



Prise en main



Borland®
Delphi™ 7
pour Windows™

Reportez-vous au fichier DEPLOY situé dans le répertoire racine de votre produit Delphi 7 pour obtenir la liste complète des fichiers que vous pouvez distribuer conformément aux termes du contrat de licence de Delphi.

Les applications mentionnées dans ce manuel sont brevetées ou en attente de brevet. Ce document ne donne aucun droit sur ces brevets. Reportez-vous au CD du produit ou à la boîte de dialogue A propos.

COPYRIGHT © 1983–2002 Borland Software Corporation. Tous droits réservés. Tous les produits Borland sont des marques commerciales ou des marques déposées de Borland Software Corporation aux Etats-Unis ou dans les autres pays. Toutes les autres marques sont la propriété de leurs fabricants respectifs.

Imprimé en Irlande

HDE1370WW21000 7E5R0802

0203040506-9 8 7 6 5 4 3 2 1

D3

Table des matières

Chapitre 1

Introduction 1-1

Qu'est-ce que Delphi ?	1-1
Enregistrement de Delphi	1-2
Où trouver des informations ?	1-4
Aide en ligne	1-5
Aide F1	1-5
Services de support développeur	1-6
Conventions typographiques	1-7

Chapitre 2

Parcours rapide de l'environnement 2-1

Démarrage de Delphi	2-1
L'EDI	2-1
Menus et barres d'outils	2-3
Le concepteur de fiche, la palette de composants et l'inspecteur d'objets	2-4
L'arborescence d'objets	2-5
Le référentiel d'objets	2-6
L'éditeur de code	2-7
Audit de code	2-7
Achèvement de classe	2-8
Navigateur de code	2-9
La page Diagramme	2-9
Visualisation du code des fiches	2-10
L'explorateur de code	2-11
Le gestionnaire de projet	2-12
L'explorateur de projet	2-13
Listes A faire	2-13

Chapitre 3

Programmation avec Delphi 3-1

Création d'un projet	3-1
Ajout de modules de données	3-2
Construction de l'interface utilisateur	3-2
Positionnement des composants sur une fiche	3-2
Définition des propriétés des composants	3-3
Ecriture du code	3-5
Ecriture de gestionnaires d'événements	3-5
Utilisation de la bibliothèque de composants	3-5

Compilation et débogage des projets	3-6
Déploiement des applications	3-8
Internationalisation d'applications	3-8
Types de projets	3-9
Applications CLX	3-9
Applications serveur Web	3-9
Applications de bases de données	3-10
Administrateur BDE	3-11
Explorateur SQL (explorateur de base de données)	3-11
Module Base de données	3-12
Dictionnaire de données	3-12
Composants personnalisés	3-12
DLL	3-12
COM et ActiveX	3-13
Bibliothèques de types	3-13

Chapitre 4

Personnalisation du bureau 4-1

Organisation de votre espace de travail	4-1
Agencement des menus et des barres d'outils	4-1
Ancrage des fenêtres d'outils	4-2
Enregistrement des dispositions du bureau	4-4
Personnalisation de la palette de composants	4-5
Organisation de la palette de composants	4-5
Création de modèles de composants	4-6
Installation de paquets de composants	4-6
Utilisation des cadres	4-7
Ajout de contrôles ActiveX	4-8
Définition des options des projets	4-8
Définition des options de projet par défaut	4-8
Spécification des modèles de projet et de fiches par défaut	4-8
Ajout de modèles au référentiel d'objets	4-9
Définition des préférences pour les outils	4-10
Personnalisation du concepteur de fiche	4-10
Personnalisation de l'éditeur de code	4-11
Personnalisation de l'explorateur de code	4-11

Index I-1

1

Introduction

Ce manuel, *Prise en main*, présente l'environnement de développement de Delphi pour vous permettre d'utiliser immédiatement le produit. Il indique aussi où rechercher des renseignements sur l'utilisation des outils et des fonctionnalités disponibles dans Delphi.

Le [Chapitre 2, "Parcours rapide de l'environnement"](#), décrit les principaux outils disponibles sur le bureau de Delphi, c'est-à-dire l'environnement de développement intégré (EDI). Le [Chapitre 3, "Programmation avec Delphi"](#), explique l'utilisation de certains de ces outils pour créer une application. Le [Chapitre 4, "Personnalisation du bureau"](#), décrit comment personnaliser l'EDI de Delphi pour qu'il réponde à vos besoins de développement.

