

# Installation de Virtual Pascal sous Windows 7

## et utilisation de Geany comme éditeur alternatif

Par Alcatîz 

Date de publication : 14 août 2012

DÉBUTANT

Durée : 15 minutes

**Virtual Pascal** est un compilateur multiplateforme très performant, permettant de créer des applications pour Windows, Linux et OS/2 (32 bits).

Il n'a malheureusement pu évoluer au même rythme que son concurrent Free Pascal.

Ce tutoriel détaille les étapes d'installation de Virtual Pascal sous Windows 7 ainsi que la configuration de l'environnement de développement **Geany** en tant qu'interface de substitution à l'éditeur natif.

**Commentez**

Introduction.....	3
Installation de Virtual Pascal.....	4
Téléchargement.....	4
Installation.....	6
Configuration.....	12
Compilation de la RTL.....	17
Compilation d'un programme console.....	18
Compilation d'une application GUI.....	25
Options du compilateur.....	29
Installation de Geany.....	31
Téléchargement.....	31
Installation.....	31
Configuration.....	35
Conclusion.....	39
Remerciements.....	39

## Introduction

Jusqu'en 2005, par rapport à son concurrent **Free Pascal**, **Virtual Pascal** compilait du code plus compact, plus rapide et il le faisait même beaucoup plus vite. Par contre, il ne supportait que trois plateformes : **Windows**, **Linux** et **OS/2**. L'éditeur natif fonctionnait de manière identique sur les trois plateformes, tandis que Free Pascal ne fournissait un éditeur que pour Windows. Alors pourquoi Free Pascal s'est-il imposé et Virtual Pascal est-il tombé dans l'oubli ? Parce que ce dernier était développé en Assembleur, ce qui rendait son évolution très pénible.

Virtual Pascal est compatible avec **Turbo Pascal 7**, **Borland Pascal 7** et **Delphi** - jusqu'à la version 3. Il constitue une très bonne solution pour le portage en 32 bits de code 16 bits ; le portage de la bibliothèque OWL est particulièrement réussi.

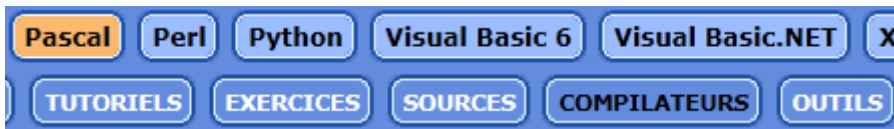
Jusqu'à Windows XP, l'intégration au système de l'éditeur natif était très bonne (par exemple, le copier/coller depuis ou vers le presse-papier de Windows). Mieux, la présence d'un système de macros d'édition était un atout très intéressant. Sous Windows 7, cela n'est plus du tout le cas, au point que certaines combinaisons de touches ne sont même plus reconnues. C'est pour cela que ce tutoriel va au-delà de l'installation de Virtual Pascal et propose son utilisation avec un éditeur externe comme **Geany**.

## Installation de Virtual Pascal

**Developpez.com** est un des rares sites qui proposent le téléchargement de Virtual Pascal et même le seul qui mette à votre disposition un script d'installation sous Linux. Mais là n'est pas le propos, nous nous limitons à Windows 7 dans cet article.


## Téléchargement

Vous pouvez accéder à l'interface de téléchargements de Developpez.com depuis **le forum** ou depuis **le portail Pascal**, en cliquant sur le bouton **Compilateurs** de la barre supérieure :



Dans la liste des environnements de développement, sélectionnez **Virtual Pascal** (ou utilisez le moteur de recherche au-dessus de la page) :



 **Virtual Pascal**  
Éditeur : Vitaly Mirianov et Allan Mertner + Mise à jour le :  
07/06/2011 · Licence : Gratuit · Téléchargé 93 fois ·  
12,9 Mo Date de sortie : 16/12/2004

---

*Version 2.1 build 279 - 16/12/2004.*

Virtual Pascal est un compilateur très performant et d'une compatibilité évoluée avec Borland Pascal et Delphi (jusqu'à la version 3). Il offre une passerelle 32 bits vers la bibliothèque OWL (Object Windows Library). Son EDI avec debugger intégré est excellent. Virtual Pascal est fourni avec toute la documentation nécessaire.

Virtual Pascal permet de créer des applications pour Windows, Linux et OS/2 (32).

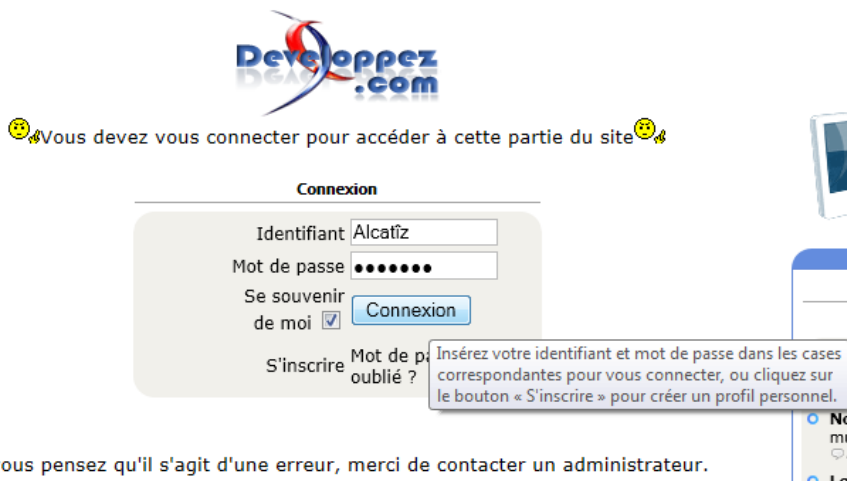
Différents tutoriels sont à votre disposition :

- [Configuration de Virtual Pascal sous Windows](#)
- [Installation de Virtual Pascal sous Linux](#)
- [Compiler et exécuter une application en Virtual Pascal sous Linux](#)
- [Cours complet de programmation d'applications Win32 en Virtual Pascal avec OWL](#)

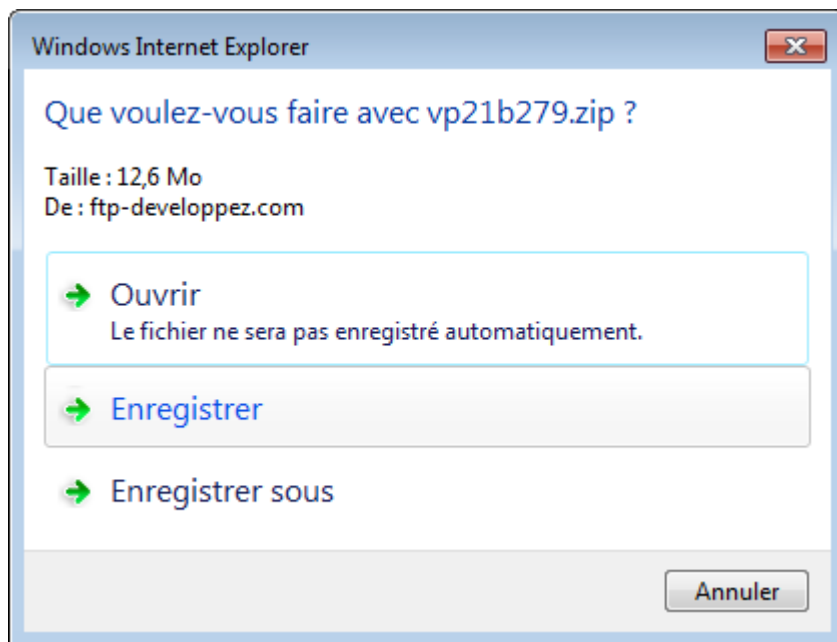
Sur la fiche du produit, cliquez sur le gros bouton **Télécharger** :



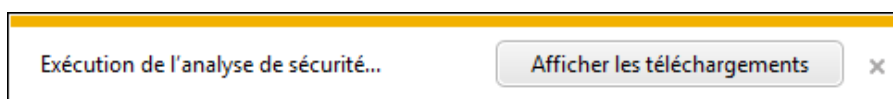
Pour pouvoir effectuer le téléchargement, vous devez être identifié. Si ce n'est pas le cas, vous serez confronté à cet écran :



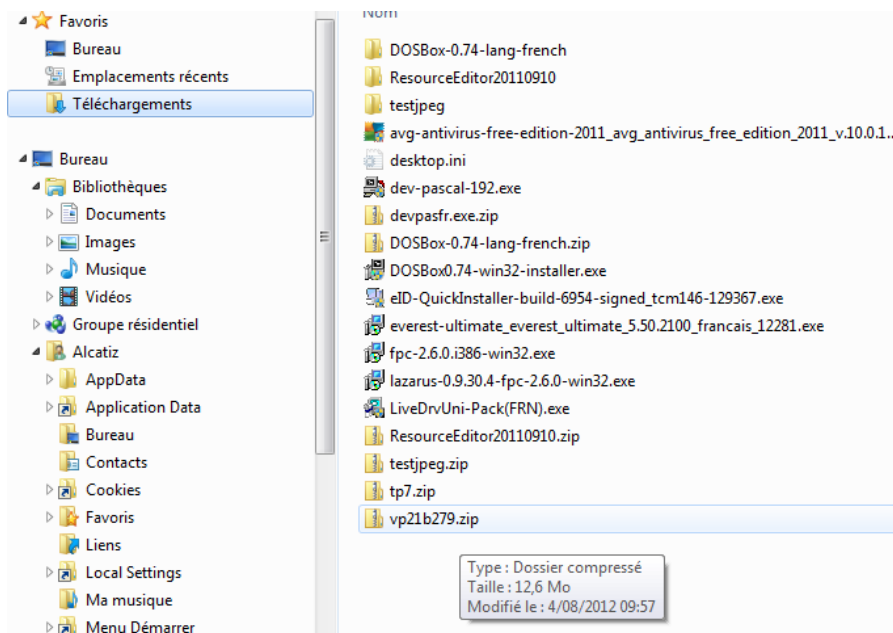
Le fichier à télécharger s'appelle **vp21b279.zip** (pour *Virtual Pascal 2.1 build 279*). Indiquez à Windows que vous voulez l'enregistrer :



Windows effectue une analyse de sécurité :

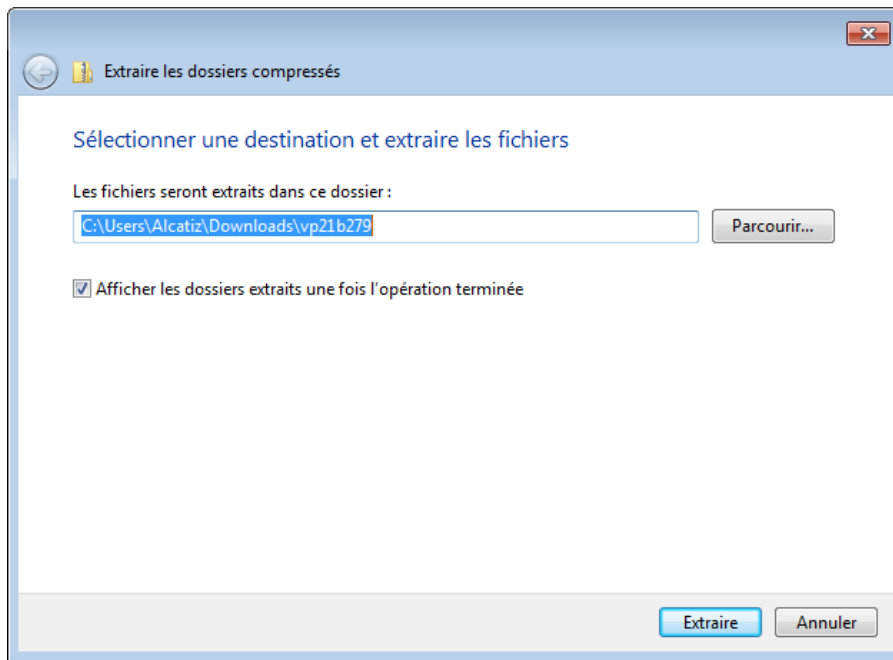


Le fichier est bien présent dans le dossier de téléchargements :

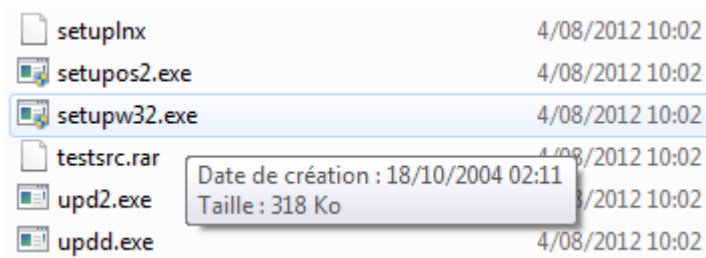


## Installation

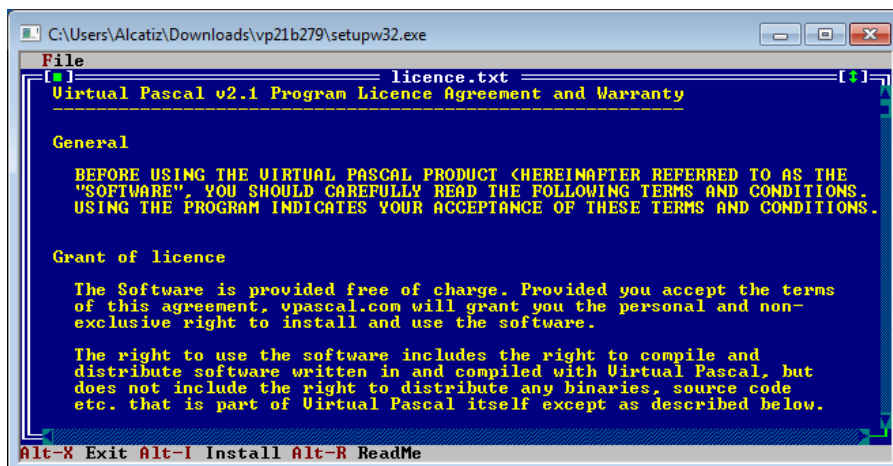
Faites un clic droit sur le fichier et sélectionnez **Extraire** dans le menu contextuel. Vous pouvez conserver le choix par défaut du dossier de destination et laisser cochée l'option **Afficher les dossiers extraits une fois l'opération terminée** :



