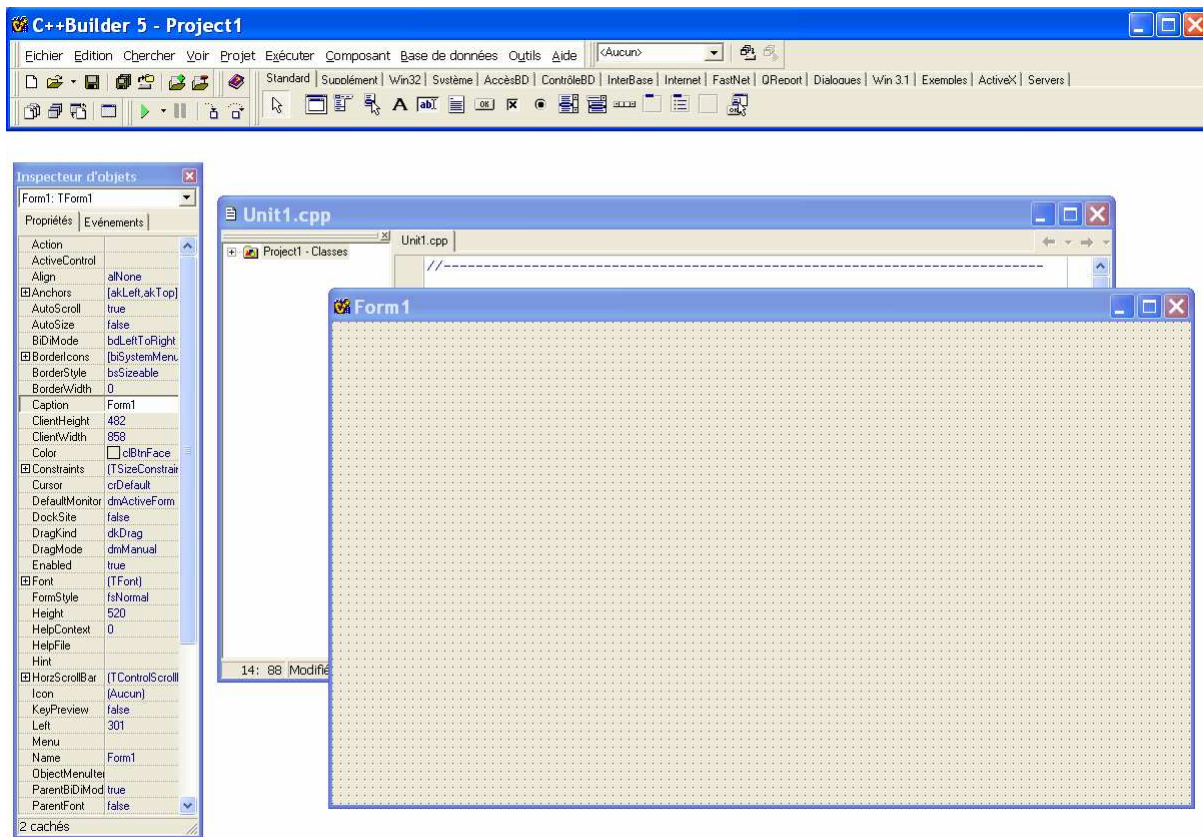
Ainsi : $\sqrt{456,7301} = 21,3712446993618$ (avec 15 chiffres significatifs).

II- Démarrage d'une nouvelle application

Sur votre disque dur, créez un dossier appelé **Tutorial**.

Chaque application est représentée par un projet.

Lorsque vous démarrez C++Builder 5, il ouvre par défaut un projet vide :



Si un autre projet est déjà ouvert, choisissez **Fichier | Nouvelle application** pour créer un nouveau projet.

Choisissez **Fichier | Tout enregistrer** pour enregistrer vos fichiers sur disque dur.

Lorsque la boîte de dialogue **Enregistrer** apparaît :

- Positionnez-vous sur votre dossier **Tutorial**.
- Enregistrez **Unit1** en utilisant le nom par défaut **Unit1.cpp**
- Enregistrez le projet sous le nom **Tutorial.bpr**

C++Builder 5 crée automatiquement les six fichiers suivants :

- **Tutorial.cpp** : le fichier de code source associé au projet.
- **Tutorial.bpr** : le fichier qui contient les options du projet (ce qu'on appelle aussi le makefile).
- **Tutorial.res** : le fichier des ressources Windows.
- **Unit1.cpp** : le fichier de code source associé à la fiche principale du projet. C'est ce que nous appellerons un fichier unité.
- **Unit1.h** : un fichier en-tête associé à la fiche principale du projet. C'est ce que nous appellerons fichier en-tête d'unité.