Pour avoir des instructions étape par étape sur la façon d'utiliser Delphi pour écrire des programmes, comme un éditeur de texte ou une application de base de données, voir l'aide en ligne ("Tutoriels" dans le sommaire) ou les fichiers PDF des tutoriels dans le répertoire d'installation de Delphi.

Qu'est-ce que Delphi ?

Delphi est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (RAD). En utilisant Delphi, vous pouvez créer de puissantes applications pour Microsoft Windows XP, Microsoft Windows 2000 et Microsoft Windows 98, avec un minimum de programmation. Delphi devient une solution multiplate-forme lorsqu'il est utilisé avec Kylix, l'outil RAD de Borland pour Linux. Delphi fournit tous les outils nécessaires pour développer, tester et déployer des applications, notamment une importante bibliothèque de composants réutilisables, une suite d'outils de conception, des modèles d'applications et de fiches et des experts de programmation.

Enregistrement de Delphi

Delphi peut être enregistré de plusieurs façons. La première fois que vous lancez Delphi après son installation, vous êtes invité à entrer un numéro de série et une clé d'autorisation. Après cela, la boîte de dialogue d'enregistrement vous propose quatre modes d'enregistrement correspondant aux options suivantes :

- S'enregistrer en utilisant la connexion Internet.

Utilisez cette option pour effectuer l'enregistrement en ligne via votre connexion internet en cours.

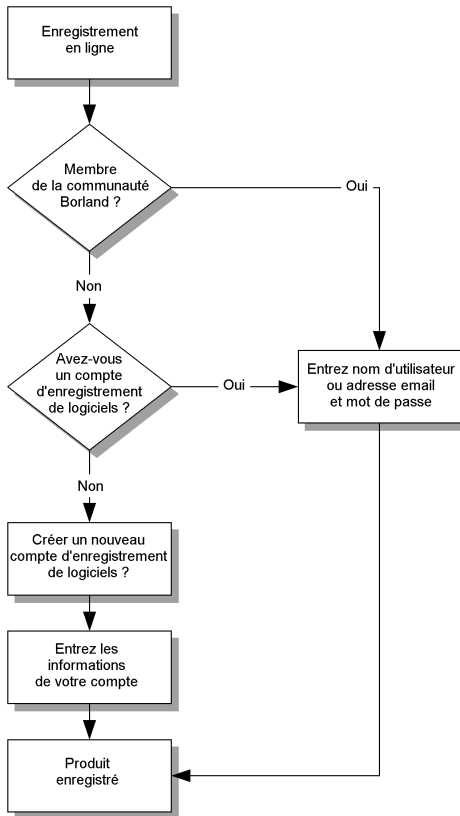
- S'enregistrer par téléphone ou par le Web.

Utilisez cette option pour effectuer l'enregistrement par téléphone ou via votre navigateur web. Si vous avez reçu une clé d'activation par e-mail, utilisez cette option pour sélectionner le fichier.

- Importer les informations d'activation depuis un fichier ou un e-mail.
- S'enregistrer ultérieurement.

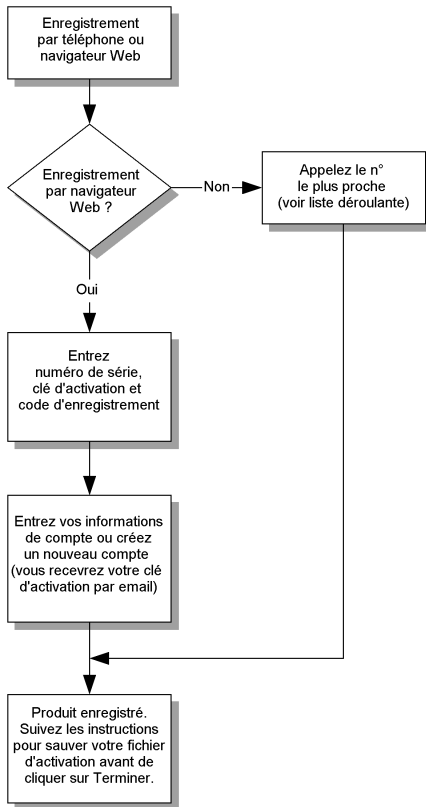
L'enregistrement en ligne est le moyen le plus simple d'enregistrer Delphi, mais il nécessite qu'une connexion internet soit active. Si vous êtes déjà membre de la Communauté Borland ou si vous avez un compte d'enregistrement de logiciels, entrez simplement les informations concernant ce compte. Cela enregistrera automatiquement Delphi.

