

# **Borland Delphi 8 for Microsoft .NET framework**

# *Sommaire*

Avant-propos

1. Nouveau look

2. Nouveau noyau, nouveau compilateur

2.1. Pourquoi le Framework .NET et plus Win32

2.2. .NET c'est quoi?

3. Le langage : le pascal objet

4. Les nouvelles fonctionnalités

4.1. Together

4.2. Optimized Profiler for .NET

4.3. ASP.NET

4.4. Enterprise Core Objects (ECO)

5. Portage des applications Delphi 7 et antérieures

6. Compatibilité Linux et CLX

7. Conclusion(s)

7.1. Conclusion objective

7.2. Mon avis sur Delphi 8 pour .NET

## Avant-propos

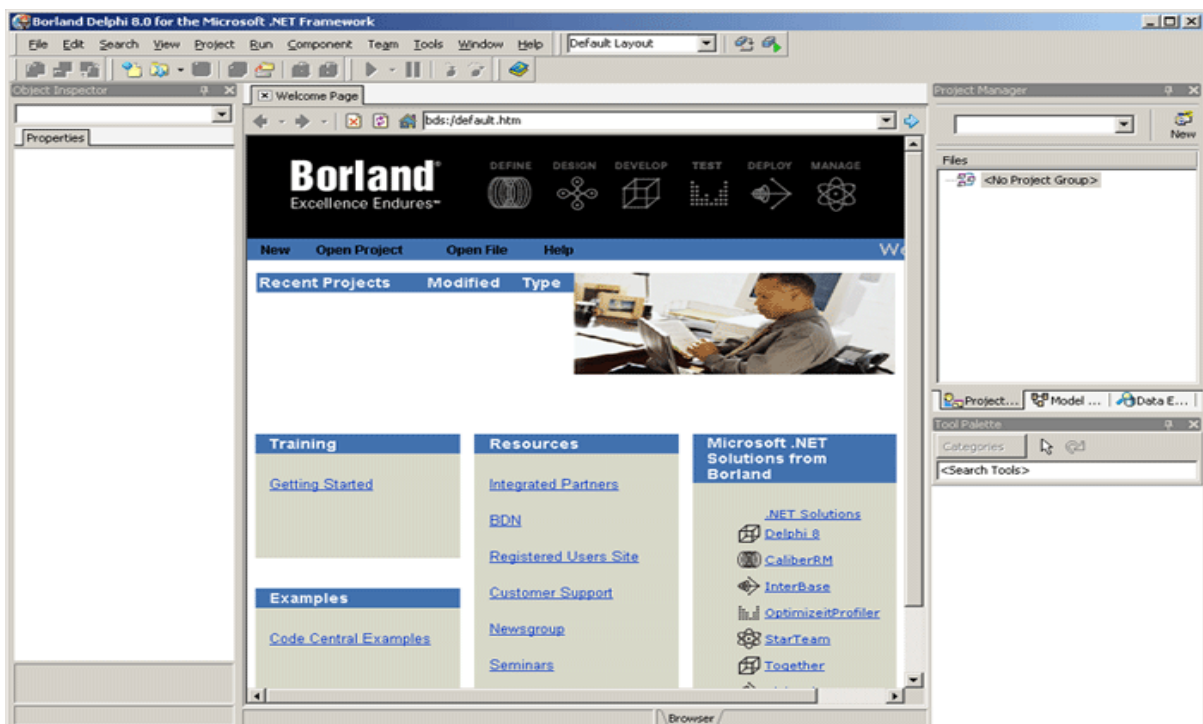
Ce document n'a pas la prétention d'être exhaustif. Il est simplement une description des points que j'ai trouvé intéressants ou non dans cette version de Borland Delphi. Il n'est rien de plus qu'un avis personnel mais que j'espère malgré tout objectif.

Cette présentation est basée sur plusieurs versions bêta, il se peut donc que le produit final présente des différences significatives.