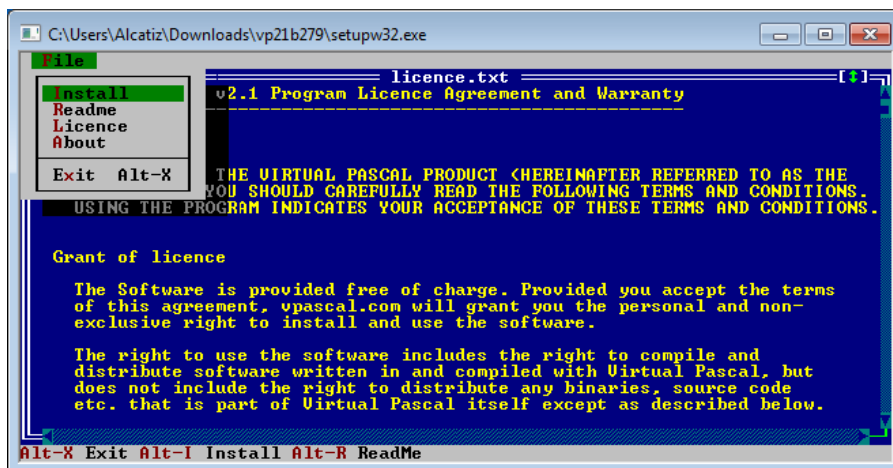
Dans le dossier de destination, cherchez le programme d'installation pour Windows, qui se nomme **setupw32.exe** :



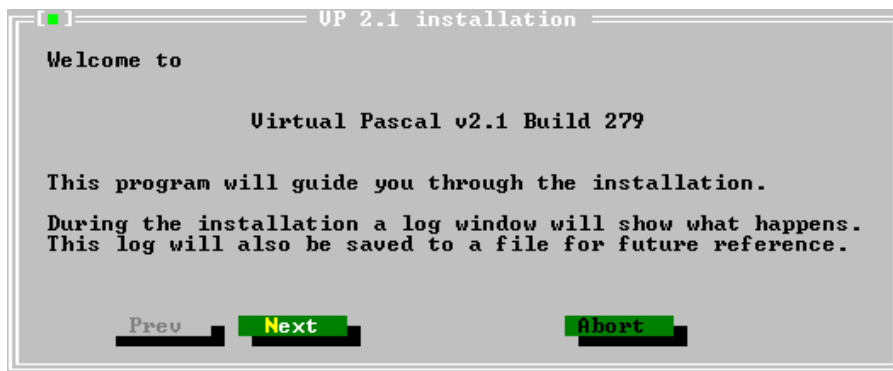
Vous avez sans doute deviné que *setuplnx* était destiné à Linux et *setupos2.exe* à OS/2. Double-cliquez sur **setupw32.exe** et répondez par l'affirmative au dialogue de confirmation de Windows. Le programme d'installation se lance :



Vous pouvez tout de suite constater que Windows 7 intercepte les combinaisons de touches comme **Alt-I**. Aussi est-ce à la souris que nous allons lancer l'installation. Cliquez sur le menu **File/Install** :



Comme le programme d'installation est poli, il vous souhaite la bienvenue. Cliquez sur **Next** :



Dans le dialogue suivant, vous pouvez sélectionner votre type d'installation. Vous avez le choix entre :

- une installation complète (**Everything**) ;
- une installation minimale ne reprenant que tout ce qui est libre (**Everything that's free**) ;
- une installation personnalisée (**Custom selection**).



Quelques explications sont nécessaires. Comme vous l'avez lu dans l'introduction, Virtual Pascal est compatible avec **Borland Pascal 7** et **Delphi**. Du premier nommé, il reprend la bibliothèque OWL et du second, la VCL. Seulement, ces deux bibliothèques sont soumises à licence ; Virtual Pascal ne supportera pas OWL si vous ne disposez pas de Borland Pascal 7 et la VCL si Delphi n'est pas installé sur votre machine.

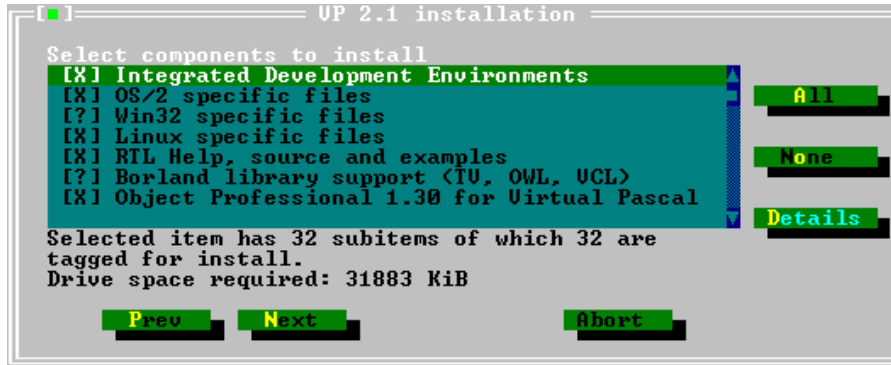
Donc, si Borland Pascal et Delphi sont tous deux installés sur votre machine, sélectionnez l'installation complète (ou l'installation personnalisée si vous voulez sélectionner finement les composants à installer) ; si seulement l'un des deux l'est, sélectionnez l'installation personnalisée ; sinon, conservez l'option par défaut. Cliquez ensuite sur **Next**.





Notez que l'installation complète ou personnalisée nécessitera de votre part d'indiquer les répertoires dans lesquels se trouvent les codes sources d'OWL (pour Borland Pascal) ou de la VCL (pour Delphi).

Voici le dialogue de choix d'options de l'installation personnalisée :



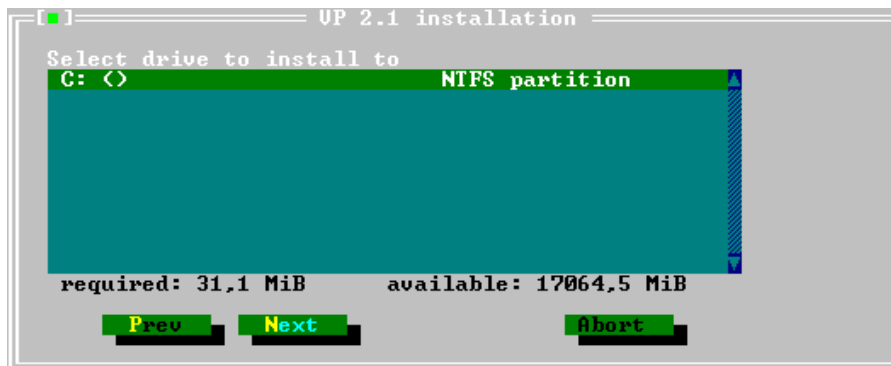
Le bouton **Details** ouvre une sous-liste de composants. Lorsqu'une option est marquée avec le point d'interrogation, cela signifie que seule une partie des sous-options sont cochées.

Une fois que vous avez fini votre sélection de composants, cliquez sur **Next**.



Pour la suite du tutoriel, nous avons opté pour l'installation minimale.

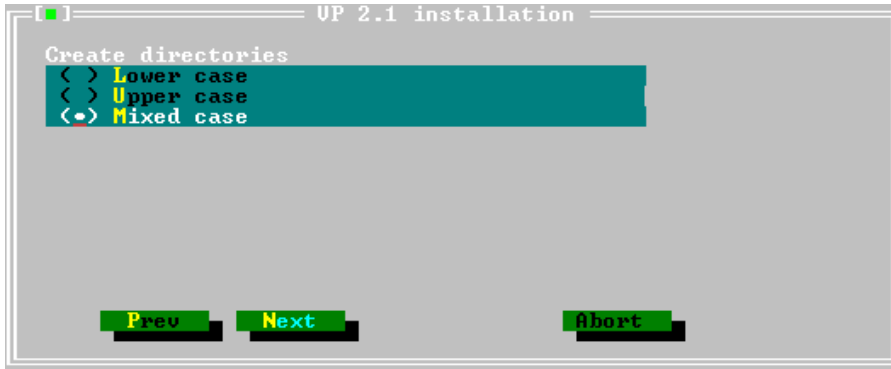
Quel que soit le type d'installation choisi, il faudra choisir sur quel disque Virtual Pascal doit être installé :



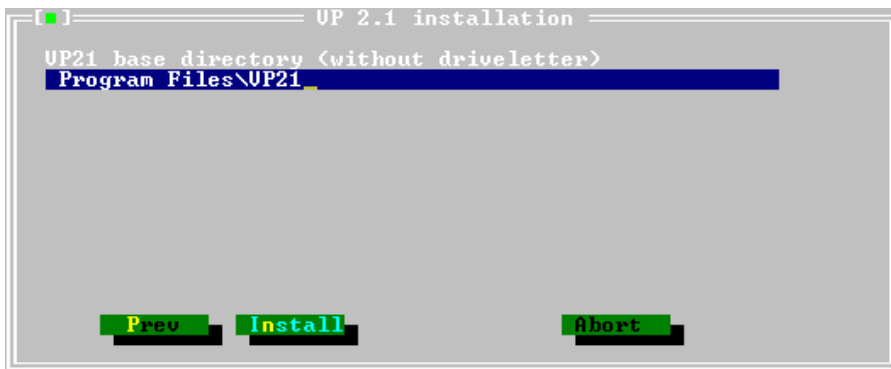
Cliquez sur **Next**.

Dans le dialogue suivant, vous pouvez décider de la manière dont Virtual Pascal va gérer la casse des noms de répertoires lors de l'installation :

- tout en minuscules (**Lower case**) ;
- tout en majuscules (**Upper case**) ;
- majuscules et minuscules mélangées (**Mixed case**).



Sous Windows, la casse des noms de fichiers et de répertoires n'a pas une importance aussi cruciale que sous Linux ; c'est plus une question de goût qu'une nécessité technique. Faites votre choix puis cliquez sur **Next**. C'est le moment de choisir dans quel répertoire Virtual Pascal doit être installé. Contrairement à Free Pascal, il n'a aucun problème avec les noms longs de fichiers ; vous pouvez donc franchement choisir un répertoire comme celui-ci :

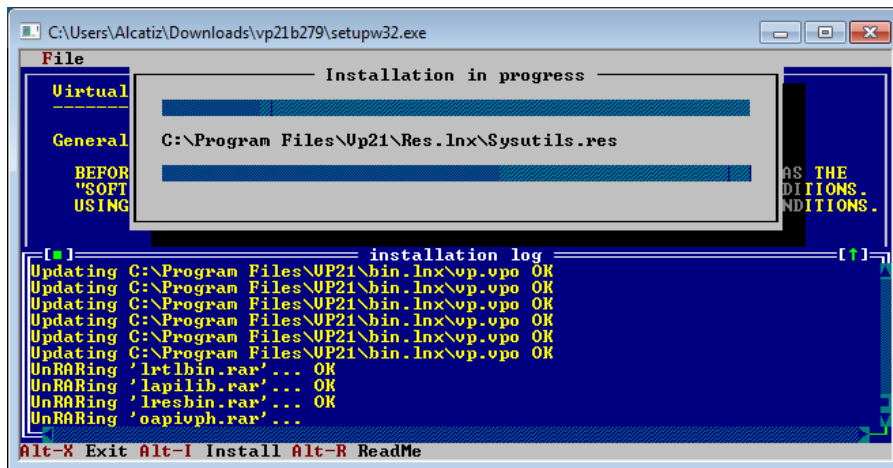


*Il ne faut pas indiquer la lettre du lecteur !*

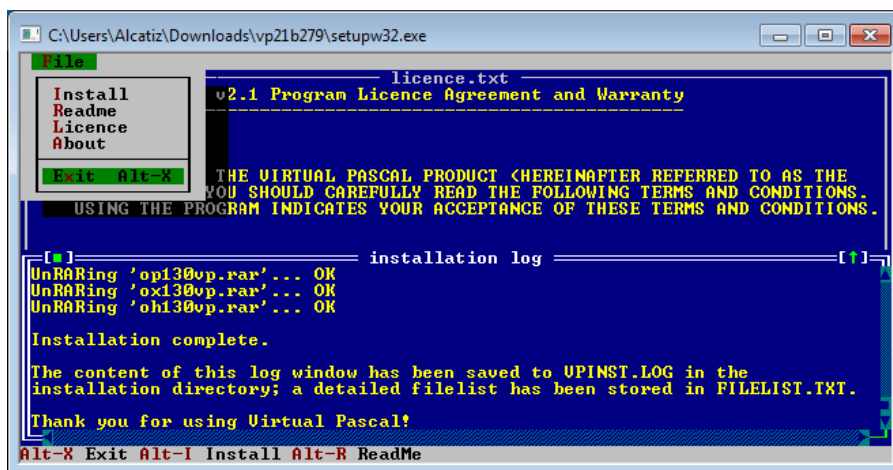
Si le répertoire choisi n'existe pas, le programme d'installation vous demande de confirmer sa création :



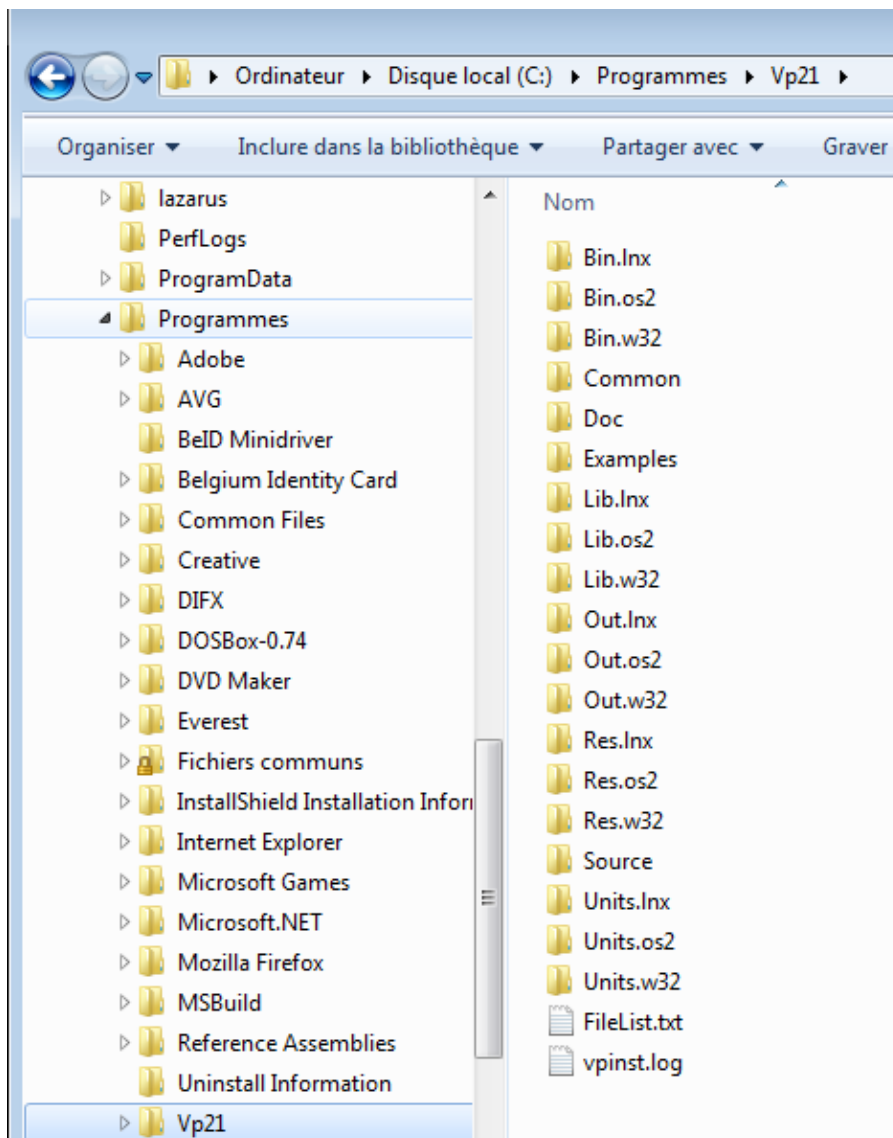
Répondez bien évidemment **Yes** !  
L'installation se déroule sous vos yeux :