Si vous n'avez aucun compte, le processus d'enregistrement vous permet de créer un compte de membre de la Communauté Borland ou un compte d'enregistrement de logiciels.



Où trouver des informations ?

La deuxième option (enregistrement par téléphone ou par le Web) est utile si la machine sur laquelle vous effectuez l'installation n'est pas connectée à Internet ou si un pare-feu bloque les enregistrements en ligne.



Si vous avez déjà reçu les informations d'activation du logiciel, vous pouvez sélectionner l'option *Importer les informations d'activation depuis un fichier ou un e-mail* puis le fichier activation.slip.

Remarque Sauf si vous avez une raison particulière de ne pas le faire, préférez l'option d'enregistrement en ligne.

Où trouver des informations ?

Vous pouvez trouver des informations sur Delphi de l'une des façons suivantes :

- dans l'aide en ligne,
- dans la documentation imprimée,
- sur le site Web de Borland et de ses services de support aux développeurs.

Pour des informations sur les nouveautés de cette version, reportez-vous à la rubrique Nouveautés de l'aide en ligne et au site Web www.borland.fr.

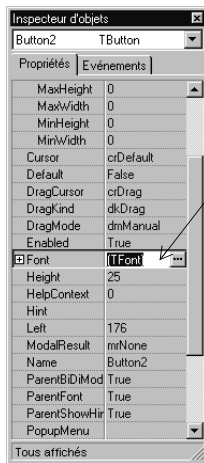
Aide en ligne

Le système d'aide en ligne fournit des informations détaillées sur les fonctionnalités de l'interface utilisateur, l'implémentation du langage, les tâches de programmation et les composants. Il contient toutes les informations se trouvant dans le *Guide du développeur* et dans le *Guide du langage Delphi*, ainsi que de nombreux fichiers d'aide relatifs aux autres éléments fournis avec Delphi.

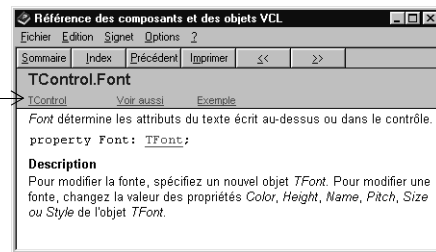
Pour afficher la table des matières, choisissez Aide | Aide Delphi et Aide | Outils Delphi, puis choisissez l'onglet Sommaire. Pour rechercher des informations sur les composants ou sur d'autres sujets, choisissez les onglets Index ou Rechercher et saisissez votre requête.

Aide F1

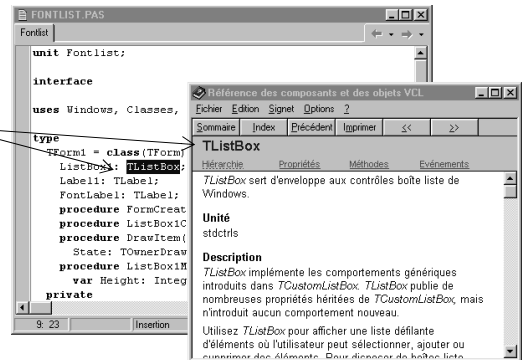
Pour obtenir de l'aide sur tout élément de l'environnement de développement (y compris les éléments de menu, les boîtes de dialogue, les barres d'outils et les composants) sélectionnez l'élément et appuyez sur **F1**.

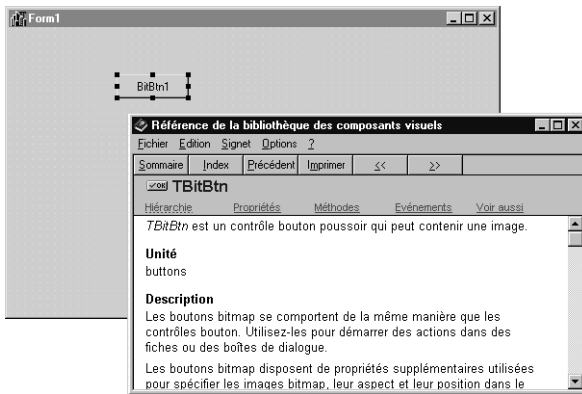


Appuyez sur **F1** au niveau d'un nom de propriété ou d'événement dans l'inspecteur d'objets afin d'afficher l'aide VCL.