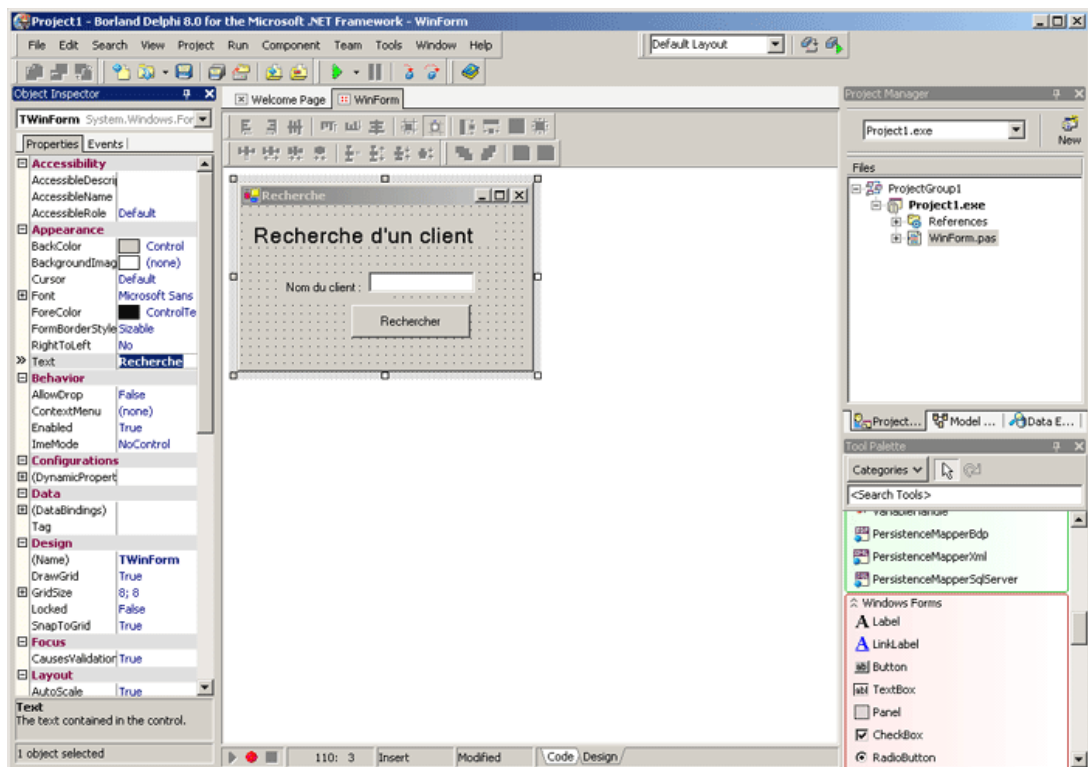
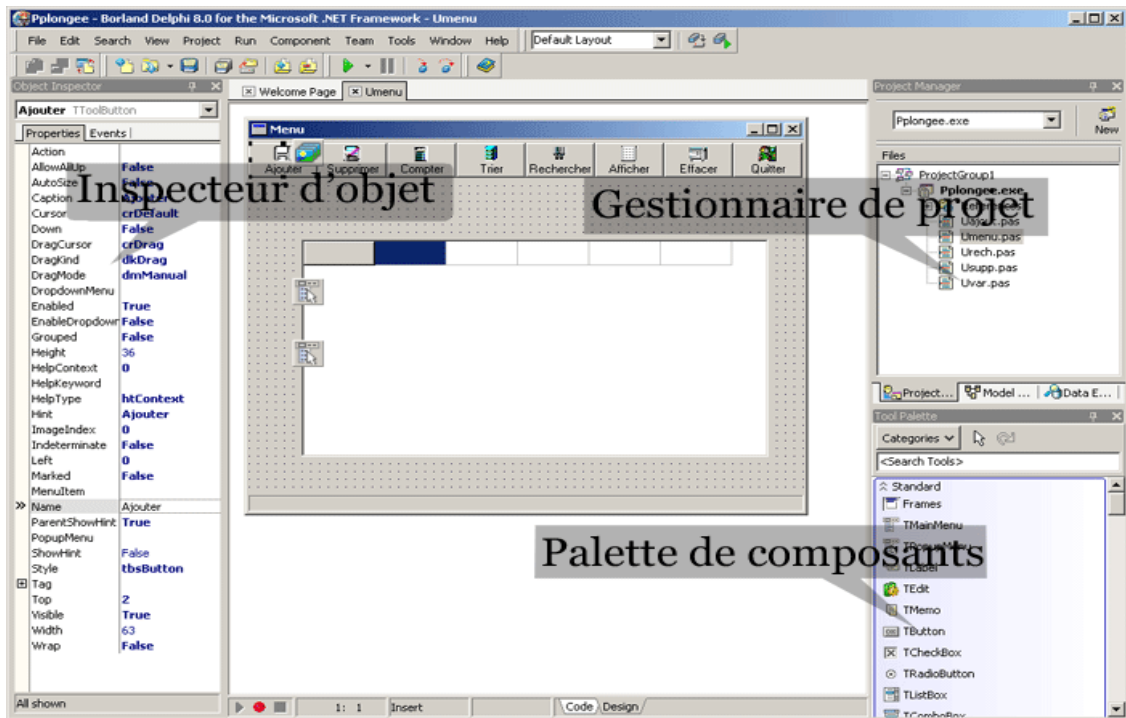
## 1. Nouveau look