Au bas de la fenêtre, vous devez voir apparaître le message salvateur : **Installation complete**. Pour quitter le programme d'installation, cliquez sur **File** puis **Exit** :



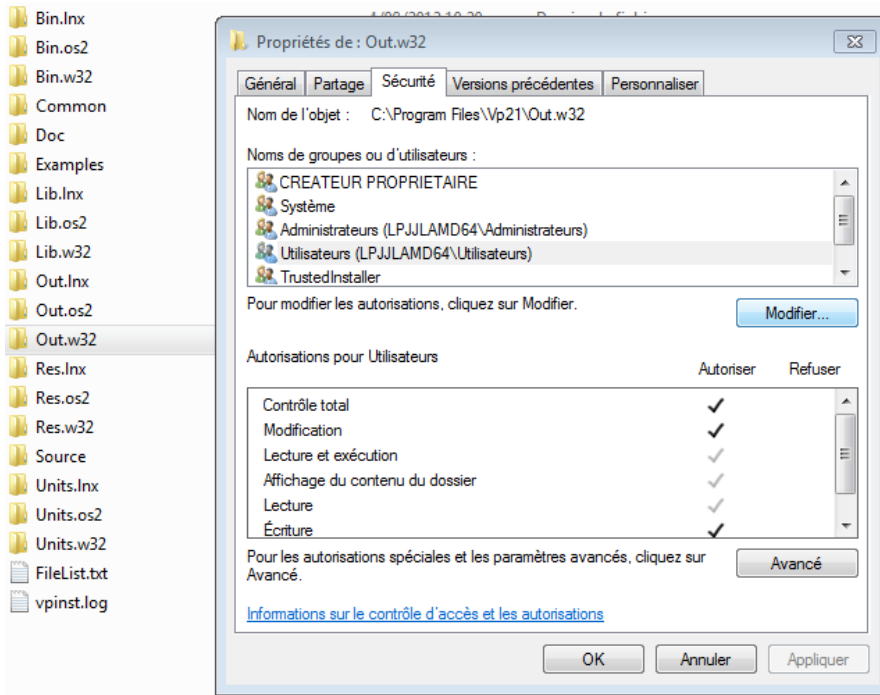
Et voici Virtual Pascal dans l'explorateur de fichiers :



## Configuration

Tant que nous sommes dans l'explorateur, profitons-en pour aller faire une modification de droits sur les sous-répertoires **Bin.w32** et **Out.w32**. En effet, pour des raisons de sécurité, Windows 7 bloque l'accès en écriture à ces répertoires pour les utilisateurs s'ils se trouvent sur le disque C. Si nous ne faisons rien, il sera impossible d'enregistrer la configuration de Virtual Pascal et, plus grave encore, le compilateur ne pourra pas enregistrer les fichiers qu'il générera.

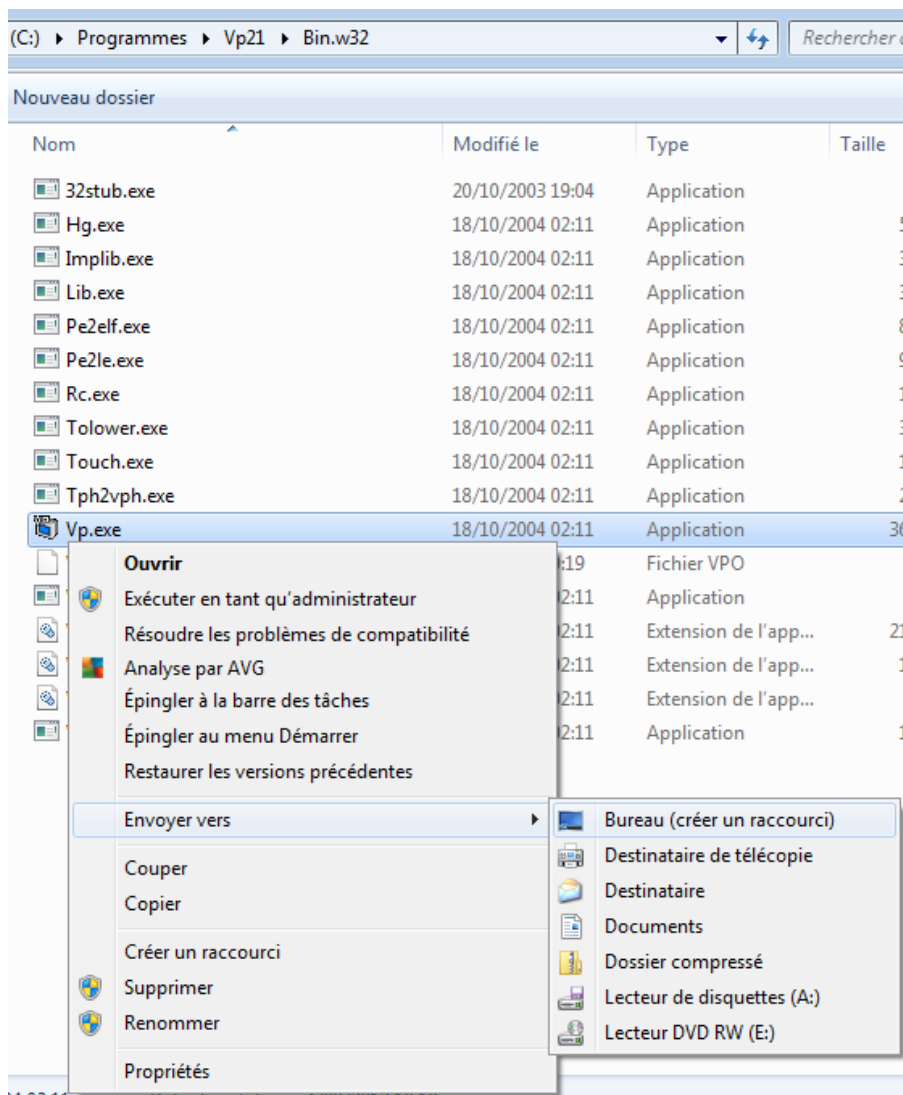
Faites un clic droit sur le sous-répertoire et choisissez **Propriétés** dans le menu contextuel (ici, pour **Out.w32**) :



Allez dans l'onglet **Sécurité**, sélectionnez le groupe **Utilisateurs** dans le cadre supérieur, cliquez sur **Modifier** et allez cocher les autorisations dans le cadre inférieur. Cliquez sur **Appliquer** et **OK**.

Nous allons ensuite créer sur le bureau un raccourci vers l'éditeur. Ce faisant, nous pourrons aller modifier certaines options permettant de configurer Virtual Pascal pour une utilisation plus confortable.

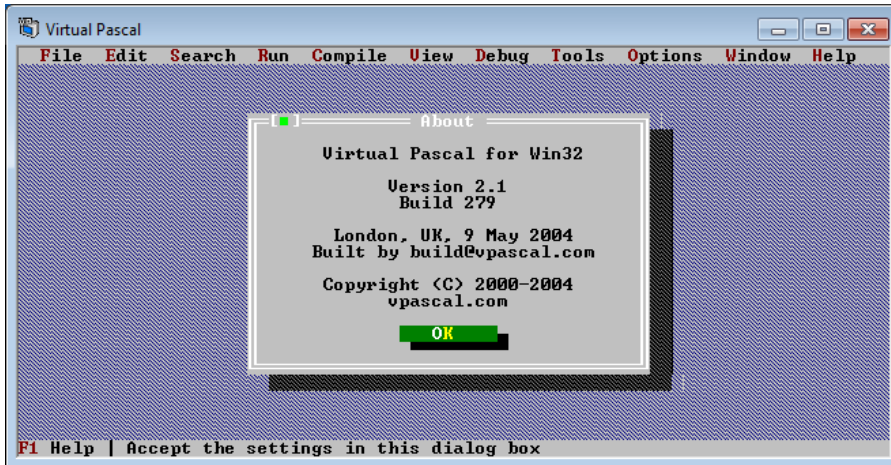
Allez dans le sous-répertoire **bin.w32** du répertoire dans lequel Virtual Pascal a été installé. Faites un clic droit sur le fichier **Vp.exe**, choisissez **Envoyer vers** puis **Bureau (créer un raccourci)** :



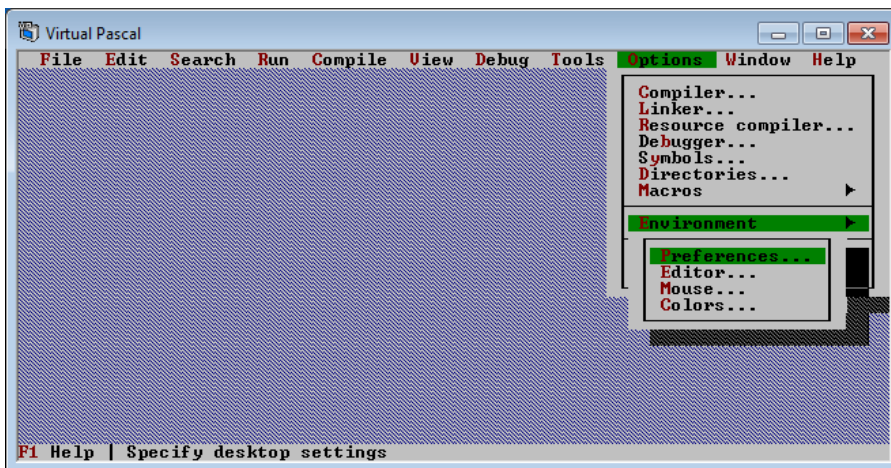
Et voici le raccourci :



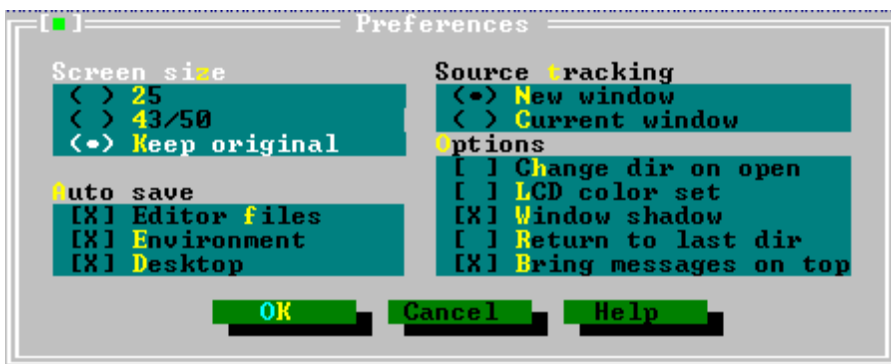
Double-cliquez dessus pour lancer Virtual Pascal pour la toute première fois :



Lors de cette première exécution, nous allons simplement fixer une option qui nous permettra d'augmenter le nombre de lignes et de colonnes de l'éditeur. Cliquez sur le menu **Options/Environment/Preferences** :

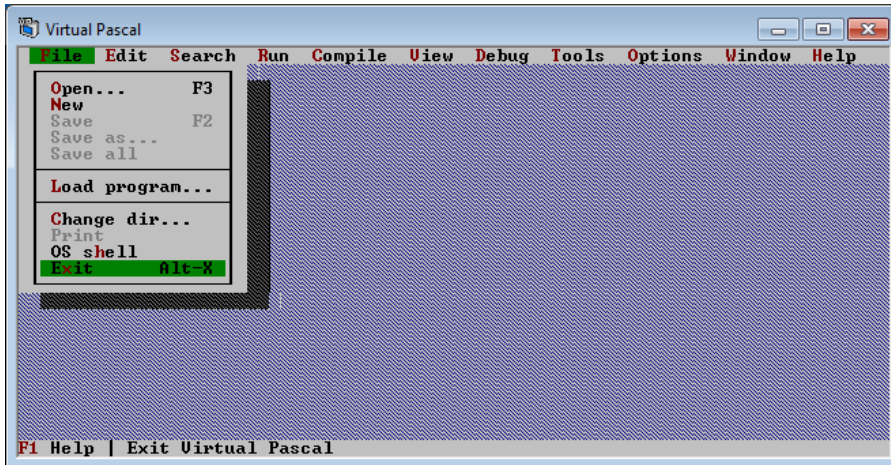


Dans le dialogue qui apparaît, dans le premier cadre, cochez l'option **Keep original** puis pressez **OK** :

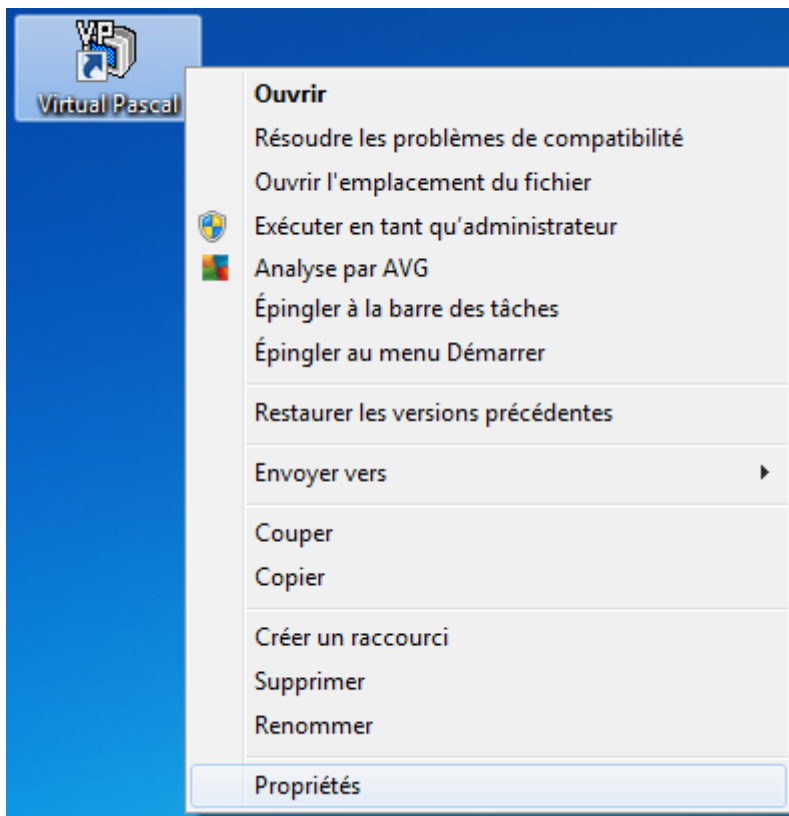


Quittez Virtual Pascal, en cliquant sur **File/Exit** :



