

# **Delphi 2006 et C#**

avec Borland Developer Studio

**Olivier Dahan**

© Groupe Eyrolles, 2006,

ISBN : 2-212-11768-X

**EYROLLES**



# De Delphi à C# avec BDS 2006

---

Borland Developer Studio 2006, ou comme nous l'écrivons plus souvent dans cet ouvrage, BDS 2006, est la quatrième version du nouvel EDI de Borland. Après C# Builder, Delphi 8 et Delphi 2005, BDS 2006 apparaît comme un environnement de développement mature, riche de ces trois années d'expérience et de sa fusion des quatre grands langages que sont Delphi Win32, Delphi .NET, C# et C++. Studio de développement complet, de l'analyse au déploiement en passant par les tests, BDS 2006 offre à tous les développeurs le meilleur des deux mondes que sont Win32 et la plate-forme .NET.

## Un EDI unifié

BDS 2006 est un environnement de développement qui intègre à la fois des langages Win32 (Delphi et C++) et des langages .NET (Delphi .NET et C#). Face à d'autres EDI aujourd'hui totalement orientés .NET on peut s'interroger sur ce « grand écart » qui, de prime abord, semble risqué. Il n'en est heureusement rien et chacune des technologies présentes dans l'EDI y est parfaitement à sa place, parfaitement intégrée dans un tout cohérent. Pouvoir maintenir les applications Win32 et en créer de nouvelles sous .NET sans avoir à changer d'EDI ni à acquérir des produits différents est une manifestation positive d'un sens des réalités que nous devons mettre au crédit de Borland qui se démarque de ses compétiteurs.

BDS 2006 est le point d'orgue d'une gamme d'environnements riche de produits phares comme Delphi Win32, C++ Builder, C# Builder, Delphi 8 et 2005. Ce nouvel EDI fusionne tous ces produits en un seul et s'appuie sur l'expérience accumulée depuis 2003 pour ce qui est de son interface et de bien plus pour les langages. L'interface de BDS 2006 est en effet en rupture avec la série des Delphi 1 à 7 et des C++ Builder : tous ces EDI, pour excellents qu'ils furent, étaient avant tout de « super éditeurs » de code, orientés développeur « pur et dur », celui qui doit absolument produire du code toute la journée.

C'est une conception du métier de développeur qui appartient aujourd'hui au passé. Un développeur doit, en plus de l'écriture de code, produire ou au moins intégrer des schémas UML, suivre au plus près l'expression des besoins, mettre en place des procédures de test, gérer le déploiement des logiciels et collaborer avec d'autres développeurs.

Tout cela réclame une nouvelle approche, un saut quantitatif et qualitatif que les EDI du passé ne pouvaient pas supporter. C'est pourquoi depuis trois ans Borland nous propose un nouvel environnement qui, d'année en année, s'enrichit et se peaufine. Cet EDI moderne dans lequel l'écriture de code n'est qu'un des aspects couverts unifie et centralise les services de Together pour les schémas UML, ceux de CaliberRM pour la gestion des besoins, de StarTeam pour le travail collaboratif et de bien d'autres outils comme DUnit ou NUnit pour les tests unitaires.

Nous ne traiterons pas des aspects Win32 dans le présent ouvrage, non parce qu'ils ne présentent pas d'intérêt propre, mais simplement parce que l'auteur a déjà publié un ouvrage dédié au développement Win32, « Delphi 7 Studio » aux éditions Eyrolles, ouvrage auquel les lecteurs pourront se référer pour tout ce qui concerne la partie Delphi Win32 de BDS 2006 (ce dernier reprenant 99% de Delphi 7).

## Un EDI plus moderne

La toute première chose qui frappe est ce sentiment particulier qui balance entre une impression de déjà vu et la découverte de quelque chose de nouveau. Le nouvel EDI présente, comme le montre la figure 1-1, un grand espace de travail où les différents cadres sont ancrés.