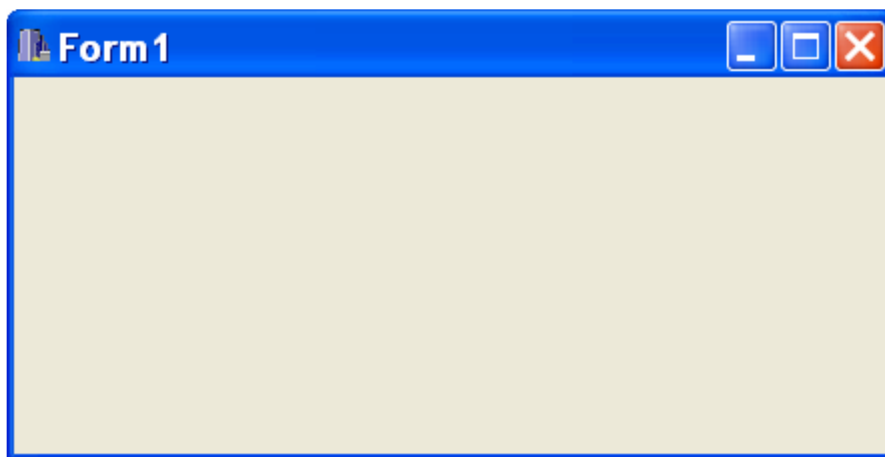
- **Unit1.dfm** : le fichier de ressources stockant les informations sur la fiche principale du projet. C'est ce que nous appellerons un fichier fiche.

Par la suite, vous pourrez enregistrer votre travail à tout moment en choisissant **Fichier | Tout enregistrer**.

Lorsque vous ouvrez un nouveau projet, C++Builder 5 affiche la fiche principale du projet, appelée par défaut **Form1**.

Vous allez créer l'interface utilisateur et les autres parties de votre application en plaçant des composants dans cette fiche.

La fiche par défaut dispose des boutons **Agrandissement** et **Réduction**, d'un bouton **Fermeture** et d'un menu de contrôle. Si vous exécutez à présent la fiche en appuyant sur **F9**, vous verrez ces boutons à l'œuvre :



Pour revenir en mode conception, cliquez sur le bouton de fermeture **X** de la fiche.

Notez qu'après exécution (**F9**), quatre nouveaux fichiers sont créés :

- **Tutorial.exe** : le fichier exécutable
- **Tutorial.obj** : le fichier objet du projet
- **Tutorial.tds** : le fichier des informations de débogage
- **Unit1.obj** : le fichier objet de la fiche

A côté de la fiche, vous verrez l'inspecteur d'objets (touche **F11**).

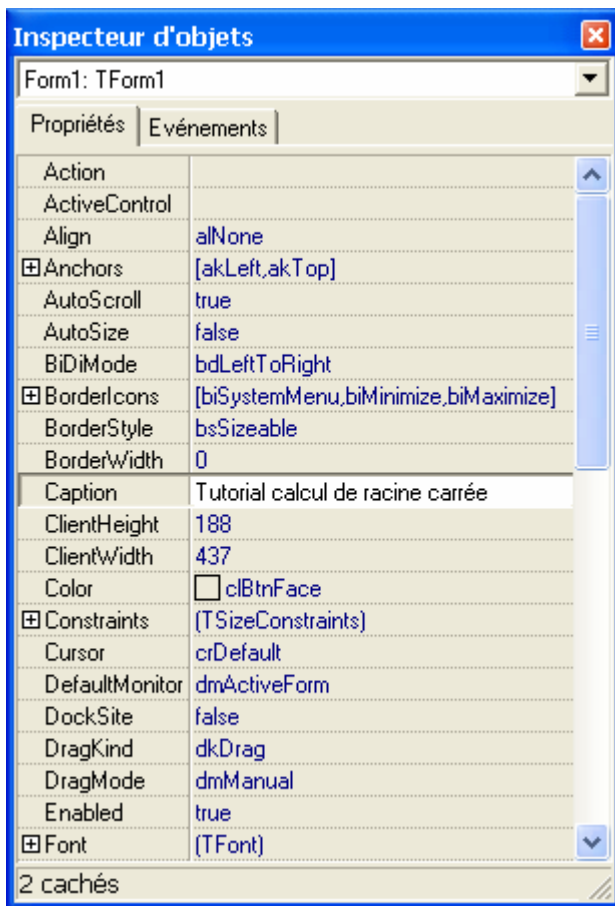
Vous l'utiliserez pour définir les propriétés de la fiche et des composants que vous placerez dessus.

III- Définition des valeurs d'une propriété

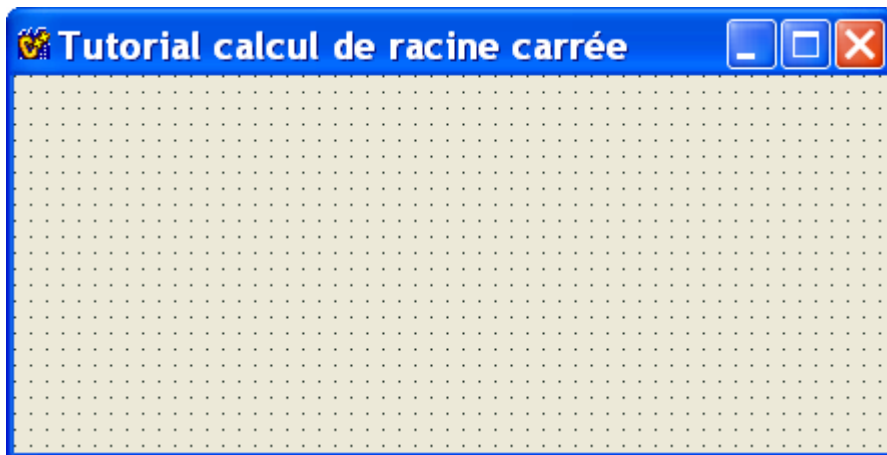
Lorsque vous utilisez l'inspecteur d'objets pour définir des propriétés, c'est C++Builder 5 qui met à jour pour vous le code source. La définition des valeurs dans l'inspecteur d'objets est appelée paramétrage en mode conception.