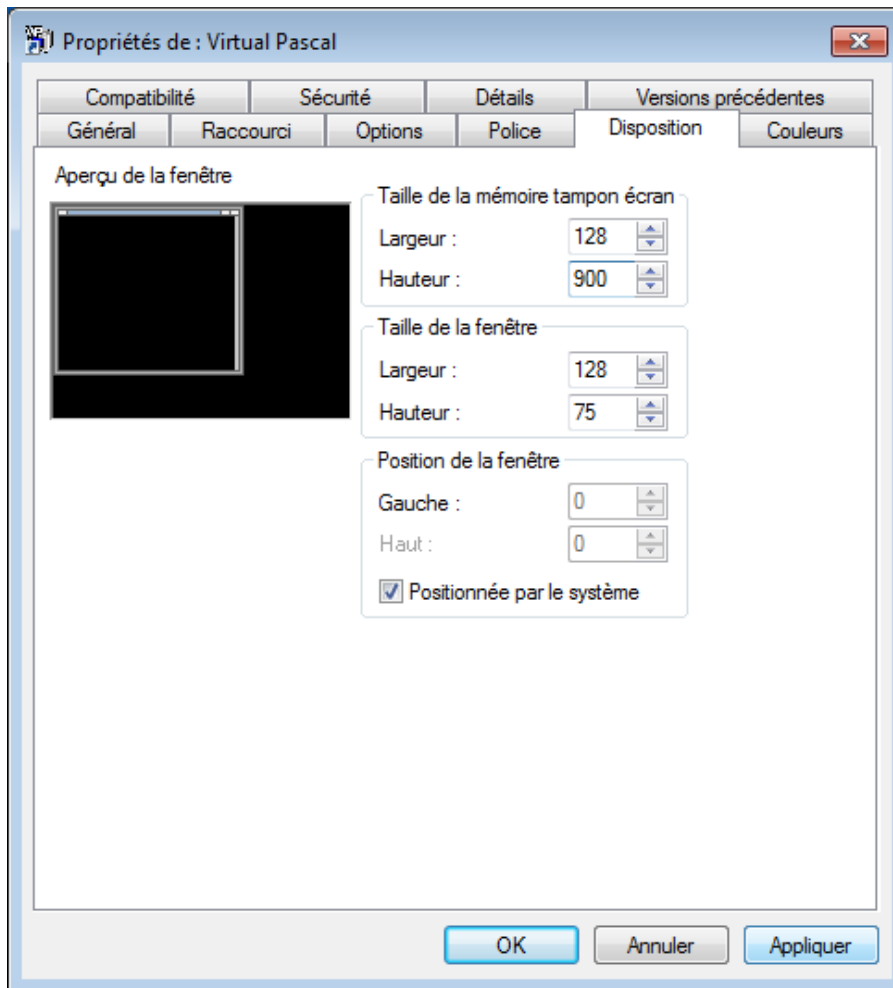


Faites un clic droit sur le raccourci de Virtual Pascal sur le bureau puis, dans le menu contextuel, choisissez **Propriétés** :



Sélectionnez l'onglet **Disposition** :





Vous pouvez augmenter la largeur et la hauteur dans le cadre **Taille de la fenêtre**. Ces valeurs sont en nombre de lignes ou de colonnes.

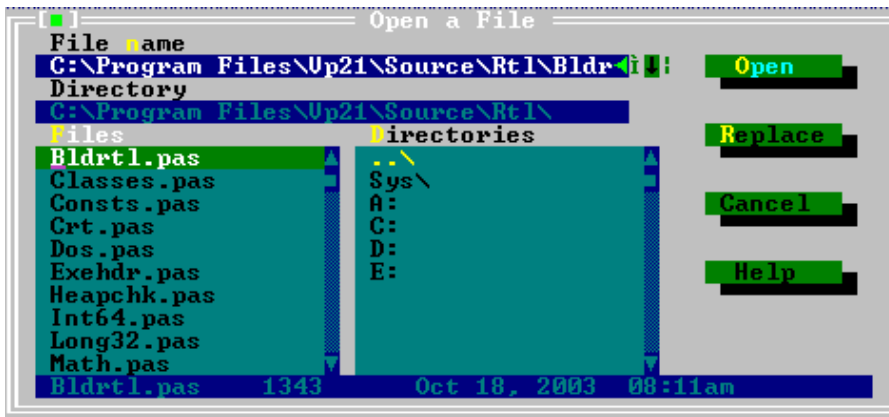


**Veillez également à ce que les valeurs de largeur et de hauteur du cadre **Taille de la mémoire tampon écran** soient au moins égales aux valeurs que vous avez fixées.**

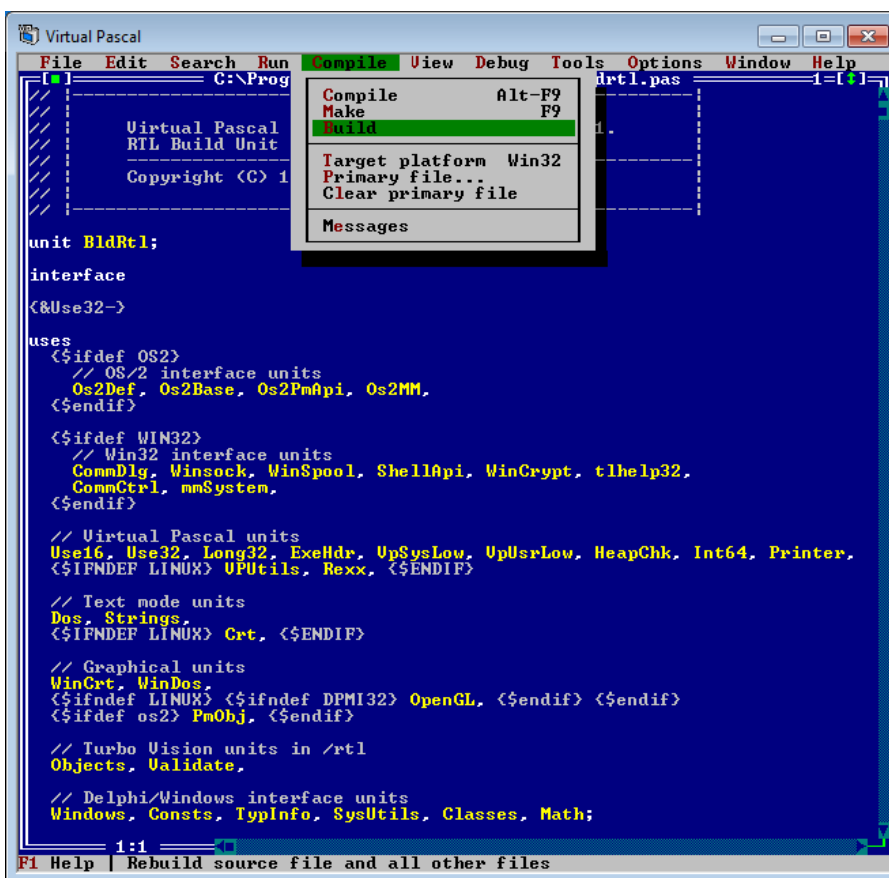
Appliquez les modifications et relancez Virtual Pascal : le nombre de lignes et de colonnes de l'éditeur est modifié selon vos *desiderata* et, grâce à l'option **Keep original** cochée auparavant, ces changements sont permanents (sauf bien sûr si vous effectuez d'autres modifications dans les options du raccourci).

## Compilation de la RTL

Avant de pouvoir compiler votre premier programme, vous devez compiler la *Run-Time Library* (RTL). Virtual Pascal va ainsi créer toutes ses bibliothèques et unités de base. Cette opération ne doit être faite qu'une seule fois. Cliquez sur **File/Open** et naviguez dans l'arborescence jusqu'au sous-répertoire **Source/Rtl** :



Choisissez le fichier **Bldrtl.pas** (pour « *build RTL* », c'est-à-dire « *construire la RTL* ») puis cliquez sur **Open**. Lorsque le fichier est chargé dans l'éditeur, cliquez sur **Compile/Build** :

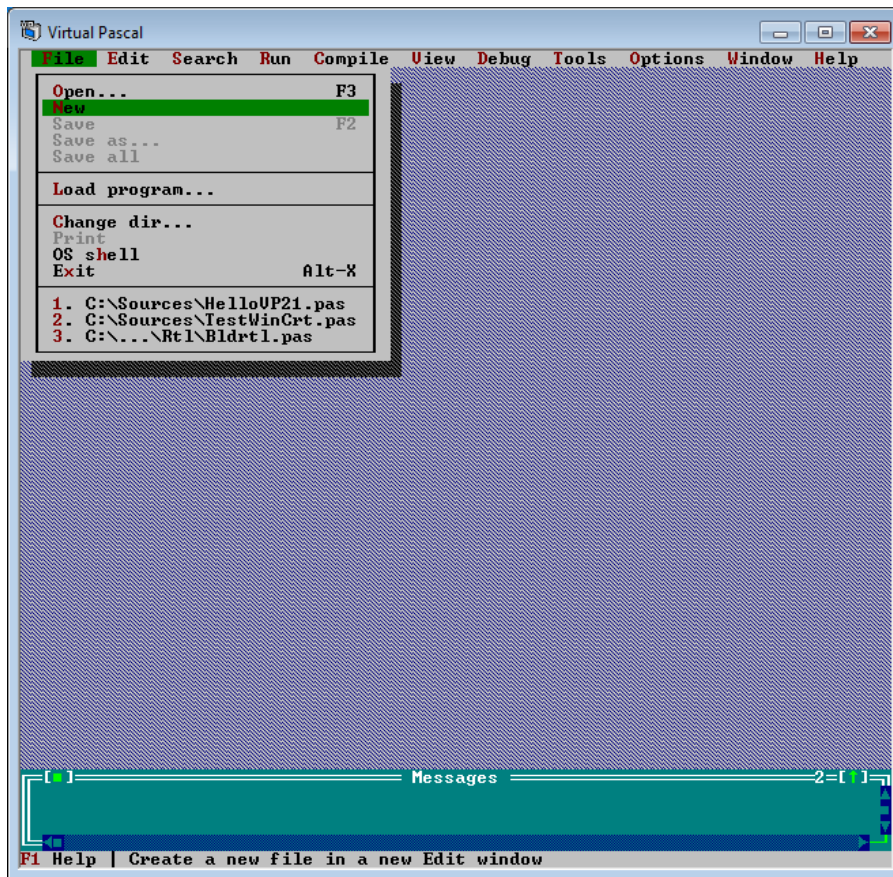


Si vous avez suivi toutes les étapes d'installation, la compilation devrait être couronnée de succès et vous devriez voir apparaître dans la fenêtre de messages :



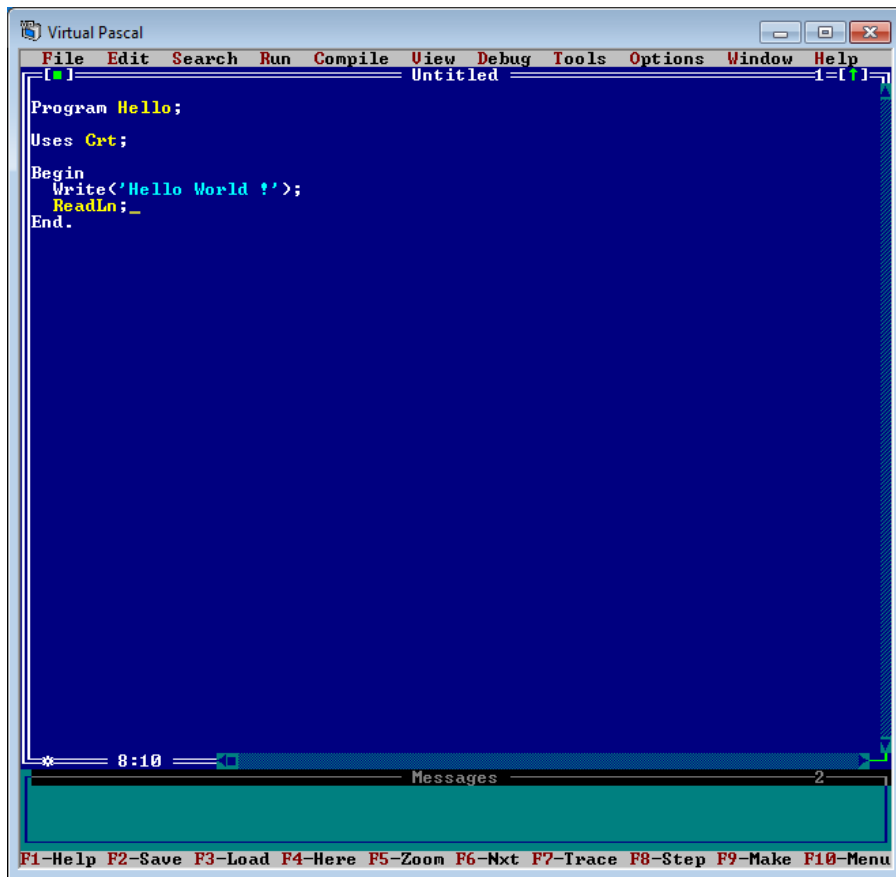
## Compilation d'un programme console

Créez un nouveau fichier source en cliquant sur **File/New** :