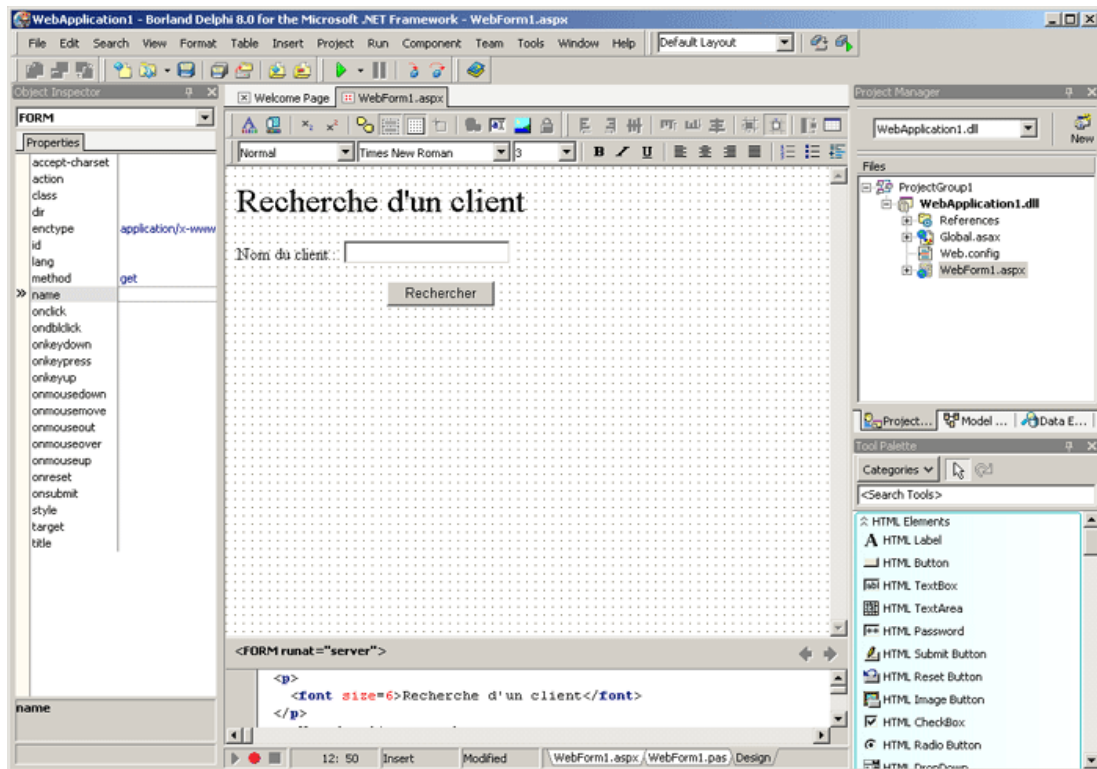
Pour cette nouvelle version de Delphi, Borland a complètement revue l'interface utilisateur qui ne reprend pas le style de la version précédente (fenêtrage multiple) mais qui se base sur l'interface de Borland C#Builder. Désormais, le développeur se retrouve face à une fenêtre unique regroupant toutefois les mêmes informations qu'avant. Ces styles très proches reflètent bien la volonté de borland d'unifier ces 2 environnements.

Cependant, pour ceux qui le désirent, on garde la possibilité de retrouver la disposition multi fenêtre des versions précédentes, toujours pratique pour les possesseurs de DualScreen.