Vous pouvez ainsi changer le libellé de la fiche **Form1**.

Trouvez dans l'inspecteur d'objets la propriété **Caption** de la fiche et entrez "Tutorial calcul de racine carrée" à la place du libellé par défaut "Form1" :



La fiche devient :



IV- Ajout d'objets à la fiche

Avant de commencer à ajouter des objets (appelés également composants) à la fiche, vous devez réfléchir à la meilleure manière d'organiser l'interface utilisateur de votre application.

L'interface utilisateur est ce qui permet à l'utilisateur de votre application d'interagir avec elle, elle doit donc être conçue pour simplifier l'utilisation.

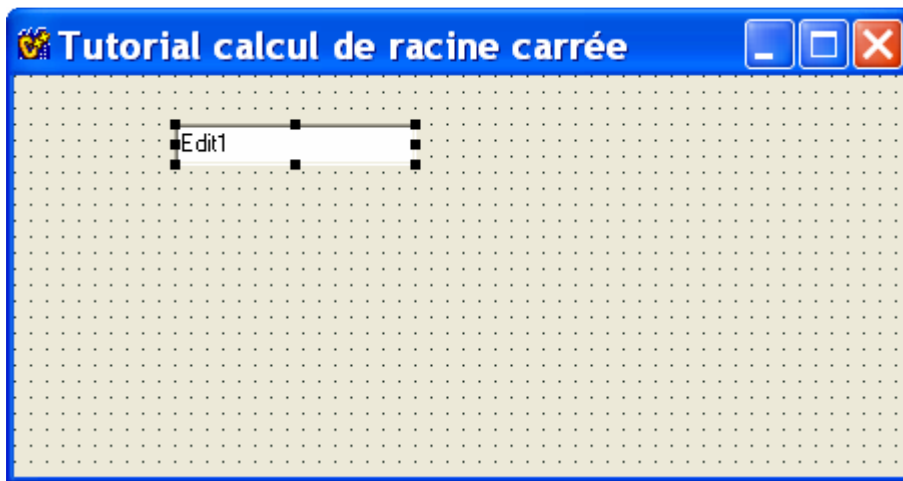
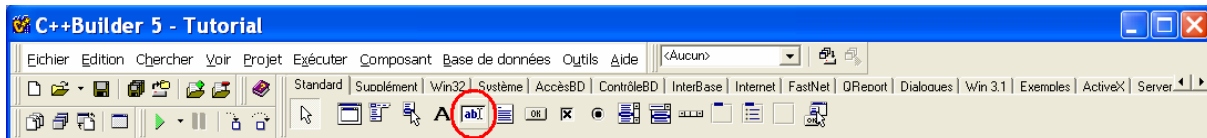
Ici les choses sont simples : notre application nécessite une zone de texte et un bouton.

La palette de composants représente les composants par des icônes et les regroupe sur plusieurs pages à onglet. Pour ajouter un composant à une fiche, il suffit de le sélectionner dans la palette et de cliquer. Pour obtenir de l'aide sur un composant, sélectionnez-le (dans la palette ou dans la fiche) et appuyez sur **F1**.

Pour créer une zone de texte, déposez dans la fiche un composant **Edit** :

Cliquez dans la page **Standard** de la palette de composants. Pour trouver le composant **Edit**, pointez un moment sur une icône de la palette : C++Builder 5 affiche un message d'aide montrant le nom du composant.

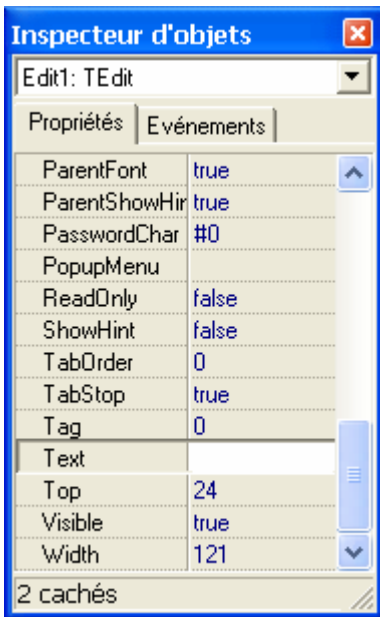
Quand vous avez trouvé le composant Edit, double-cliquez dessus pour le placer dans la fiche :