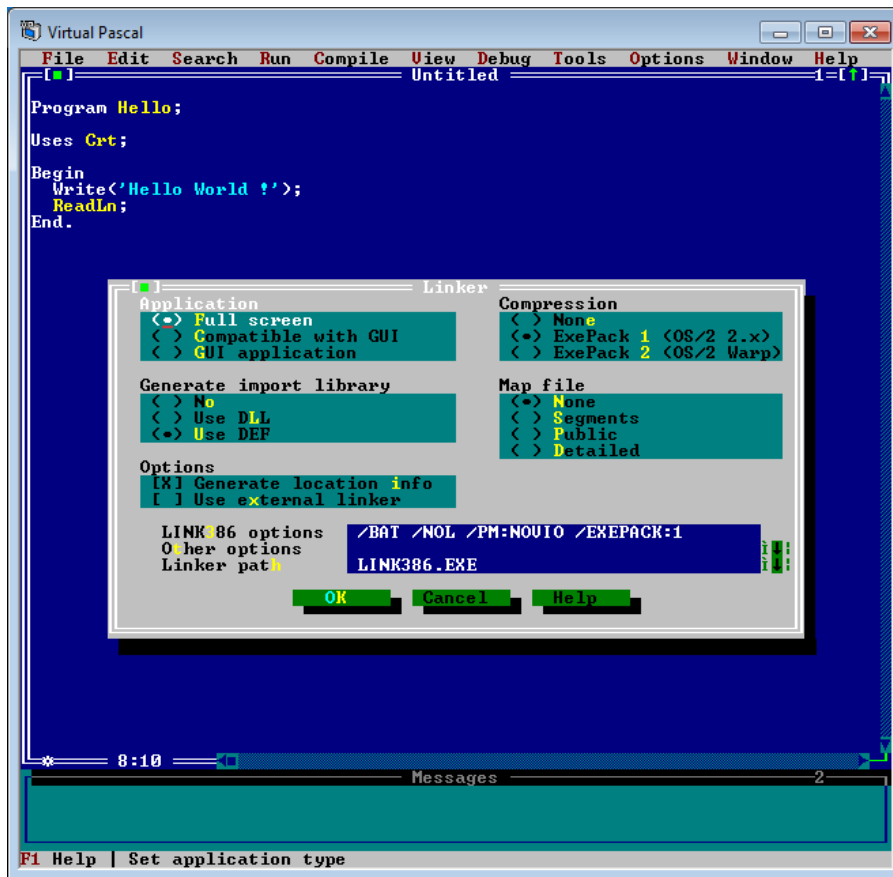


Saisissez le code source suivant, d'une redoutable puissance :

```
Program Hello;  
  
Uses Crt;  
  
Begin  
  Write('Hello World !');  
  ReadLn;  
End.
```



Comme il s'agit d'un programme destiné à tourner sur la console, il faut l'indiquer à Virtual Pascal. Cliquez sur le menu **Options/Linker** :



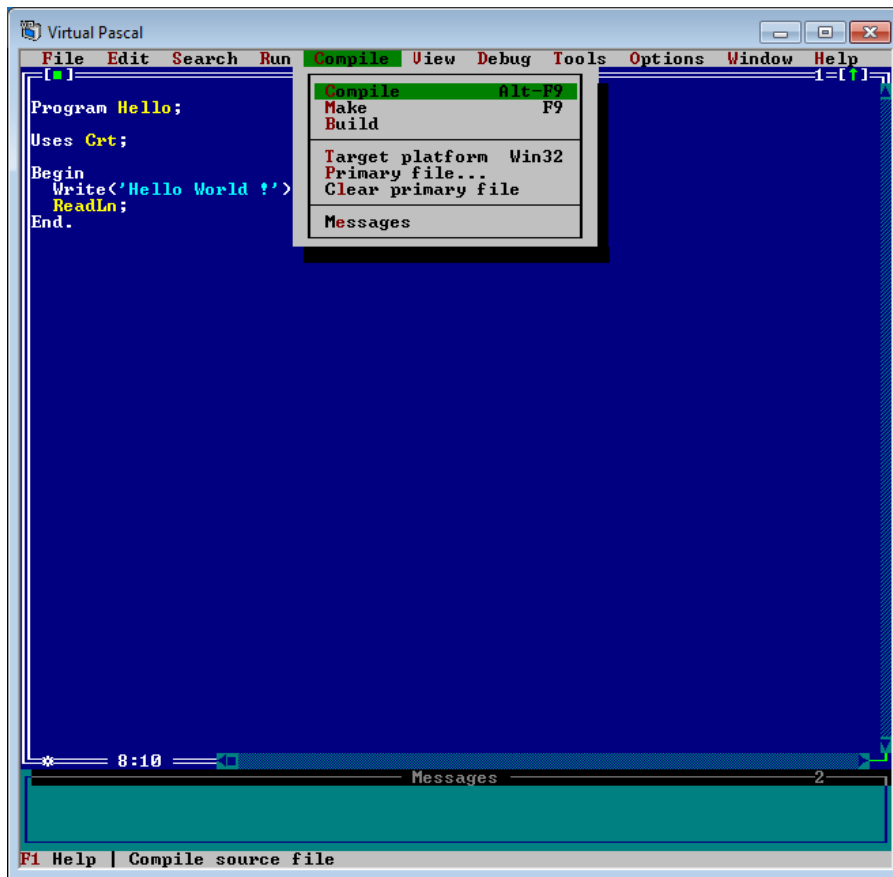
Le premier cadre des options de l'éditeur de liens vous permet de déterminer de quel type d'application il s'agit :

- une application console (**Full screen**) ;
- une application compatible avec le système graphique (**Compatible with GUI**) ;
- une application graphique (**GUI application**).

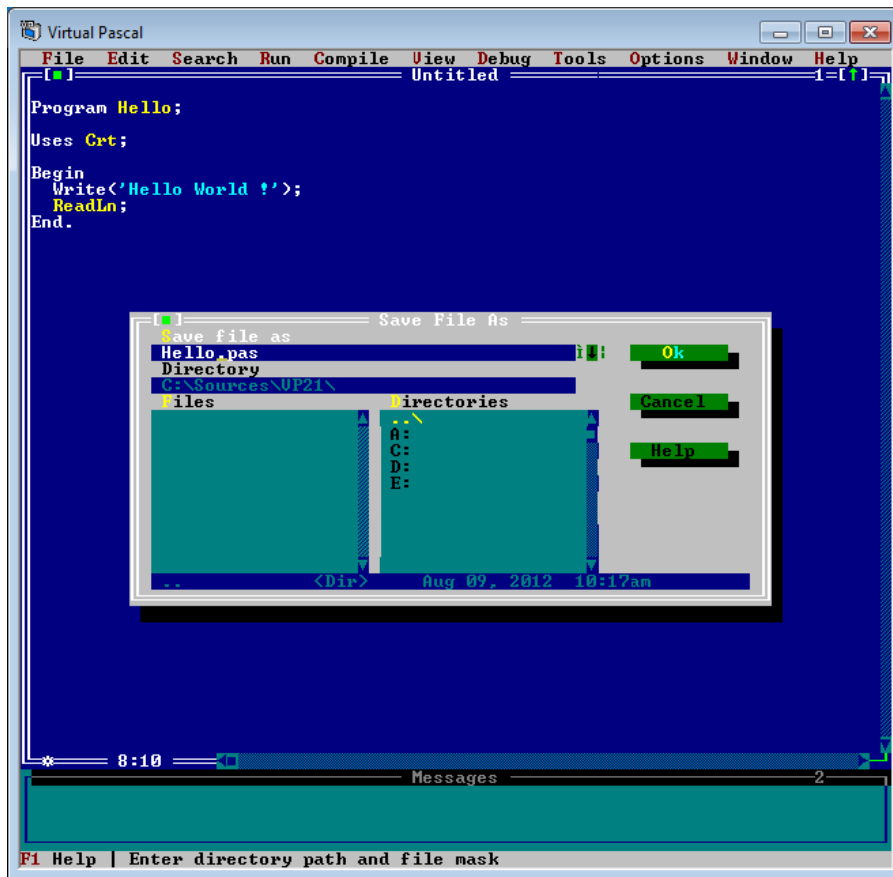
**i** GUI signifie *Graphical User Interface* .

C'est la première option qu'il faut choisir ici. Cliquez sur **OK** pour valider.

Lancez maintenant la compilation du programme, en cliquant sur **Compile/Compile** :

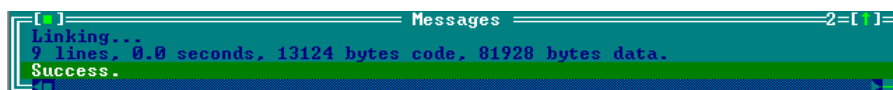


Avant de pouvoir compiler, Virtual Pascal vous demande d'enregistrer votre fichier source. Naviguez dans l'arborescence et enregistrez le fichier (appelé **Hello.pas** pour cet exemple) dans le répertoire de votre choix. Notez que le navigateur de Virtual Pascal ne permet pas de créer de répertoire ; il faut utiliser l'explorateur de fichiers de Windows pour ce faire.

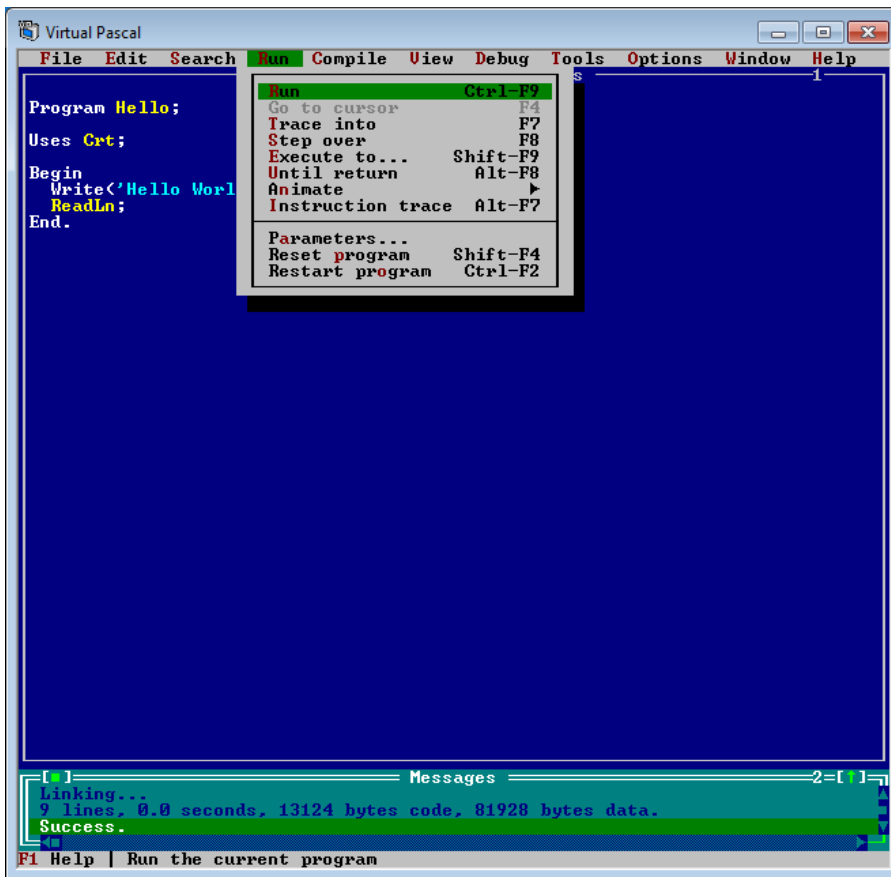


*Évitez de mélanger vos codes sources avec ceux de Virtual Pascal.*

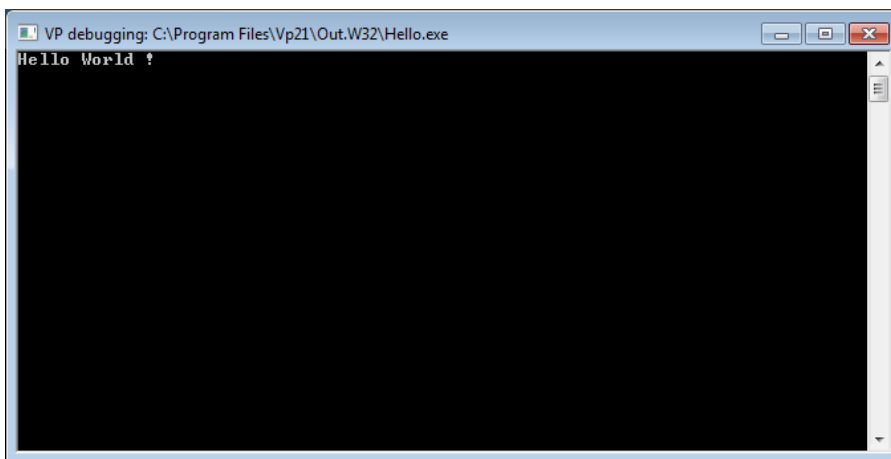
Normalement, la fenêtre de messages doit vous annoncer que la compilation est correcte :



Lancez l'exécution du programme en cliquant sur le menu **Run/Run** :



Et le programme se lance dans une console :



Le programme attend que vous pressiez la touche **Entrée** pour se fermer. Une fois que la console disparaît, Virtual Pascal vous affiche le résultat de l'exécution :







Vous pouvez bien évidemment lancer vos programmes en dehors de l'environnement de développement. Pour les trouver après compilation, c'est très simple : ils se trouvent systématiquement dans le sous-répertoire **Out.w32** .

## Compilation d'une application GUI

Nous allons à présent compiler une application utilisant l'interface graphique de Windows.



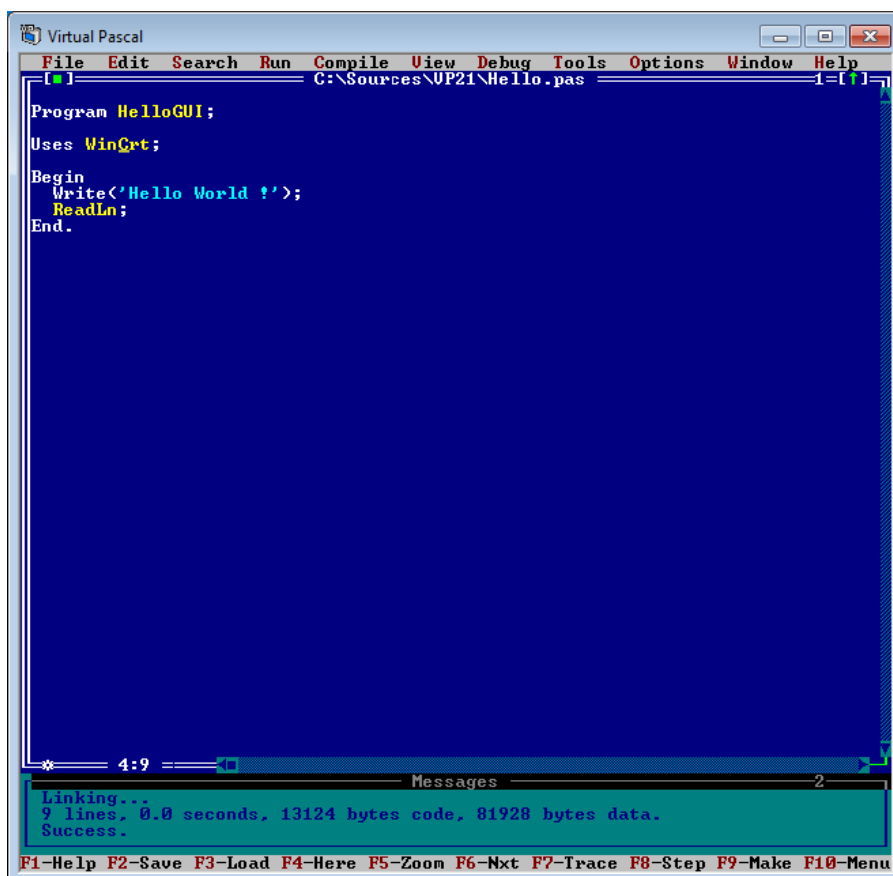
Virtual Pascal ne possède pas d'unité **Graph** . Pour faire du graphisme, il faut utiliser les fonctions de Windows (fonctions **GDI** ) dans une application GUI.