On retrouve dans cet EDI les principales entrées de menu et les principaux cadres comme l'inspecteur d'objets ou le gestionnaire de projets. L'EDI est hautement configurable et on dispose de la possibilité d'enregistrer ses préférences sous un nom pour retrouver le même bureau au fil des utilisations (même si plusieurs personnes partagent le même poste de travail). De façon plus fine il est possible d'enregistrer un bureau pour le développement et un autre pour le débogage, BDS passant de l'un à l'autre automatiquement ce qui est vraiment très pratique.

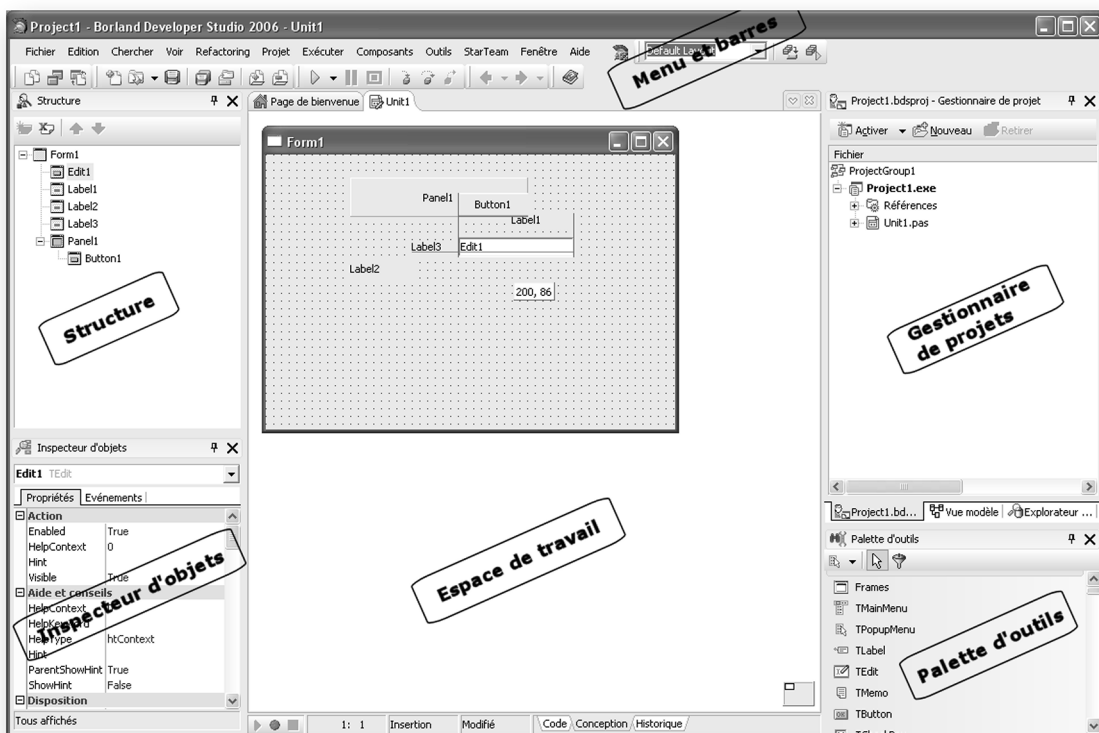


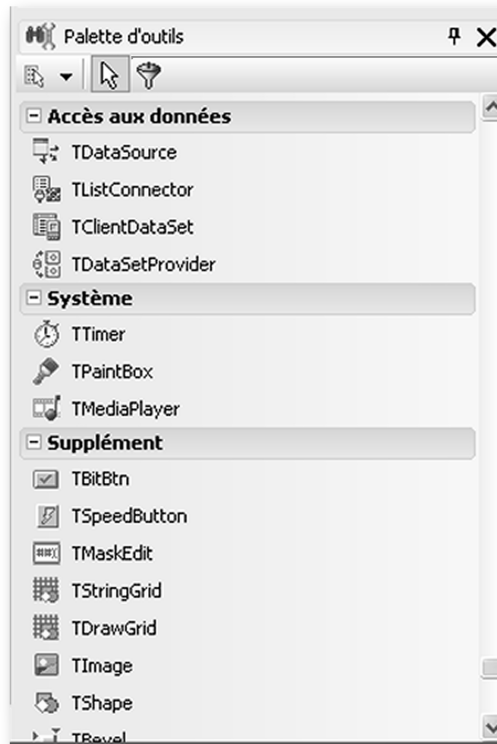
Figure 1-1

Le nouvel EDI de BDS 2006

Les différences avec les anciens EDI commencent à apparaître plus nettement dans la gestion de la palette de composants. Si vous démarrez un nouveau projet, vous constaterez que la célèbre palette introduite dans Delphi 1 n'existe plus. En réalité, elle s'appelle aujourd'hui palette d'outils et se présente sous la forme d'un menu à onglets placé en bas à droite de l'EDI, ce qui peut être modifié par l'utilisateur (figure 1-2).

Figure 1-2

*La palette d'outils*



Les objets de la palette sont organisés par onglets qui reprennent dans l'esprit la segmentation des palettes des versions précédentes de Delphi. La palette s'adapte automatiquement au type de projet en cours, les composants et les onglets affichés étant liés à ce dernier, notamment Windows Forms, VCL.NET ou Web Forms.

Le menu Composant permet d'installer soit des composants VCL, soit des composants .NET. En réalité, pour inspecter la liste des composants installés et en supprimer des palettes, il faut choisir l'option « Composants .NET installés » que l'on veuille consulter les composants VCL ou .NET.

## Les multiples personnalités

BDS 2006 est un environnement à multiple personnalités, une personnalité définissant un ensemble d'outils autour d'un langage donné. On trouve quatre personnalités qui sont :

- Delphi Win32 reprenant l'essentiel de Delphi 7 avec des améliorations tant du compilateur que de l'éditeur de form ou des composants ;

- C++ Builder signant la renaissance de cet excellent produit qui bénéficie de toutes les améliorations de l'EDI ainsi que de l'ajout d'outils propres à C++ ;