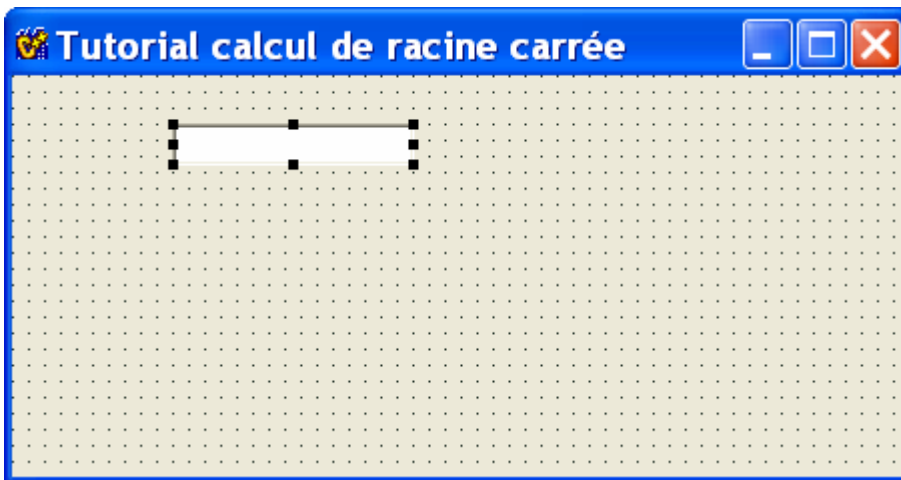
Chaque composant C++Builder 5 est une classe ; placer un composant dans la fiche crée une instance de cette classe. Une fois le composant placé dans la fiche, C++Builder 5 génère le code nécessaire à la construction d'un objet instance lors de l'exécution de votre programme.

Initialisation de la propriété **Text** de **Edit1** :

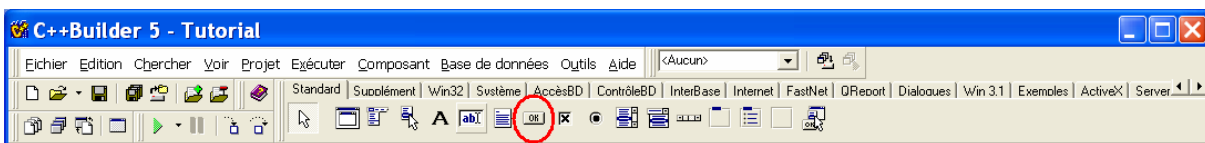
Pour ce faire, cliquez dans la fiche sur **Edit1** pour le sélectionner, puis choisissez la propriété **Text** dans l'inspecteur d'objets. Remplissez avec un « texte vide » :

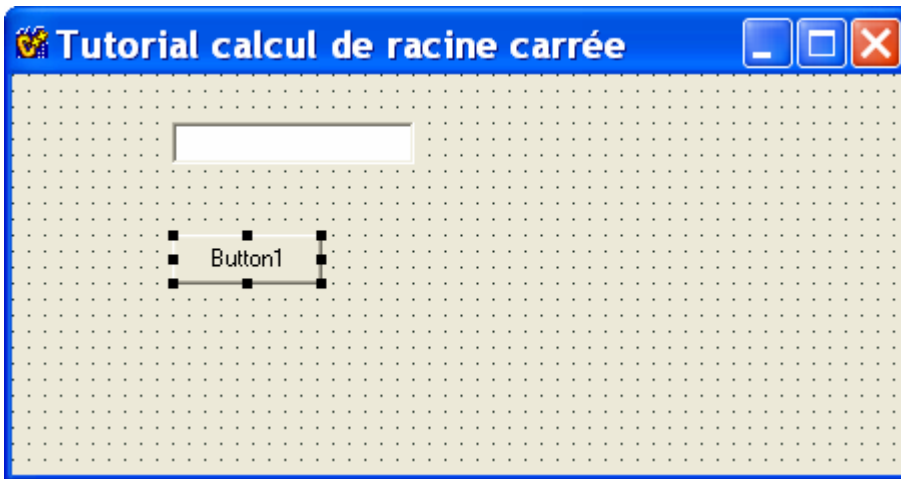


La fiche devient :

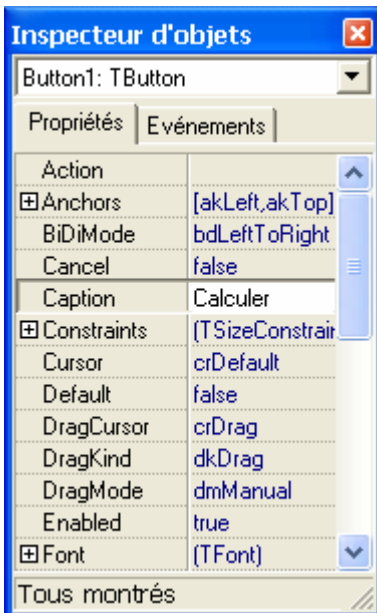


Ajoutez à la fiche un composant **Button** :