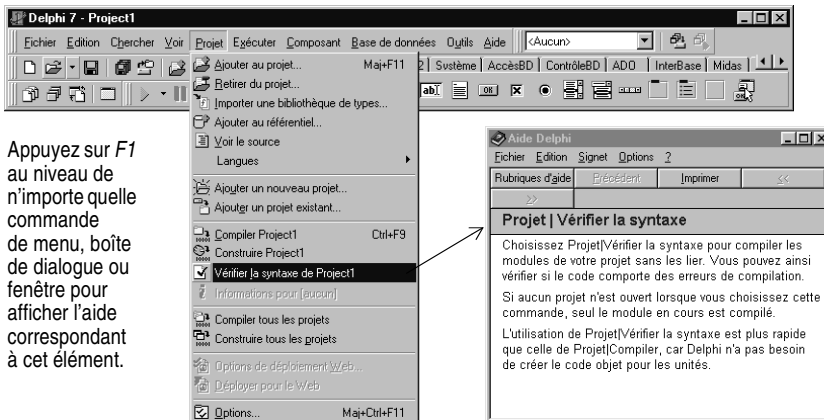
Dans l'éditeur de code, appuyez sur **F1** au niveau d'un élément du langage, de la VCL ou de la CLX.





Appuyez sur **F1** au niveau d'un composant dans une fiche.

L'appui sur le bouton Aide dans toute boîte de dialogue affiche également une aide contextuelle.



Les messages d'erreur issus du compilateur et du lieu apparaissent dans une fenêtre spéciale située sous l'éditeur de code. Pour avoir de l'aide sur les erreurs de compilation, sélectionnez un message dans la liste et appuyez sur **F1**.

Services de support développeur

Borland offre de nombreuses possibilités de support si vous avez besoin d'informations supplémentaires. Pour des informations sur les services de support développeur de Borland, visitez le site Web à l'adresse <http://www.borland.fr/support>.

A partir de ce site, vous pouvez accéder à divers groupes de discussion où les développeurs Delphi échangent informations, astuces et techniques. Ce site propose également une liste de livres traitant de Delphi, d'autres documents techniques Delphi et des foires aux questions (FAQ).

Conventions typographiques

Ce manuel utilise les types de caractères décrits dans le tableau ci-dessous.

Tableau 1.1 Conventions typographiques

Types de caractères	Signification
Espacement fixe	Les caractères d'espacement fixe représentent du texte qui apparaît sur l'écran ou du code. Ils indiquent aussi les valeurs à saisir au clavier.
Gras	Les mots en gras dans le texte ou les programmes représentent des mots réservés ou des options du compilateur.
<i>Italique</i>	Les mots en italique dans le texte représentent des identificateurs Delphi tels que des noms de variables ou de types. L'italique sert aussi à faire ressortir certains mots comme les nouveaux termes.
<i>Touches</i>	Cette police sert à indiquer une touche du clavier. Par exemple, "Appuyez sur <i>Echap</i> pour fermer un menu."

Parcours rapide de l'environnement

Ce chapitre explique comment démarrer Delphi et donne un aperçu rapide des principaux composants et des outils de l'environnement de développement intégré (EDI).

Démarrage de Delphi

Vous pouvez démarrer Delphi de plusieurs façons :

- Double-cliquer sur l'icône Delphi (si vous avez créé un raccourci).
- Choisir Programmes | Borland Delphi 7 | Delphi 7 dans le menu Démarrer de Windows.
- Choisir Exécuter dans le menu Démarrer de Windows, puis entrer Delphi32.
- Double-cliquer sur Delphi32.exe dans le répertoire Delphi\Bin.