- Delphi .NET dans sa troisième version (après Delphi 8 et Delphi 2005) ;
- et enfin C# toujours servi par le compilateur Microsoft sous licence ce qui garantit un code parfaitement compatible avec la plate-forme .NET ainsi qu'avec tous les autres outils de développement du marché.

La personnalité Delphi .NET bénéficie d'une grande étendue d'action puisqu'elle permet de travailler dans les modes suivants :

- VCL.NET reprenant les composants de Delphi 7 et assurant une compatibilité ascendante proche de la perfection avec .NET ;
- WinForms pour développer des interfaces de même type que celles disponibles avec C# sous Windows ;
- WebForms pour développer, tout comme en C#, des applications Web en ASP.NET.

Le support de toutes ces personnalités fait de BDS 2006 l'outil le plus versatile du marché, le seul permettant de travailler à la fois sous Win32 et sous .NET sans aucun sacrifice au niveau fonctionnel dans aucun de ces domaines.

#### **Framework 1 ou 2 ?**

Notons que BDS 2006 tourne toujours sous le framework 1.1 de .NET. Cela s'explique par le simple fait que la version finale de .NET 2.0 est sortie à quelques jours près en même temps que BDS 2006 et que Borland ne finalise jamais un EDI sur des versions bêta de Microsoft. Désormais .NET 2.0 est stabilisé et Borland sortira fin 2006, comme tous les ans, un nouveau produit fonctionnant sous .NET 2.0 et connu sous le nom de code de Highlander. C'est d'ailleurs cette version qui supportera aussi le mode de conception visuelle pour le Compact Framework dédié aux unités mobiles. Le code écrit en framework 1 est parfaitement compatible .NET 2.

Travailler en version 1.1 du framework en 2006 n'est pas une gêne, à la fois parce que les applications écrites sont portables en framework 2.0 sans souci et aussi en raison du temps qu'il va falloir à .NET 2.0 pour être déployé sur les machines alors que .NET 1.1 est présent sur presque toutes les machines sous XP.

## **Les nouveautés de BDS 2006**

Sans reprendre la présentation du produit diffusée par Borland sur son site, nous pouvons relever quelques nouveautés particulièrement marquantes.