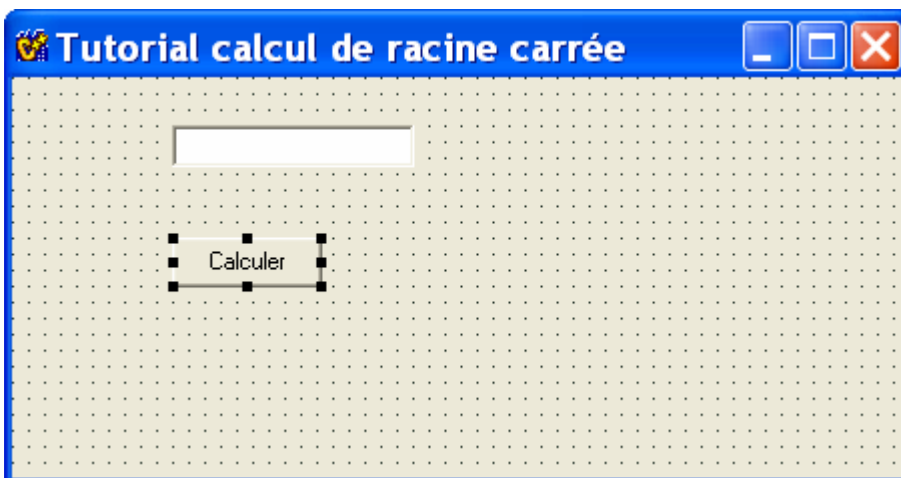




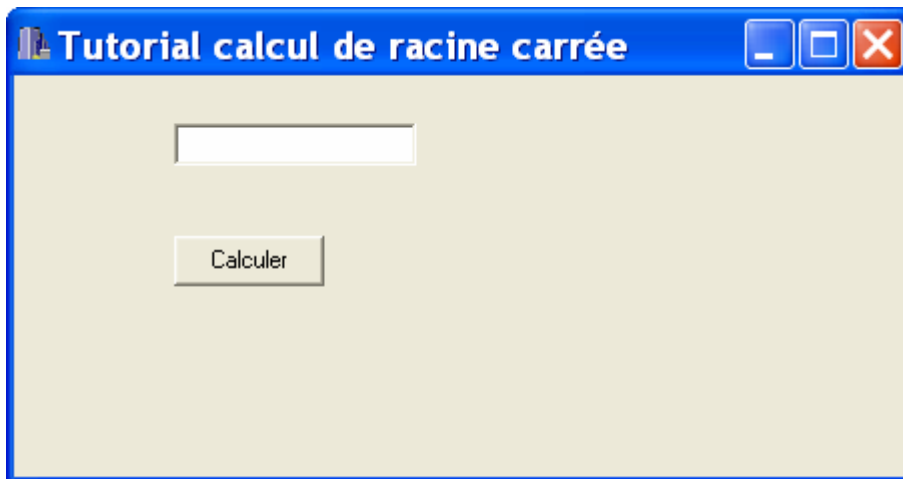
Initialisation de la propriété **Caption** de **Button1** :



La fiche devient :



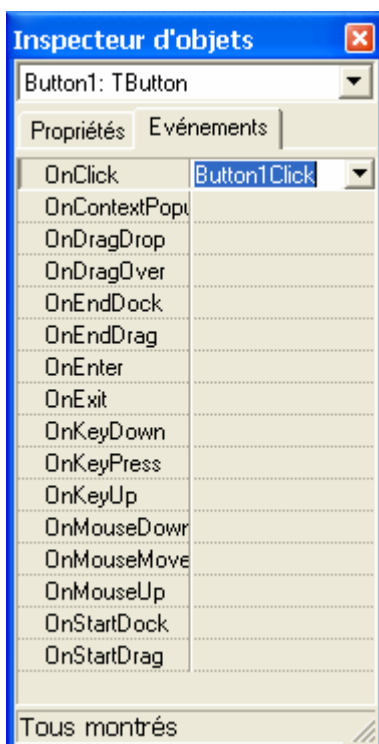
Si vous exécutez maintenant la fiche en appuyant sur **F9**, vous obtenez :



Pour revenir en mode conception, cliquez sur le bouton de fermeture **X** de la fiche.

V – Ecriture du code source de la fiche

Cliquez dans la fiche sur **Button1** pour le sélectionner, puis choisissez l'événement **OnClick** dans l'inspecteur d'objets :



Dans la fenêtre **Unit1.cpp** les lignes de code suivantes sont automatiquement créées :

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
}
```


Complétez manuellement avec les quatre lignes de code suivantes :

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    double nombre;
    nombre = Edit1->Text.ToDouble();
    nombre = sqrt(nombre);
    Edit1->Text = FloatToStr(nombre);
}
```

Le spécificateur de type **double** permet de définir un identificateur (nombre) comme type de données à virgule flottante.

La méthode **ToDouble** convertit une chaîne de texte de type AnsiString en valeur à virgule flottante.

La fonction mathématique **sqrt** calcule la racine carrée d'un nombre.

FloatToStr convertit un nombre à virgule flottante en une représentation de type chaîne.

La conversion utilise le format général des nombres avec 15 chiffres significatifs.

Pour résumer, cette fonction sera appelée à chaque fois que l'on cliquera sur le bouton.

La chaîne de texte est alors convertie en nombre.

On calcule ensuite la racine carrée de ce nombre.

Enfin, on convertit le résultat numérique en chaîne de texte pour affichage dans la zone de texte.