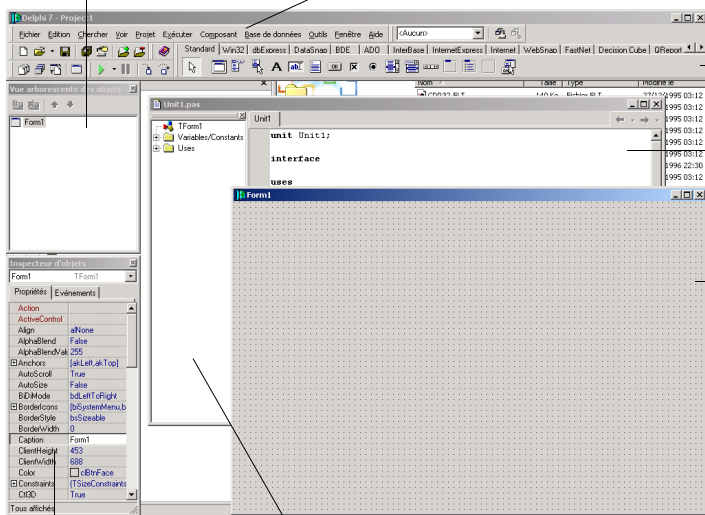
L'EDI

Lorsque vous démarrez Delphi pour la première fois, vous voyez apparaître quelques-uns des principaux outils proposés dans l'EDI. Dans Delphi, l'EDI comprend des barres d'outils, des menus, la palette de composants, l'inspecteur d'objets, l'arborescence d'objets, l'éditeur de code, l'explorateur de code, le gestionnaire de projet et bien d'autres outils.

Selon l'édition de Delphi, les caractéristiques et les composants disponibles varient.

La vue arborescente de l'objet affiche une représentation hiérarchique des relations parent-enfant des composants.

Les menus et les barres d'outils permettent d'accéder à tout un ensemble de fonctions et d'outils qui vous aident à écrire vos applications.



La palette de composants contient des composants prêts à être utilisés dans vos projets.

L'éditeur de code permet d'afficher et de modifier le code.

Le concepteur de fiche contient une fiche vierge pour démarrer la conception de l'interface utilisateur de votre application. Une application peut comporter plusieurs fiches.

L'inspecteur d'objets permet de changer des propriétés et de sélectionner des gestionnaires d'événements.

L'explorateur de code montre les classes, variables et routines de votre unité et vous permet de naviguer rapidement.

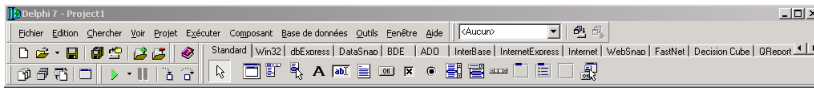
Le modèle de développement Delphi repose sur des outils *bidirectionnels*. Cela signifie que vous pouvez passer alternativement des outils de conception visuelle aux modifications du code sous forme de texte. Par exemple, après avoir utilisé le concepteur de fiche pour agencer les boutons et les autres éléments d'une interface graphique, vous pouvez voir immédiatement le fichier texte contenant la description textuelle de votre fiche. De même, vous pouvez modifier manuellement tout code généré par Delphi sans perdre l'accès à l'environnement de programmation visuelle.

Dans l'EDI, tous les outils de programmation sont facilement accessibles. Vous pouvez concevoir des interfaces graphiques, parcourir les bibliothèques de classes, écrire du code, compiler, tester, déboguer et gérer des projets sans quitter l'EDI.

Pour apprendre à organiser et à configurer l'EDI, reportez-vous au [Chapitre 4, "Personnalisation du bureau"](#).

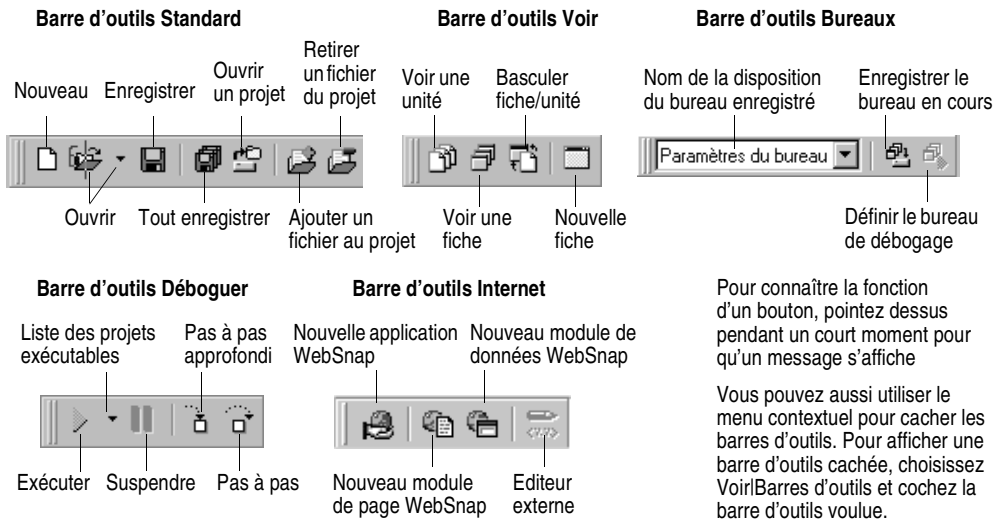
Menus et barres d'outils

La fenêtre principale, qui occupe la partie supérieure de l'écran, contient le menu principal, les barres d'outils et la palette de composants.