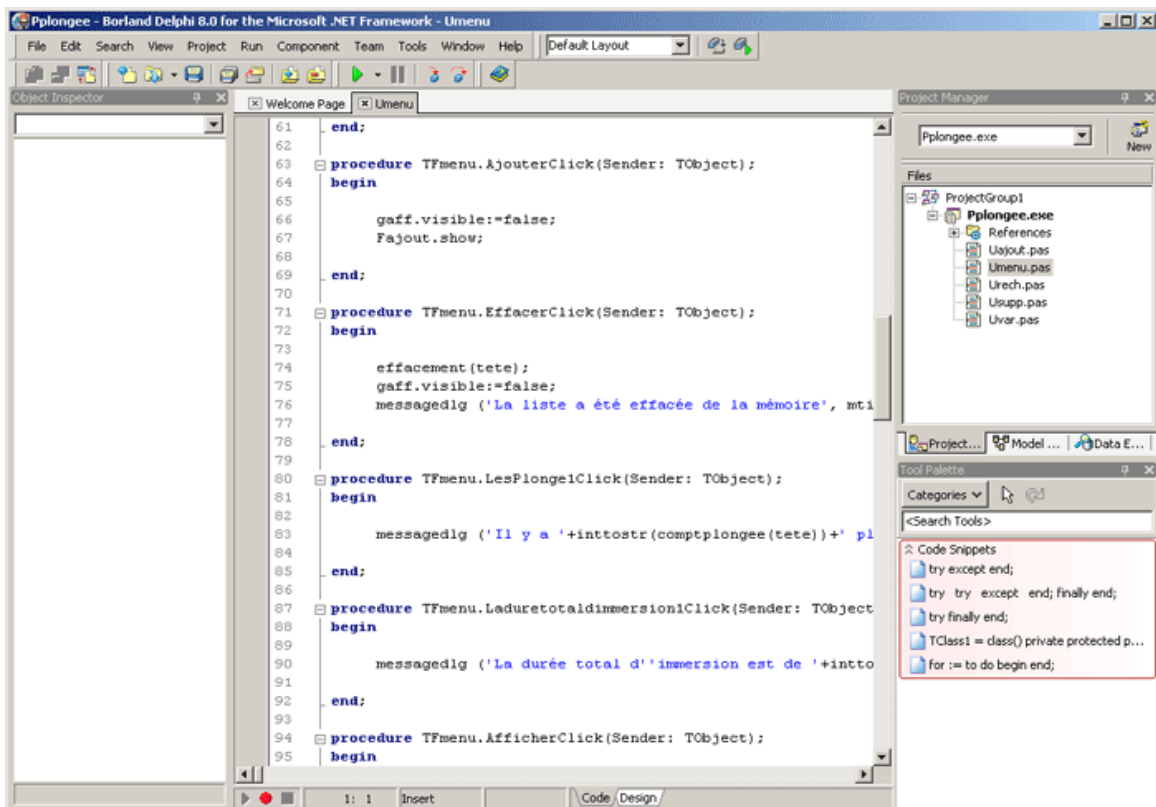


Delphi 8 possède un Form Designer permettant la conception, sous forme graphique, des fenêtres, qu'elles soient de type VCLForm, WinForm ou WebForm.





L'éditeur de code a lui aussi été revue, on y retrouve là encore la possibilité de masquer/afficher des portions de codes afin de permettre une meilleure lisibilité.



## 2. Nouveau noyau, nouveau compilateur

Si Delphi 8 ressemble tant à C#Builder, ce n'est pas un hasard. En effet, les 2 EDI phares de Borland pour la plateforme .NET partagent le même noyau appelé Galliléo.

Le changement de noyau pour Delphi 8 s'est aussi accompagné d'un changement de compilateur, et c'est sûrement là, la principale évolution que propose Delphi 8 pour .NET. En effet, l'ancien compilateur Win32 qui a en grande partie fait le succès des anciennes versions a laissé la place à un tout nouveau compilateur (DCCIL) qui permet de compiler le code Delphi en langage IL, langage intermédiaire de la plate-forme Microsoft .NET.

### 2.1. Pourquoi le Framework .NET et plus Win32

Faire ici un plaidoyer en faveur de .NET serait en dehors du cadre de cet article, mais il ne fait aucun doute que dans les prochaines années, le Framework .NET sera la plate-forme incontournable pour le développement Windows. Le projet de système d'exploitation, jusque là baptisé Longhorn par Microsoft, devrait même intégrer le framework .NET de façon native, faisant ainsi de la plate-forme Win32 une obsolescence au même titre que le DOS aujourd'hui.