Nous n'allons pas mettre plus de dix secondes pour réaliser notre application GUI. Dans le code source du programme **Hello.pas**, remplacez la déclaration de l'unité **Crt** par **WinCrt** (et, accessoirement, changez le nom du programme en **HelloGUI**) :

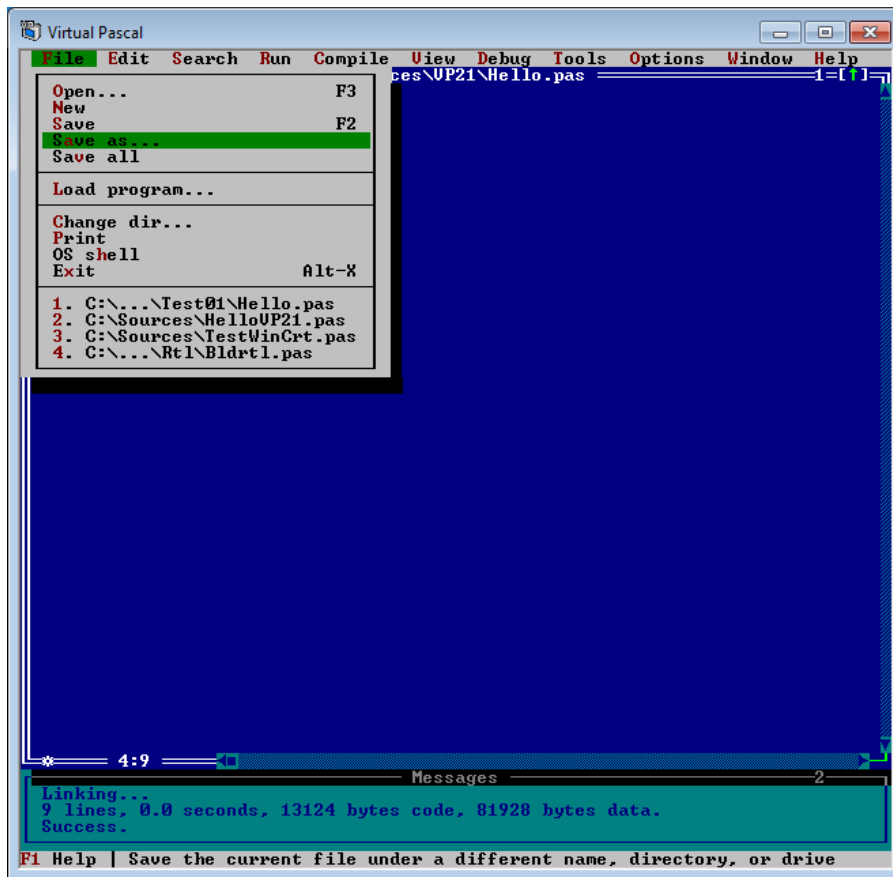
```
Program HelloGUI;

Uses WinCrt;

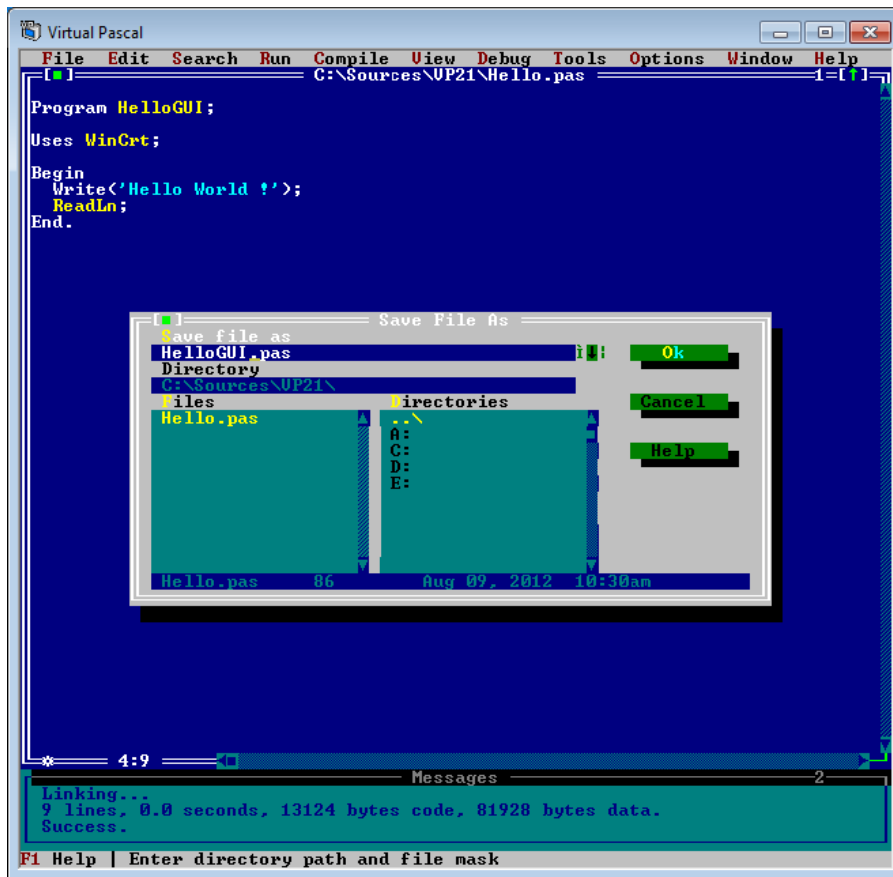
Begin
  Write('Hello World !');
  ReadLn;
End.
```



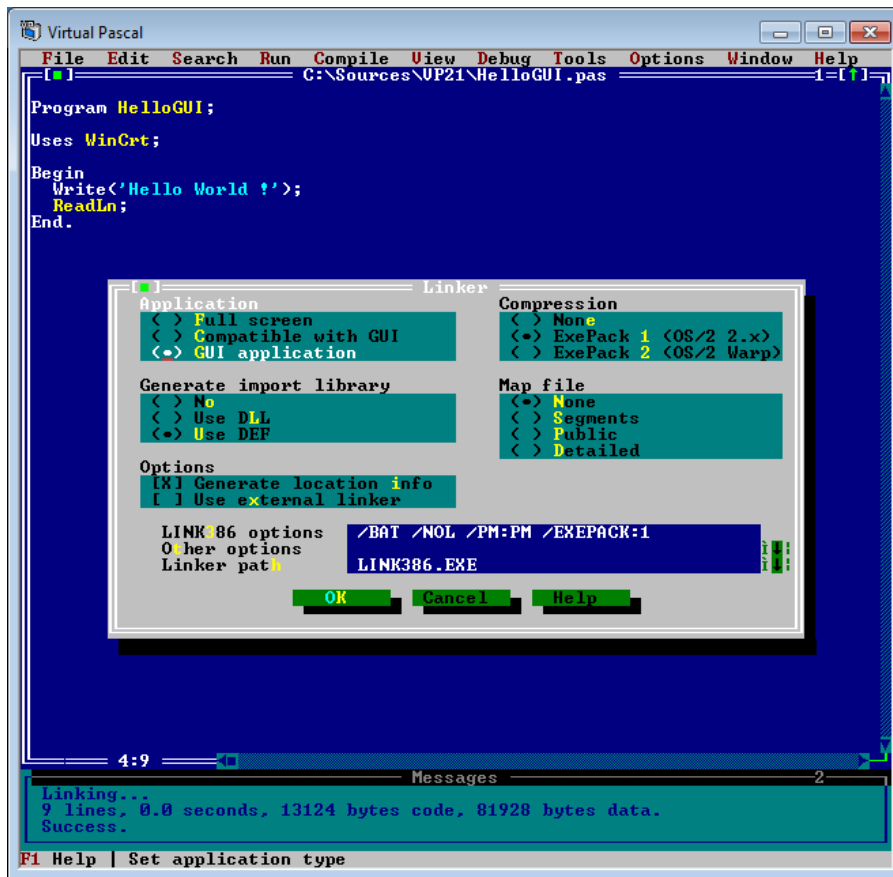
Comment ces infimes modifications ont-elles fait du programme console initial une application GUI ? L'unité **WinCrt** applique une partie des fonctionnalités de l'unité **Crt** à une véritable fenêtre Windows, qui émule la console. Nous allons enregistrer cette nouvelle application. Cliquez sur le menu **File/Save as...** :



Donnez-lui un nouveau nom par exemple, **HelloGUI.pas** :

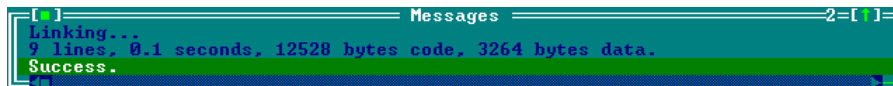


Avant de compiler, il faut aller indiquer à l'éditeur de liens qu'il s'agit d'une application GUI. Comme précédemment, cliquez sur le menu **Options/Linker** :

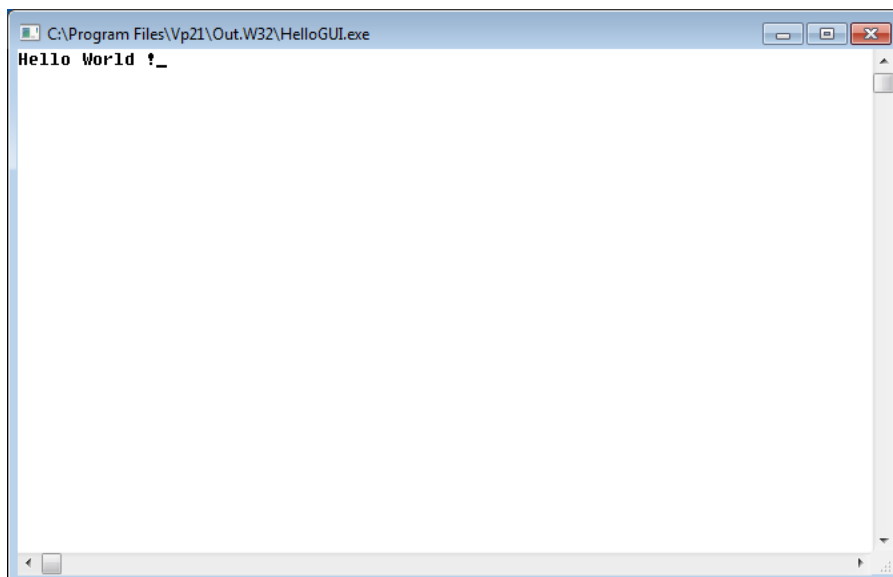


Cochez cette fois l'option **GUI Application** puis cliquez sur **OK**.

Cliquez sur **Compile/Compile** ; le résultat de la compilation s'affiche dans la fenêtre des messages :

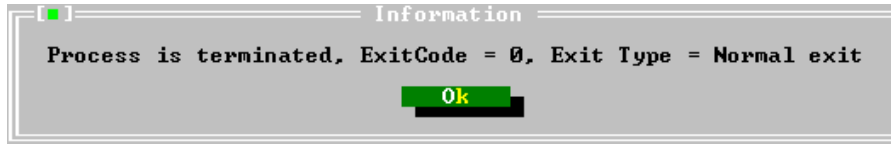


Exécutez l'application via le menu **Run/Run**. La fenêtre qui apparaît a un air de famille avec la console mais il s'agit bien d'une véritable fenêtre Windows :



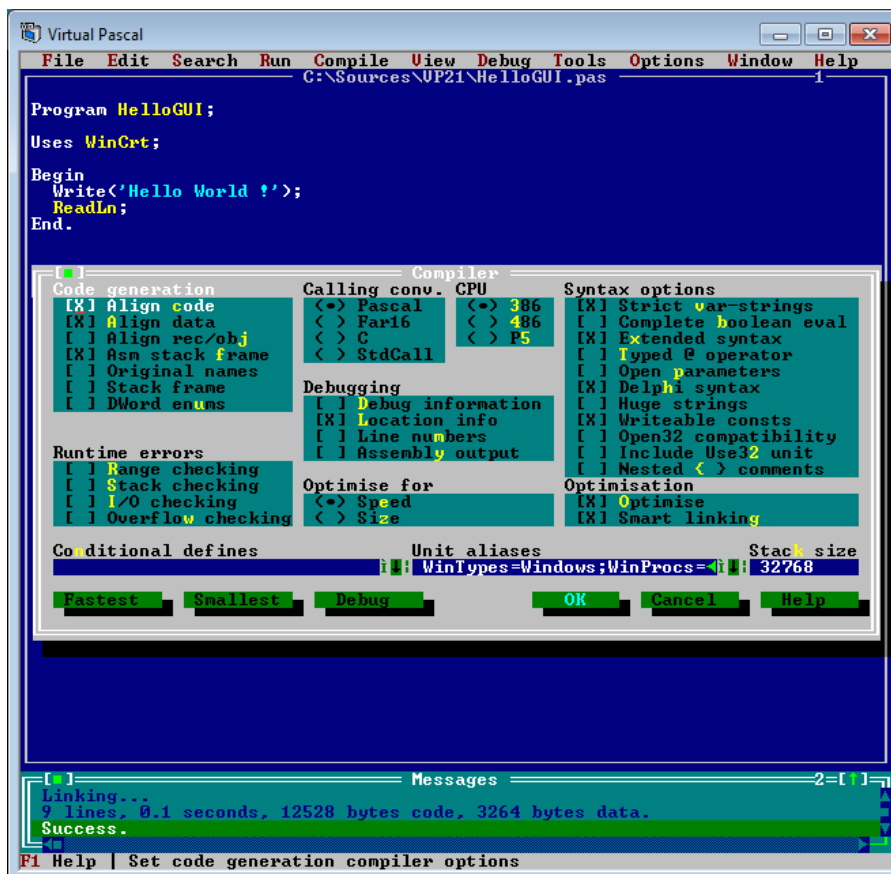
Pressez **Entrée** pour mettre fin au programme. Vous constatez que cette fois, la fenêtre ne se ferme pas toute seule et que son titre a été modifié en "*Inactive...*". C'est tout à fait normal ; il existe des fonctionnalités de l'unité WinCrt permettant de modifier ce comportement mais ce n'est pas du tout l'objet de ce tutoriel. Fermez donc la fenêtre à la souris.

Virtual Pascal affiche de nouveau le résultat de l'exécution :



## Options du compilateur

Nous avons vu comment aller choisir le type d'application dans les options de l'éditeur de liens. Il existe également trois options relativement importantes du compilateur. Cliquez sur le menu **Options/Compiler** :



Les options en question correspondent aux trois boutons en bas à gauche :

- **Fastest** : pour optimiser la vitesse de l'exécutable ;
- **Smallest** : pour obtenir un code plus compact ;
- **Debug** : pour inclure dans le code les informations nécessaires au débogage.