Fenêtre principale dans son agencement par défaut.

Les barres d'outils de Delphi permettent un accès rapide aux opérations et aux commandes les plus utilisées. La plupart des opérations accessibles dans les barres d'outils se retrouvent dans les menus déroulants.



Pour connaître la fonction d'un bouton, pointez dessus pendant un court moment pour qu'un message s'affiche

Vous pouvez aussi utiliser le menu contextuel pour cacher les barres d'outils. Pour afficher une barre d'outils cachée, choisissez Voir\Barres d'outils et cochez la barre d'outils voulue.

De nombreuses opérations possèdent des équivalents clavier ainsi que des boutons dans la barre d'outils. Lorsqu'un raccourci clavier est disponible, il est toujours inscrit à côté de la commande dans le menu déroulant.

Vous pouvez cliquer avec le bouton droit de la souris sur la plupart des outils et des icônes pour afficher un menu de commandes appropriées à l'objet que vous manipulez. Il s'agit des *menus contextuels*.

Les barres d'outils peuvent également être personnalisées. Vous pouvez leur ajouter des commandes ou les déplacer. Pour plus d'informations, voir ["Agencement des menus et des barres d'outils"](#) à la page 4-1 et ["Enregistrement des dispositions du bureau"](#) à la page 4-4.

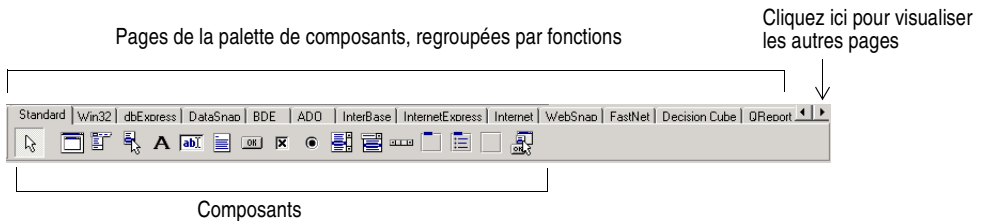
Pour plus d'informations...

Si vous voulez des informations sur une option de menu, pointez-la et appuyez sur **F1**.

Le concepteur de fiche, la palette de composants et l'inspecteur d'objets

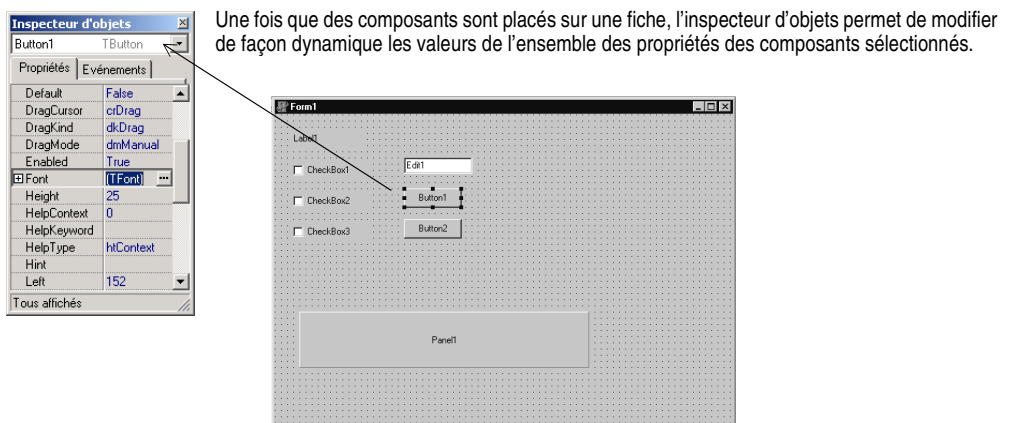
La palette de composants, le concepteur de fiche, l'inspecteur d'objets et l'arborescence d'objets vous permettent de concevoir l'interface utilisateur d'une application.