### ***Nouveau gestionnaire de mémoire***

BDS 2006 est doté d'un nouveau gestionnaire de mémoire plus rapide. L'EDI se charge plus vite et consomme moins de ressources.

### ***Les lignes de placement automatique***

Sous VCL et VCL.NET le concepteur visuel affiche désormais des lignes de placement lorsqu'on déplace un composant. Cela permet efficacement de cadrer des labels, des champs d'édition ou tout autre composant visuel en s'appuyant sur ses dimensions externes autant que sur l'alignement de son contenu. Vous pouvez

voir ce mode en cours de fonctionnement dans la capture d'écran présentée figure 1-1. Ce mode est supporté par Delphi Win32 et Delphi .NET ainsi que C++Builder en VCL.

### ***Composants de placement en mode flot***

VCL et VCL.NET introduisent deux nouveaux composants conteneurs qui permettent d'obtenir une mise en page de type HTML : les composants visuels contenus se déplaçant automatiquement selon la taille du conteneur exactement comme le contenu d'une page HTML s'adapte selon la taille de la fenêtre.

### ***Les nouveaux templates de code***

Toutes les personnalités de BDS 2006 bénéficient de ces nouveaux templates (modèles) de code. Reposant sur une description XML, ces nouveaux modèles se comportent plutôt comme des macros d'une grande richesse fonctionnelle.

### ***La complétion de code étendue***

L'éditeur de code sait désormais anticiper les saisies et ajoute automatiquement les fermetures de bloc en cours de frappe ce qui évite bien des oublis autant qu'une frappe clavier sans intérêt.

### ***Différentiel automatique***

L'outil différentiel de code, déjà présent depuis la page historique en cours de saisie de code, se matérialise désormais par une barre placée sur le côté gauche de la fenêtre d'édition. On voit instantanément le code qui vient d'être modifié.

### ***Les bases de données***

De nombreuses améliorations concernent aussi la gestion des bases de données et notamment le fonctionnement de l'explorateur intégré dans l'EDI. Ce dernier offre maintenant toutes les options permettant de créer des bases, des tables et des procédures stockées mais aussi les moyens de les tester. Plus intéressant encore est le mode de copie de table et de migration de données. Prenez une table dans une base Oracle, copiez-la dans une base Interbase et BDS fera le reste selon vos souhaits : copie de la structure et migration des données elles-mêmes.

Grâce à ces améliorations on peut développer sous BDS sans avoir besoin de basculer vers une console spécialisée de la ou des bases de données en cours d'utilisation.

### ***ECO III***

BDS 2006 est fourni avec la troisième version de ECO, un environnement objet persistant piloté par modèles UML. Il s'agit d'un outil fantastique disponible même dans les versions professionnelles de BDS 2006 alors qu'il était réservé aux seules versions architecte des versions précédentes de l'environnement. Nous ne parlerons pas d'ECO dans le présent ouvrage par respect pour cet ensemble car il faudrait lui consacrer de nombreux chapitres ce qui était en dehors des choix éditoriaux de l'auteur. Vous en trouverez toutefois une présentation sommaire dans notre précédent ouvrage (Delphi 8.NET chez Eyrolles) et sous la forme d'articles sur le site Web de l'auteur.

## ***Together et modélisation UML***

Parmi les grandes nouveautés de BDS 2006 l'intégration désormais achevée de Together mérite largement d'être relevée. En effet, il est maintenant possible de créer des projets UML supportant l'ensemble des diagrammes couramment utilisés. Il est même possible de choisir le niveau de la norme (1 ou 2) pour les diagrammes. Plus étonnant encore sont les possibilités de générations automatiques de code depuis les schémas UML ou la rétro-documentation permettant de créer des schémas depuis un code existant. Il faut noter que le moteur de Together a été réécrit et que le produit gagne ainsi en vitesse et fiabilité.

Comme pour ECO, cette partie réclamerait à elle seule plusieurs chapitres pour être traitée dans toute son étendue, ce qui sortait du cadre éditorial du présent ouvrage. Cela ne traduit donc en aucun cas un manque d'intérêt pour ces innovations majeures de l'EDI qui restent néanmoins réservées à des utilisateurs confirmés possédant de bonnes notions de modélisation objet. Un public exigeant que l'auteur respecte et pour qui un simple survol aurait eu un goût de trop peu.