Vous voyez qu'en cliquant sur ces trois boutons, vous faites varier plusieurs autres options de la boîte de dialogue. C'est très pratique. Lorsque vous en êtes à la phase de conception d'un programme, cliquez sur le bouton **Debug** afin de pouvoir utiliser le débogueur interne de Virtual Pascal. Et lorsque vous produisez votre code final, cliquez sur l'un des deux autres boutons.



*De nos jours, les capacités de stockage et de mémoire vive sont devenues énormes ; la recherche du code le plus compact possible n'a plus tellement de sens et il est très probable que vous n'utiliserez que le bouton **Fastest** .*

## Installation de Geany

Comme cela a été mentionné dans l'introduction, l'éditeur natif de Virtual Pascal était très pratique jusqu'à Windows XP ; il n'y avait nul besoin d'utiliser un éditeur alternatif. Windows Vista et Windows 7 le rendent malheureusement pénible à utiliser.

Ce tutoriel vous présente l'éditeur généraliste **Geany**. Ses atouts sont sa légèreté, sa souplesse de configuration, la facilité avec laquelle on peut lui lier un compilateur quelconque et le fait qu'il soit multiplateforme, tout comme Virtual Pascal. D'autres éditeurs conviendraient sans doute ; les commandes de compilation présentées pour Geany peuvent vous être utiles pour y intégrer Virtual Pascal.

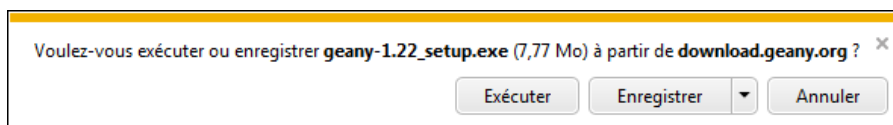
## Téléchargement

**Geany** peut être téléchargé sur [ce site](#), dans la section **Windows Binaries**.

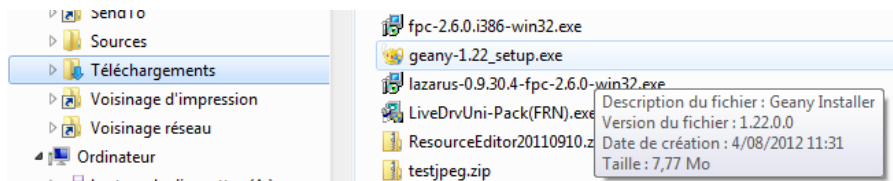
Pour fonctionner, il requiert les bibliothèques de **GTK+**. Si celles-ci sont déjà présentes sur votre système, vous pouvez vous contenter de télécharger l'installateur **geany-x.xx\_nogtk\_setup.exe** ; dans le cas contraire, téléchargez **geany-x.xx\_setup.exe** :



Lorsque vous cliquez sur l'un des deux installateurs, Windows vous demande ce que vous voulez en faire :



Répondez-lui que vous voulez enregistrer le fichier. Celui-ci se retrouve dans le dossier de téléchargements :



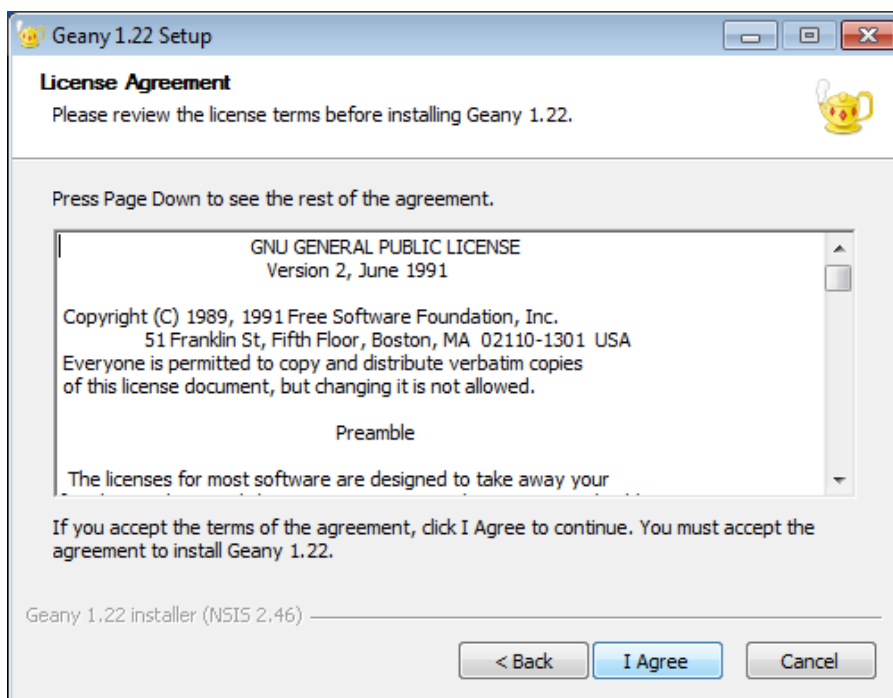
## Installation

Faites un double-clic sur l'installateur. Il vous souhaite à son tour la bienvenue :



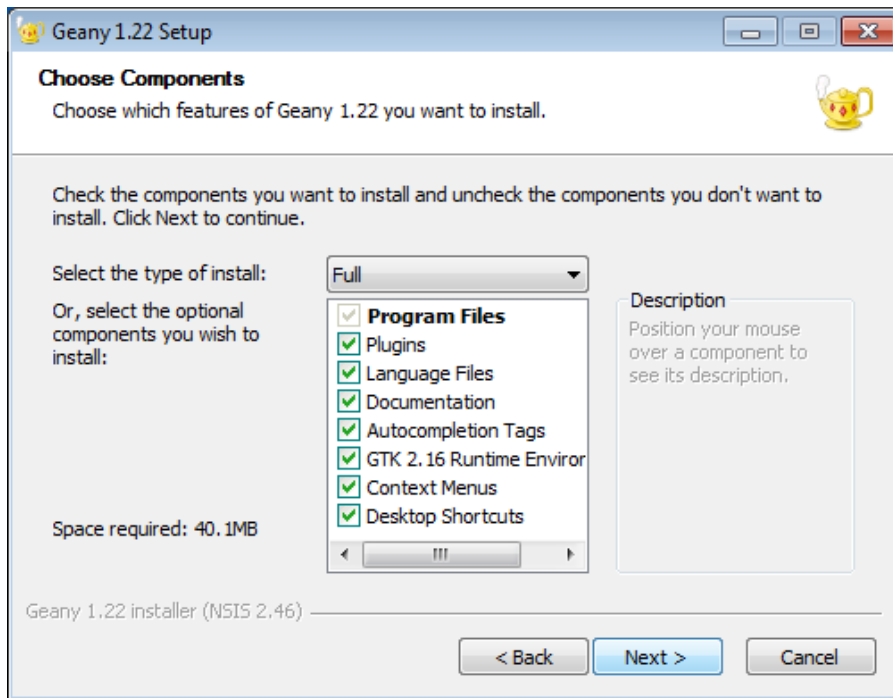
Cliquez sur **Next**.

Marquez votre accord avec la licence GNU/GPL en cliquant sur **I agree** :

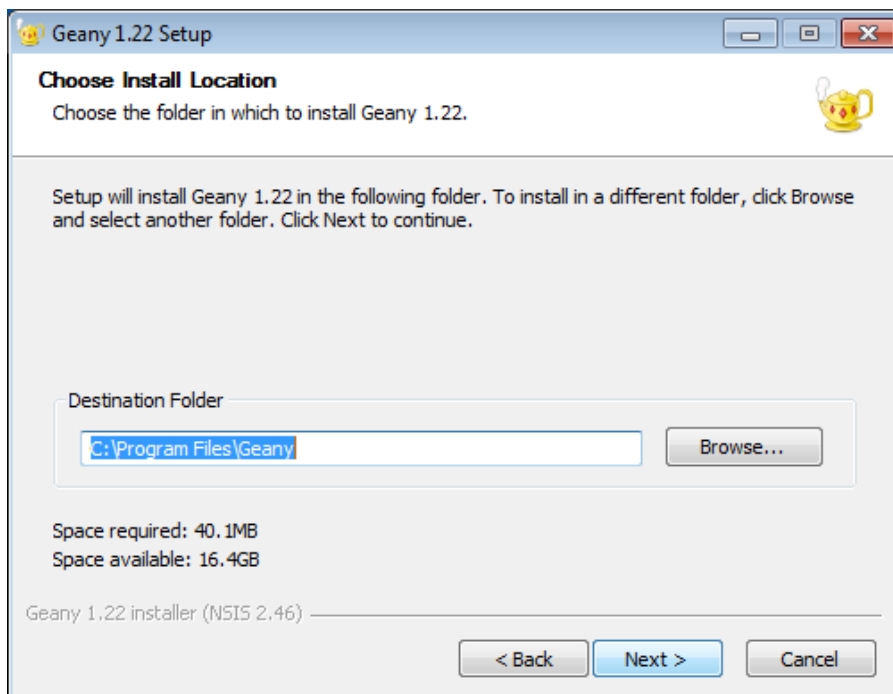


À présent, le programme d'installation vous propose de sélectionner les composants à installer :

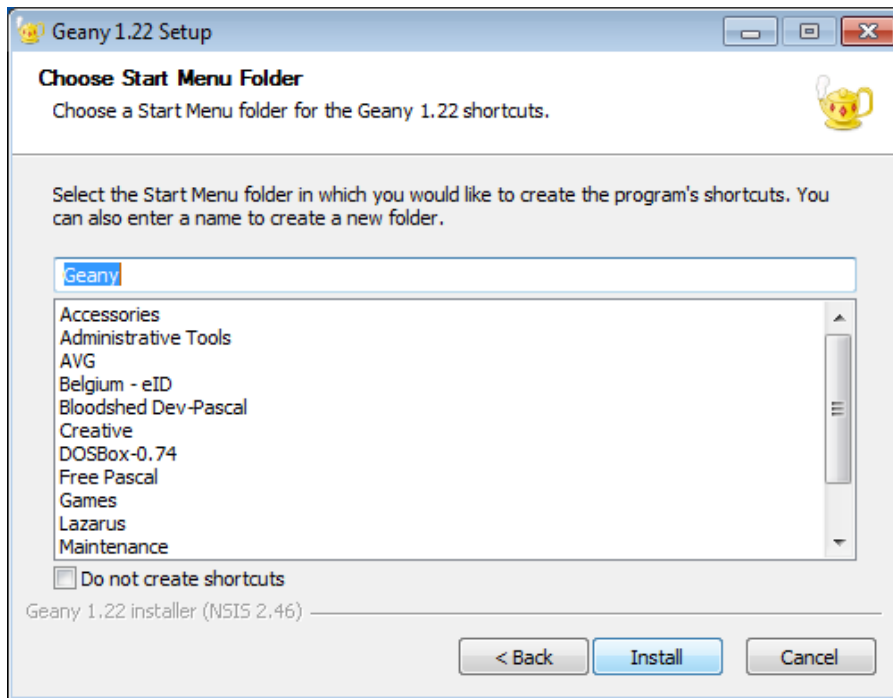




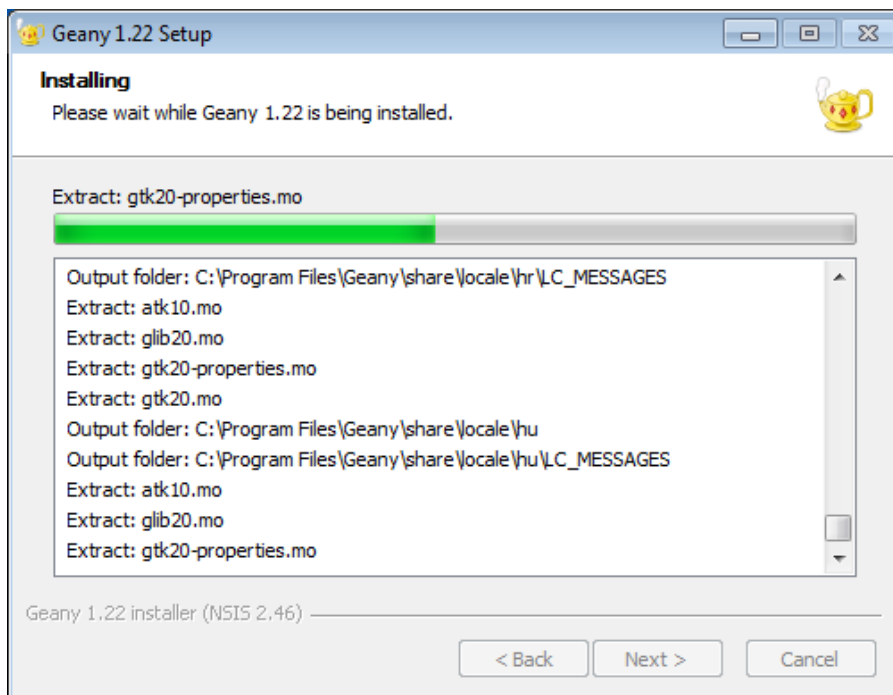
Vous pouvez laisser s'installer le paquetage complet (**Full**) ou bien faire une sélection plus fine des composants. Gardez seulement à l'esprit que **GTK+** doit être présent sur votre système. Notez également que si vous laissez s'installer le composant **Language Files**, la langue de votre système sera automatiquement activée dans l'éditeur, sans que vous ayez quoi que ce soit à faire. Une fois que c'est fait, cliquez sur **Next** :



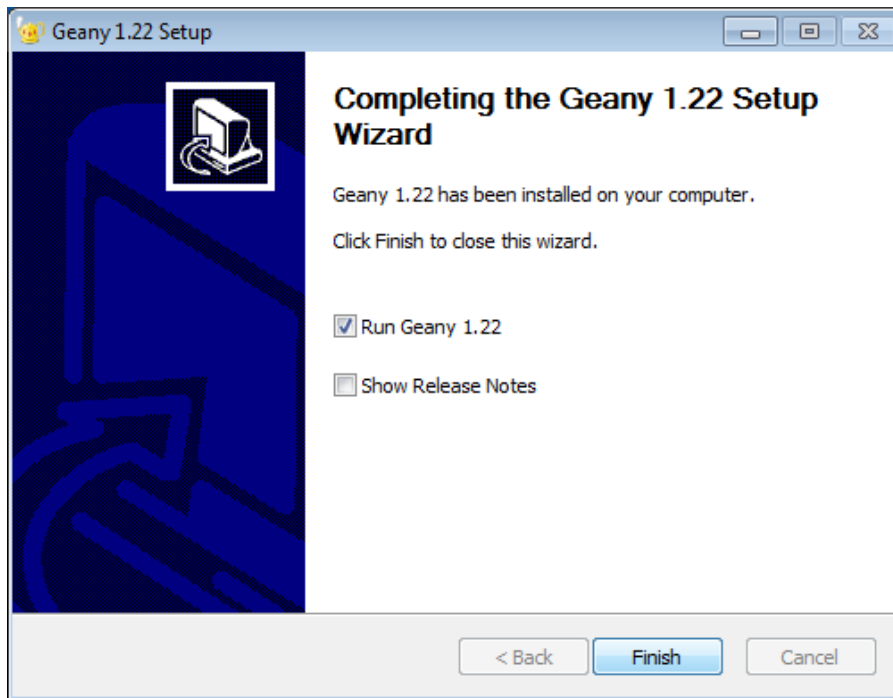
Indiquez dans quel répertoire vous souhaitez que Geany soit installé puis cliquez sur **Next** :