### 2.2. .NET c'est quoi?

Là encore, je ne me lancerai pas dans une présentation détaillée du framework Microsoft .NET, et je vous renvoie pour cela à la [section .NET](#) et notamment à la [FAQ .NET](#).

.NET est avant tout une plate-forme de développement qui fournit un ensemble très riche de classes permettant de développer facilement tout type d'applications. .NET fournit aussi 2 types de composants permettant de faire des interfaces utilisateurs pour des applications Windows (WinForm) ou des applications web (WebForm).

.NET fournit aussi un environnement d'exécution sécurisé pour vos applications. La mémoire n'était plus gérée directement par le développeur, mais par le framework, les applications s'en trouvent donc plus stables et ne peuvent donc plus mettre en danger l'ensemble du système notamment en accédant à un espace mémoire interdit. Une gestion intelligente de la mémoire permettant aussi au développeur de ne plus se soucier de la libération de mémoire, les objets étant détruits automatiquement quand plus aucune référence n'y sont faites.

## 3. Le langage : le pascal objet

Depuis sa première version, Delphi utilise le langage Pascal Objet qui à depuis longtemps prouvé ses qualités. La version 8 ne fera pas exception. Cependant, la compilation à destination de la plate-forme .NET apporte quelques changements.

En effet, la plate-forme .NET est une plate-forme utilisable avec plusieurs langages (C#, J#, VB.NET, ...) et ces langages peuvent interagir entre eux de façon transparente. Cette transparence est possible grâce à une spécification commune à tous les langages de la plate-forme .NET : la CLS (Common Language Specification).

Delphi 8 permettant d'utiliser pleinement les fonctionnalités de la paltefome .NET, Borland a donc apporté des modifications au langage Pascal Objet, pour qu'il soit conforme à ces spécifications.

De nouvelles fonctionnalités sont donc apparues dans le langage :

- Méthodes de classe statiques
- Variables de classe (champs statiques)
- Méthodes et classes finales
- Méthodes dans un record (non virtuelles). Record = Objet automatique
- Surcharge d'opérateurs
- Classes imbriquées
- Class Helpers

Tandis que d'autres ont été supprimées telles que :

- Le type Real48 (flottant sur 6 octets)
- File of [type] (mais TextFile reste supporté)
- GetMem, FreeMem, ReallocMem
- ExitProcs
- Ancienne syntaxe de déclaration des objets (type TFoo = object)

## 4. Les nouvelles fonctionnalités

### 4.1. Together

Les utilisateurs des versions antérieures de Delphi connaissaient ModelMaker qui permettait de faire de la modélisation UML et de la génération de code pour Delphi. Les utilisateurs de Delphi 8 apprendront à connaître [Together](#).

Together est lui aussi un outil CASE UML, déjà utilisé avec d'autres produits Borland tels que JBuilder, C++BuilderX, mais aussi des produits tels que Visual Studio .NET ou Eclipse.

### 4.2. Optimizeit Profiler for .NET

Optimizeit Profiler permet au développeur d'analyser dynamiquement le comportement de son application.

En d'autres termes, Optimizeit Profiler permet en temps réel de suivre l'utilisation du processeur par l'application ou l'allocation de mémoire. Cet outil permet d'identifier et de résoudre les problèmes de performance tout au long du codage de l'application.

### 4.3. ASP.NET

Comme nous l'avons vu dans la première partie, Delphi 8 permet de développer des applications Web basées sur la technologie ASP.NET. ASP.NET est la solution du Framework .NET pour la conception d'applications Web.

Avec Delphi 8, le développeur peut concevoir une application Web ASP.NET en RAD comme pour une application Windows. Delphi 8 permet aussi de développer des WebServices ASP.NET.

### 4.4. Enterprise Core Objects (ECO)

ECO est la version .NET de Bold déjà présent dans Delphi 7. ECO permet de construire une application .NET à partir d'un modèle Objet sans se soucier de la couche persistente qu'elle soit objet ou relationnelle.

## 5. Portage des applications Delphi 7 et antérieures

Il est possible de porter des applications VCL développées sous une version antérieure de Delphi (y compris des applications 16 bits développées sous Delphi 1). Mais si le langage pascal objet a été enrichi pour le support des spécifications des langages .NET, certains éléments de ce même langage ont été abandonnés pour les mêmes raisons.