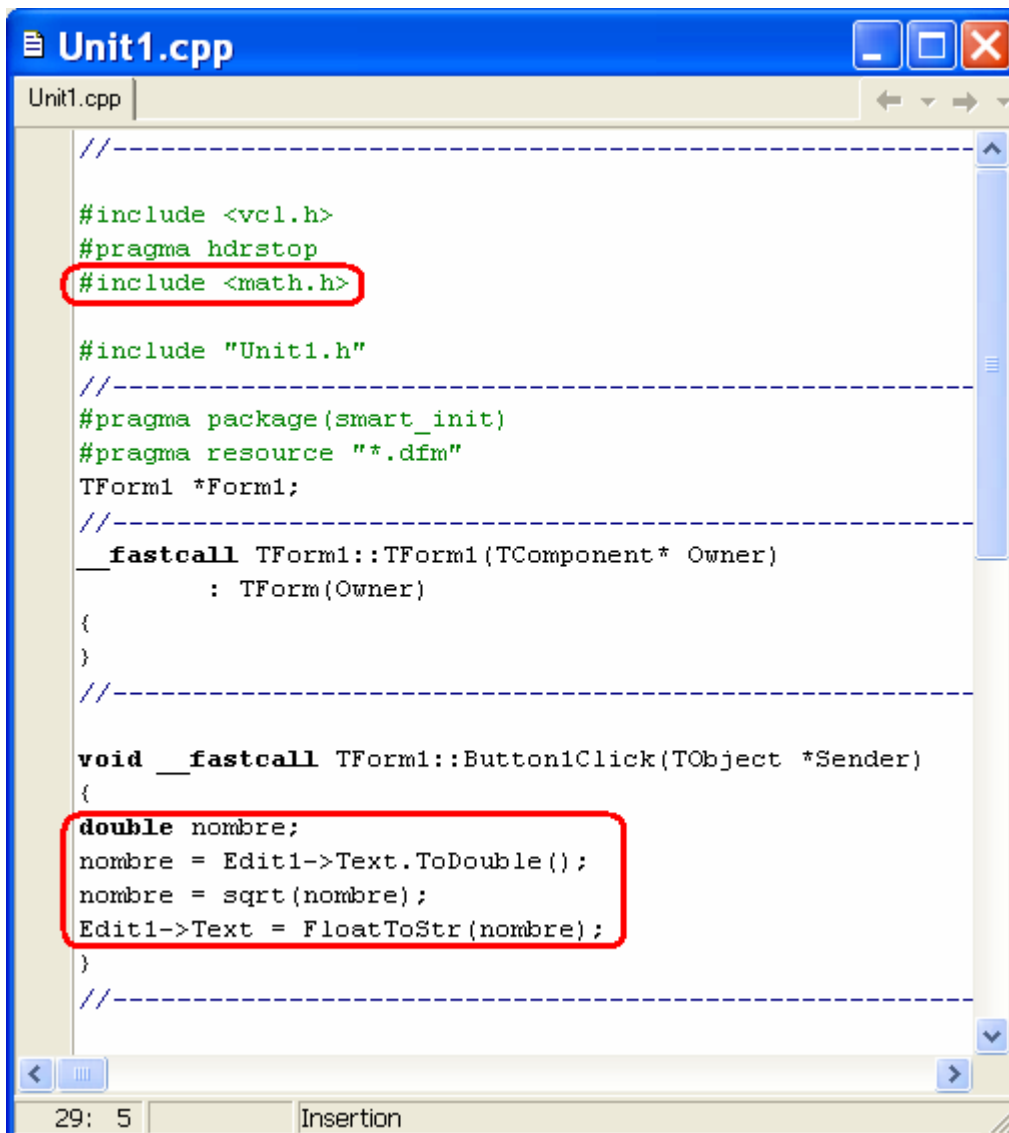
Attention : Pour pouvoir utiliser la fonction mathématique **sqrt**, vous devez inclure manuellement la directive :

```
#include <math.h>
```

Autrement, à l'exécution (**F9**) vous aurez un message d'erreur du compilateur :

```
[C++ Erreur] Unit1.cpp : E2268 Appel à une fonction non définie 'sqrt'
```

En définitive, le code source de la fiche est :