L'installateur vous propose enfin de créer une entrée pour Geany dans l'arborescence de votre menu de démarrage. Si vous ne le souhaitez pas, cochez la case **Do not create shortcuts**. Cliquez sur **Install** pour lancer l'installation de Geany :

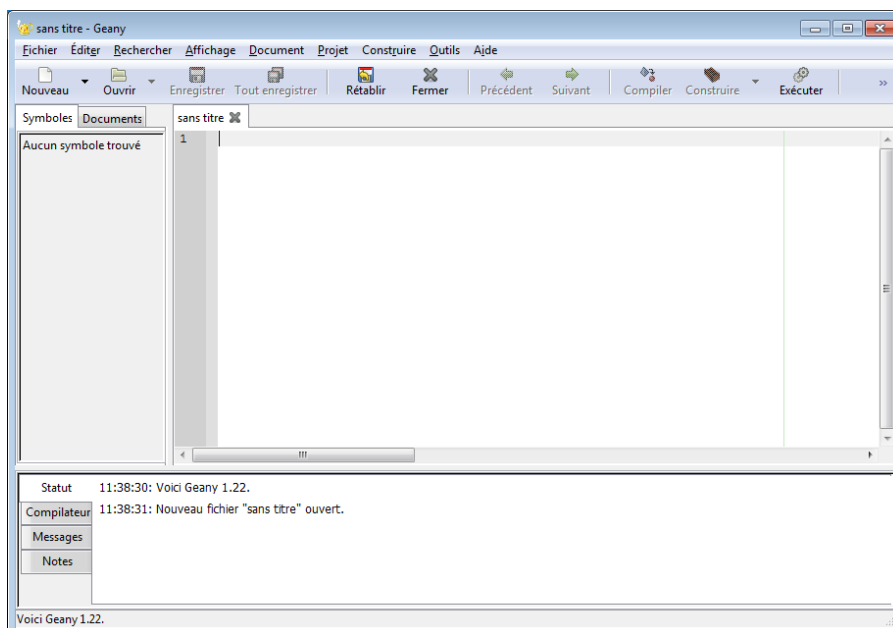


Une fois le processus d'installation terminé, cochez la case **Run Geany** puis cliquez sur **Finish** :



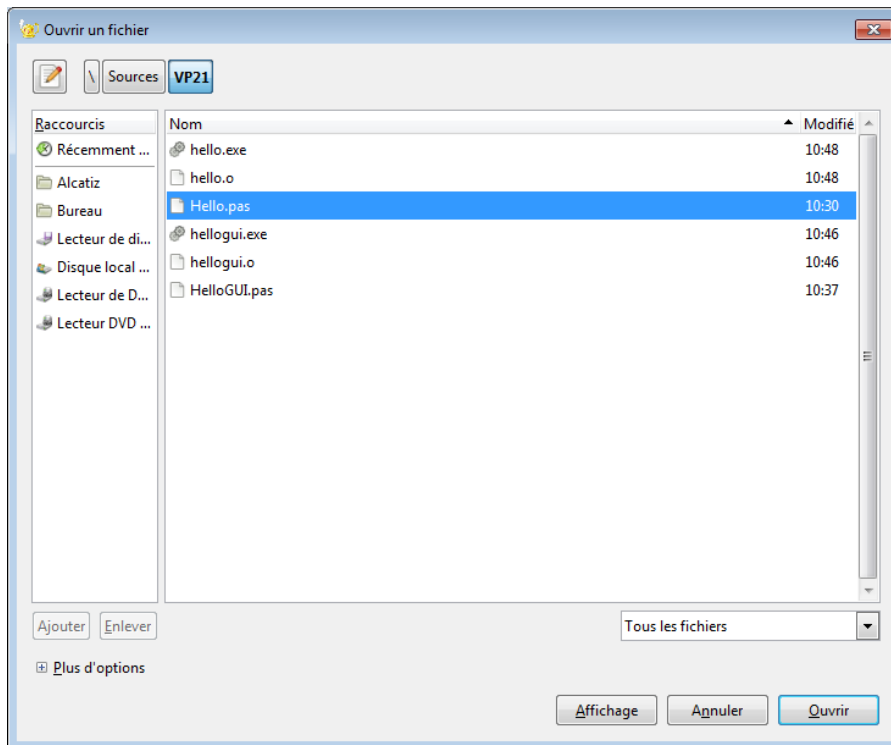
## Configuration

Geany s'exécute pour la première fois :



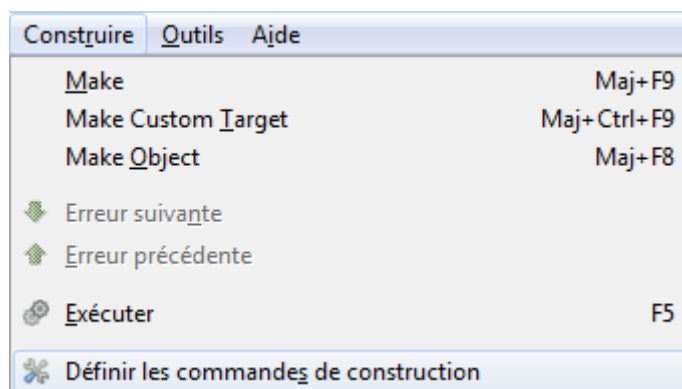
Le système Windows 7 utilisé lors de l'installation est configuré en français par défaut ; Geany s'ouvre donc directement en français !

Cliquez sur le bouton **Ouvrir** de la barre d'outils et allez sélectionner le fichier **Hello.pas** que nous avons enregistré précédemment. Cliquez sur **Ouvrir** :

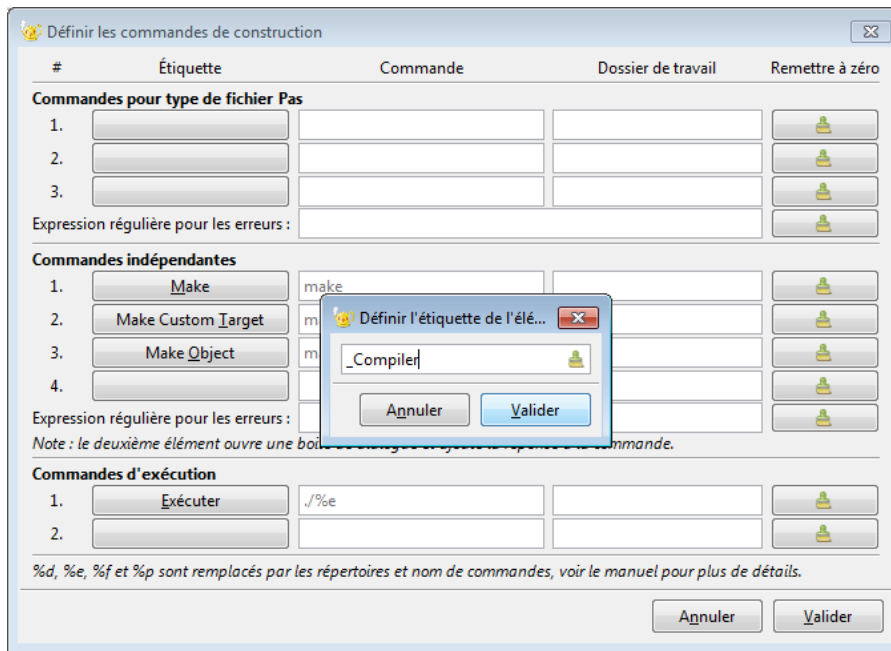


Nous allons essayer de compiler ce code source. Si **Free Pascal** est installé sur votre système, le fait de cliquer sur le bouton **Compiler** de la barre d'outils provoquera la compilation par Free Pascal sans avoir configuré quoi que ce soit ! Avouez que Geany est un éditeur particulièrement intelligent.

Mais c'est **Virtual Pascal** que nous voulons utiliser, pas Free Pascal ; il faut donc aller modifier la configuration par défaut de Geany. Cliquez sur le menu **Construire** puis sur **Définir les commandes de construction** :



Le dialogue qui apparaît permet de définir pas mal de commandes ; c'est le cadre supérieur qui nous intéresse. Cliquez sur l'étiquette n° 1, dans la section **Commandes pour fichiers de type Pas** :

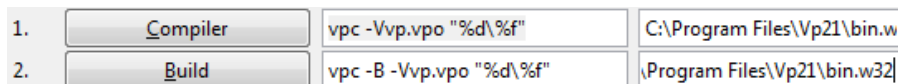


Écrivez-y le nom de la commande, en l'occurrence **Compiler**, puis cliquez sur **Valider**.

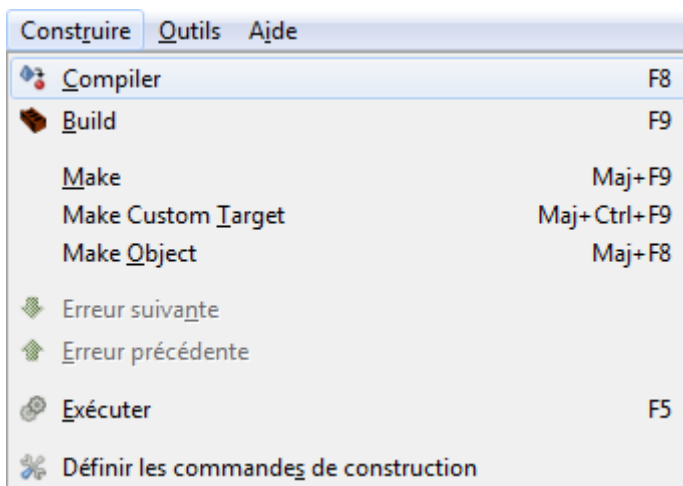
**i** *Si vous voulez assigner un raccourci clavier à votre commande, précédez une des lettres du caractère souligné. Vérifiez toutefois que cette lettre n'est pas déjà liée à un raccourci dans les autres étiquettes.*

Faites la même chose avec l'étiquette n° 2, en y écrivant **Build**. À présent, écrivez les commandes dans la colonne **Commande** et le nom du répertoire de Virtual Pascal dans la colonne **Dossier de travail** :

Étiquette	Commande	Dossier de travail
Compiler	vpc -Vvp.vpo "%d%\%f"	C:\Program Files \Vp21\Bin.w32
Build	vpc -B - Vvp.vpo "%d%\%f"	C:\Program Files \Vp21\Bin.w32

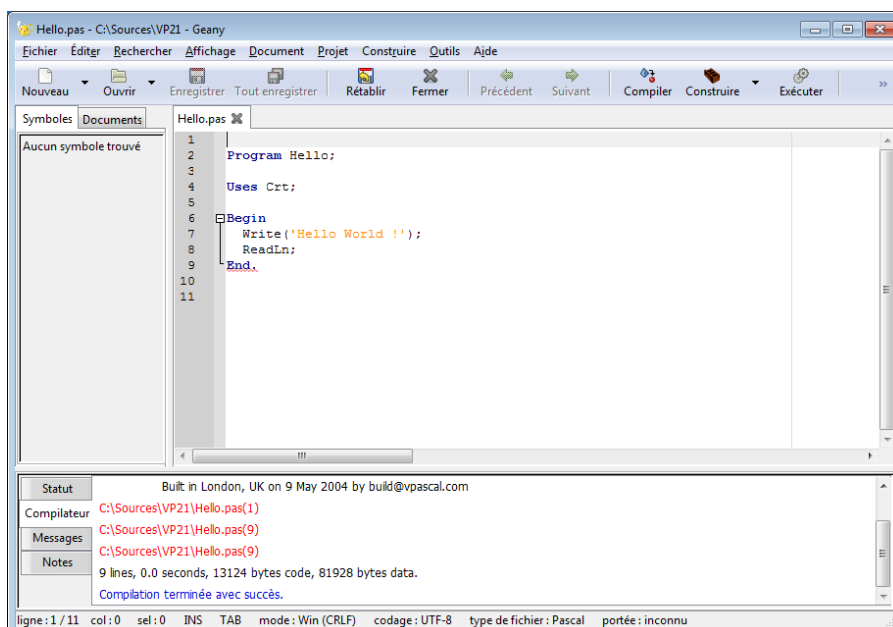


Bien entendu, il faut adapter le nom du dossier de travail à votre propre installation de Virtual Pascal. Les deux nouvelles commandes sont bien visibles dans le menu **Construire** de Geany :



Elles sont également assignées aux boutons **Compiler** et **Construire** de la barre d'outils, comme vous le voyez ci-dessous.

Cliquez sur **Compiler**. Normalement, le compilateur vous renvoie un message de succès :



Vous pouvez retourner dans le dialogue de définition des commandes et, dans le cadre du dessous, à la catégorie **Commandes d'exécution**, compléter la commande **Exécuter** :

Étiquette	Commande	Dossier de travail
Exécuter	"%e"	C:\Program Files \Vp21\Out.w32

De cette manière, vous pourrez lancer vos programmes sans devoir quitter Geany.

## Conclusion

Sur les trois plateformes qu'il supporte, **Virtual Pascal** est une excellente alternative au compilateur Free Pascal, surtout si vous portez du code de Turbo Pascal ou de Borland Pascal. Sur les versions de Windows précédant Vista, l'éditeur natif de Virtual Pascal est difficilement égalable ; par contre, sur Vista et Windows 7, l'utilisation d'un éditeur alternatif comme Geany vous offrira un meilleur confort d'utilisation.

Nous avons abordé, en introduction, plusieurs différences marquantes qui existent entre Virtual Pascal et Free Pascal. Pour bien prouver que Virtual Pascal produit, plus rapidement, du code plus compact, faites l'expérience de charger les deux programmes **Hello.pas** et **HelloGUI.pas** dans l'éditeur de Free Pascal et de les compiler. Évidemment, il s'agit d'applications ridiculement petites et les différences seront plus marquantes que sur des programmes plus lourds.

Voici un petit comparatif, avec les options par défaut des deux compilateurs :

Taille du code	Virtual Pascal	Free Pascal
Hello.pas	20 Kb	63 Kb
HelloGUI.pas	19 Kb	103 Kb

Temps de compilation	Virtual Pascal	Free Pascal
Hello.pas	0,1 s	0,3 s
HelloGUI.pas	0,1 s	1,5 s

## Remerciements

Je remercie **M.Dib** pour sa relecture technique ainsi que **Claude Leloup** pour sa relecture orthographique.