Dans le meilleur des cas, c'est-à-dire une application basée uniquement sur la VCL de Delphi 7, il suffira de re-ouvrir le projet sous Delphi 8 et de compiler pour que votre application soit portée de Win32 à .NET.

Dans les autres situations, le portage peut se révéler plus ou moins délicat, suivant les différents cas de figure.

- **Composants** : Si la VCL sous son ancienne version (Win32) a été portée sous .NET pour devenir la VCL.NET, il n'en est pas de même pour tous les composants développés par vos soins pour les bibliothèques de composants tels que la JVCL. Le portage d'une application Delphi 7 sous Delphi 8 peut donc ne pas être possible si ces



composants ne sont pas portés sous Delphi 8, ce qui peut être le cas de composants dont on ne dispose pas des codes sources, mais seulement des paquets.

- **ASM** : Là encore, le framework .NET se situe à un niveau bien supérieur à l'assembleur. Il est donc impossible d'introduire du code ASM dans un code .NET destiné à être managé par le framework.
- **Gestion directe de la mémoire** : le code manager .NET permettant une gestion intelligente de la mémoire, les cas où le développeur doit gérer lui-même la mémoire deviennent extrêmement limités. Certaines fonctions de gestion de la mémoire ne sont donc plus supportées.
- ....

Pour ceux qui souhaitent cependant encore maintenir des versions Win32 de leurs applications VCL, Borland fournit une version Delphi 7.1 [\(u\)](#) avec Delphi 8.

## 6. Compatibilité Linux et CLX

Bien qu'il existe un projet libre nommé Mono, le framework .NET n'est pas encore pleinement disponible sous Linux. Il est donc impossible de faire fonctionner une application Delphi 8 sous Linux.

Par ailleurs, la CLX introduite dans Delphi 6, qui permettait de développer des applications compilables sous Windows avec Delphi et sous Linux avec Kylix, n'est plus présente dans Delphi 8.

## 7. Conclusion(s)

### 7.1. Conclusion objective

Avec C#Builder, Borland avait été le premier éditeur indépendant à proposer un EDI pour le développement à destination du framework .NET, Borland est maintenant aussi le premier à proposer un EDI permettant le développement pour .NET basé sur un autre langage que les langages mis au point par Microsoft (à savoir C#, J# et VB.NET).

### 7.2. Mon avis sur Delphi 8 pour .NET

Durant toute cette présentation, j'ai essayé d'être le plus neutre possible et de ne m'attacher à faire qu'une présentation technique de Delphi 8 sans jugement de valeur. Cependant, je ne résiste pas à l'envie de vous donner ici un avis personnel sur Delphi 8 qui, cette fois, sera une vision plus subjective tenant compte de mes préférences et de mon expérience passée sous Delphi 7 (et antérieure).

Depuis longtemps Borland nous a habitués à des EDI de très bonne facture. Delphi 8 ne dérogera pas à la règle. Cette nouvelle version de Delphi est d'autant plus réussie que les changements introduits dans la version 8 sont importants. Les anciens utilisateurs de Delphi y retrouveront tout ce qui a fait la renommée de delphi et le langage Pascal Objet reste, malgré le passage à .NET, un modèle d'efficacité et de clarté.

Null doute que le passage réussi de Win32 vers .NET tout en conservant la possibilité de migrer l'existant sans réécrire la totalité du code fera de Delphi 8 un des EDI majeurs dans le monde du développement .NET tout comme l'ont été dans le passé les versions Win32.

Le seul regret que je peux avoir, c'est de ne plus retrouver la possibilité de faire des applications portables sous Linux grâce à la CLX. Delphi 8 étant un produit 100% .NET ne permettant plus une ouverture sur d'autres systèmes d'exploitations que ceux de Microsoft.

Enfin, lors de la dernière BonCon, Stéphane JOUANNEAU (spécialiste RAD chez Borland France) a cité une phrase de Dick LANTIM (.NET Architect Evangelist chez Microsoft), je me permet également de reprendre cette phrase pour conclure cette présentation car elle me semble bien résumer mon avis sur Delphi. **"J'ai longtemps pensé que Delphi était le meilleur environnement de développement pour Windows, et je le pense encore."**