## ***Le refactoring***

Le refactoring est une fonction intelligente qui sait réécrire le code proprement. Plutôt que de partir de règles d'écriture très strictes, rarement respectées et très contraignantes, le refactoring intègre le fait que nous ne travaillons pas toujours de façon optimale et nous aide à corriger le style ou la logique du code. Il est ainsi possible après coup d'extraire une partie de code au sein d'une méthode pour en faire une méthode autonome. BDS s'occupera d'isoler les éventuels paramètres nécessaires à son fonctionnement, il supprimera le code de son emplacement original en le remplaçant par l'appel à la nouvelle méthode qu'il aura automatiquement ajouté à la classe en cours... Et cela n'est qu'une partie des fonctions proposées ! Renommer des identificateurs de façon sûre et intelligente (pas comme un chercher/remplacer), remonter des membres dans une déclaration de classe pour en clarifier la structure, extraire automatiquement une super classe ou déclarer automatiquement des variables ou des champs en cours d'écriture de code, tout cela apporte un grand confort et un réel gain de productivité.

En plus de ces avantages, utiliser le refactoring permet de produire un code plus propre, plus facile à maintenir et de meilleur qualité.

## **Compatibilité ascendante**

BDS 2006 est bien entendu compatible pour sa partie .NET avec Delphi 2005 et Delphi 8, qu'il s'agisse du langage Delphi .NET ou de C#. Mais pour les utilisateurs venant d'une version Win32 de Delphi, il faut noter que BDS 2006 vous aidera automatiquement à traduire vos applications en Delphi .NET et ce grâce à la VCL.NET totalement compatible avec Delphi 7.

Si les composants tiers peuvent poser des problèmes dès lors qu'une version VCL.NET n'est pas proposée par leurs éditeurs, sachez que convertir un composant n'est guère plus complexe que la traduction d'une application Win32. Seul le code faisant des appels directs à l'API Windows ou contenant du code « unsafe » (pointeurs, enregistrements avec variable, etc.) ou de l'ASM en ligne devra être réécrit. Toute la VCL, c'est-à-dire autant les composants visibles que les routines de *StrUtils* ou d'autres unités de code de service ont été traduits par Borland.

Porter une application VCL vers VCL.NET offre la possibilité de disposer ensuite de tout le framework pour la faire évoluer. Migrer une application Win32, faire en sorte qu'elle puisse être maintenue tant sur le plan correctif qu'évolutif n'est pas un luxe, c'est surtout la conserver à jour et en vie ! Si toutes les applications n'ont

pas besoin d'être ainsi portées (petits utilitaires, applications en fin de vie) celles pour lesquelles vous espérez un avenir doivent l'être. BDS 2006 vous y aidera efficacement tout autant qu'il vous permettra par le biais de sa personnalité Delphi Win32 de maintenir les anciennes applications sans avoir à jongler entre les EDI.

## Les nouveautés de Delphi .NET

Pour ceux d'entre vous qui viendront à BDS 2006 depuis une version Win32 de Delphi les nouveautés du langage sont nombreuses. Elles le sont tellement que les trois prochains chapitres leurs sont totalement consacrés !

Pour les utilisateurs déjà familiarisés avec Delphi 8 ou Delphi 2005 les nuances sont plus subtiles mais elles existent, vous les retrouverez aussi dans les prochains chapitres (avec l'indication de la version dans laquelle elles ont été introduites).

## Conclusion

Un EDI est avant tout un compagnon de travail. En présenter exhaustivement toutes les nouveautés n'apporterait pas grand-chose. Un EDI se pratique, se vit. Nous avons ici attiré votre attention sur des changements ou des innovations d'importance, mais il en existe bien d'autres. À vous de les découvrir au fur et à mesure que vous prendrez possession de votre nouvel outil de travail, chemin sur lequel l'ensemble des chapitres qui vont suivre vont s'attacher à vous accompagner.