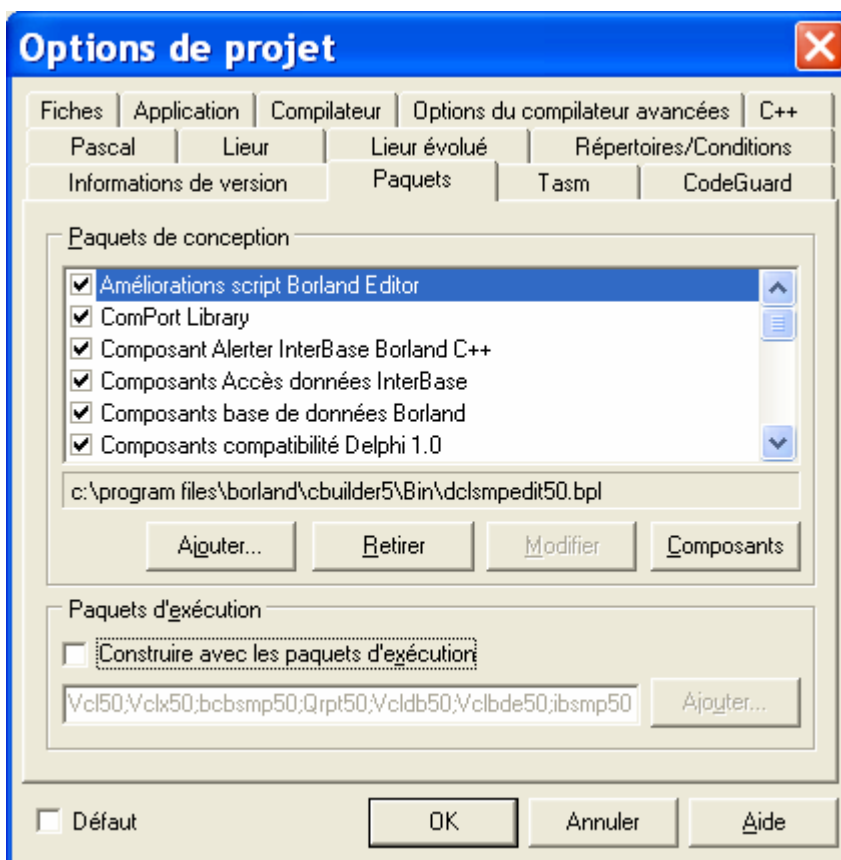
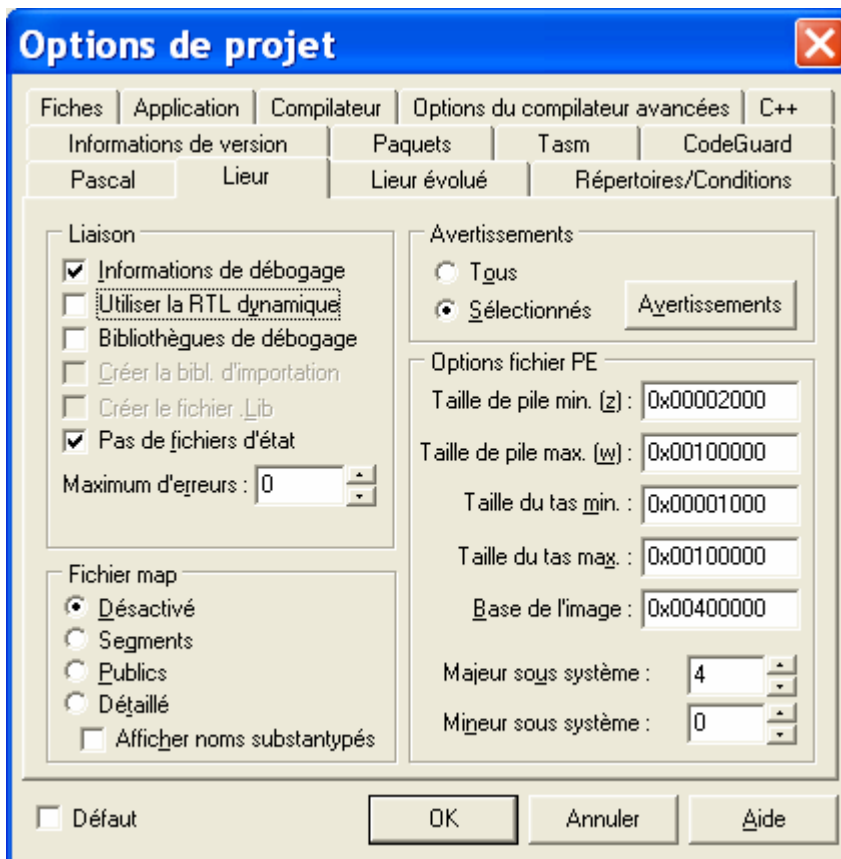


```
Unit1.cpp
//-----
#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
double nombre;
nombre = Edit1->Text.ToDouble();
nombre = sqrt(nombre);
Edit1->Text = FloatToStr(nombre);
}
//-----
29: 5 Insertion
```

VI- Options de projet

Pour créer un exécutable autonome, allez dans **Projet | Options** :

- Dans l'onglet **Lieur**, décocher **Utiliser la RTL dynamique**
- Dans l'onglet **Paquets**, décocher **Construire avec les paquets d'exécution**

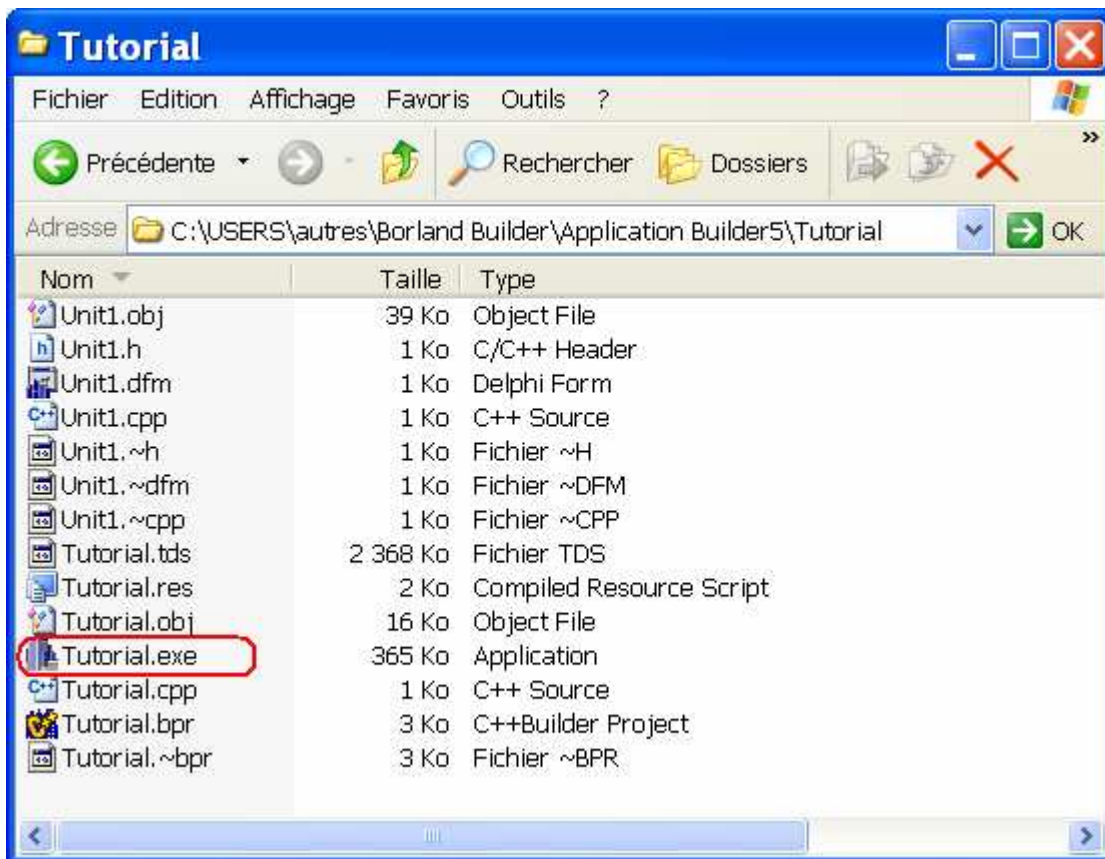


Appuyez sur **F9** pour exécuter l'application.

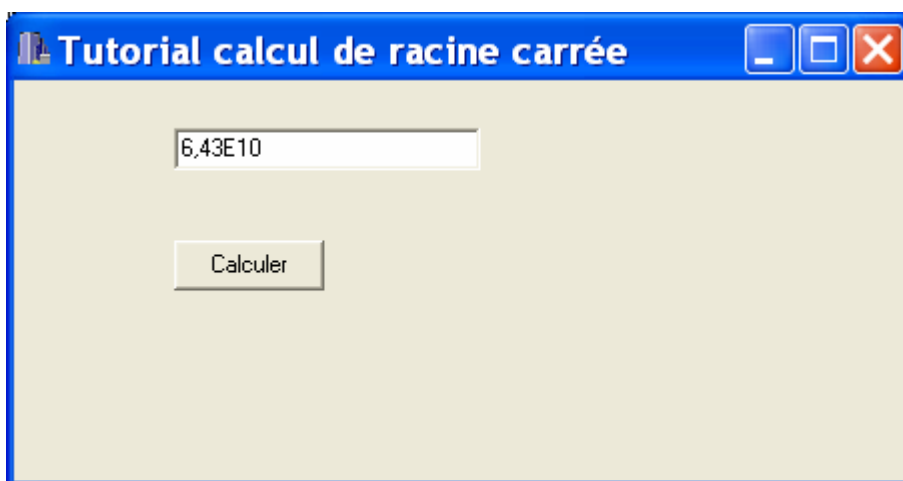
Tout enregistrer (Maj + Ctrl + S).
Puis quittez C++Builder 5.

VII - L'application Tutorial.exe

Ouvrir l'application (**Tutorial.exe**) qui se trouve dans le dossier **Tutorial** :

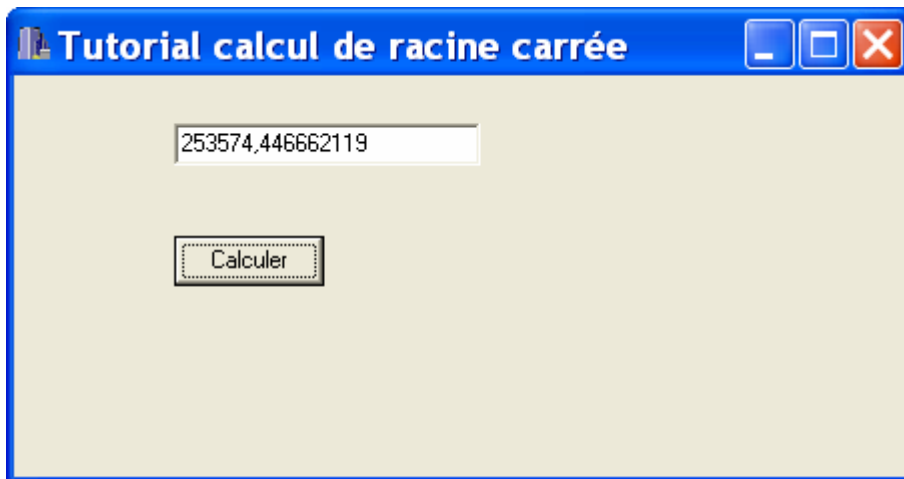


Saisissez un nombre (vous pouvez utiliser la notation scientifique) :



Cliquez sur le bouton **Calculer**.

Et voilà la racine carrée avec 15 chiffres significatifs :



L'application est totalement autonome.

Vous pouvez maintenant utiliser **Tutorial.exe** sur n'importe quel ordinateur qui fonctionne sous Windows (95, NT, 98, XP, Vista ...).

Bibliographie

- Aide de C++Builder 5