La *palette de composants* est constituée de pages à onglets où des groupes d'icônes représentent les composants visuels ou non visuels. Les pages regroupent les composants selon leurs diverses fonctionnalités. Ainsi les pages Supplément et Win32 contiennent des contrôles fenêtrés comme les boîtes de saisie ou les cases à cocher ; la page Dialogues contient les boîtes de dialogue standard utilisées dans les opérations de fichier comme l'ouverture ou l'enregistrement de fichiers.



Chaque composant possède des attributs spécifiques (propriétés, événements et méthodes) vous permettant de contrôler votre application.

Utilisez le *concepteur de fiche* pour arranger les composants à votre convenance dans l'interface utilisateur. Pour chaque composant que vous placez sur une fiche, utilisez l'*inspecteur d'objets* pour définir des propriétés de conception, créer des gestionnaires d'événements, filtrer la visibilité des propriétés et des événements, ce qui établira le lien entre l'apparence visuelle de votre application et le code qui la fait s'exécuter. Voir "[Positionnement des composants sur une fiche](#)" à la page 3-2.



Pour plus d'informations...

Voir "palette de composants" dans l'index de l'aide en ligne.

L'arborescence d'objets

L'arborescence d'objets affiche les relations parent-enfant et entre enfants des composants dans une vue hiérarchique arborescente. Cette arborescence est synchronisée avec l'inspecteur d'objets et le concepteur de fiche : si vous déplacez la sélection dans l'arborescence d'objets, elle est déplacée en conséquence dans l'inspecteur d'objets et dans la fiche.

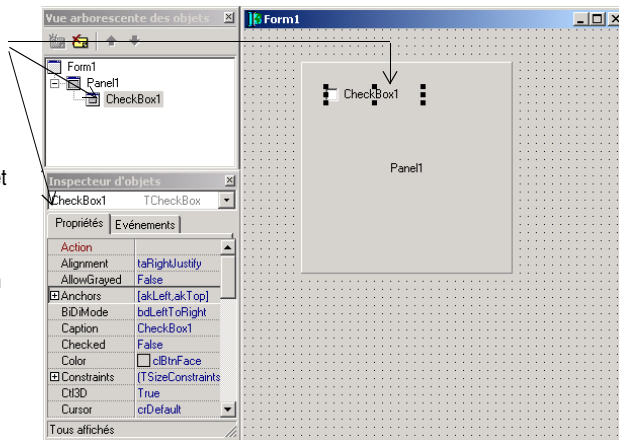
L'arborescence d'objets vous permet de changer la relation établie entre des composants. Si par exemple, vous ajoutez des composants volet et case à cocher à votre fiche, ces deux composants sont frères. Mais, si vous utilisez l'arborescence d'objets pour déplacer la case à cocher sur l'icône du volet, la case à cocher devient un enfant du volet.

Si les propriétés d'un objet n'ont pas été renseignées, l'arborescence d'objets affiche un point d'interrogation rouge à côté. Vous pouvez également double-cliquer sur un objet dans l'arborescence pour ouvrir l'éditeur de code afin d'écrire un gestionnaire d'événement.

Si l'arborescence d'objets n'apparaît pas, sélectionnez Voir | Arborescence d'objets.

L'arborescence d'objets, l'inspecteur d'objets et le concepteur de fiche fonctionnent ensemble. Quand vous cliquez sur un objet de votre fiche, la focalisation change à la fois dans l'arborescence et dans l'inspecteur d'objets et réciproquement.

Appuyez sur *Alt-Maj-F11* pour passer la focalisation à l'arborescence d'objets.



L'arborescence d'objets est particulièrement utile pour afficher les relations entre objets de base de données.

Pour plus d'informations...

Voir "arborescence d'objets" dans l'index de l'aide.

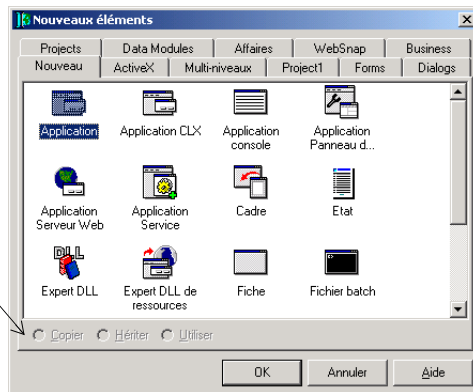
Le référentiel d'objets

Le référentiel d'objets contient des fiches, boîtes de dialogue, modules de données, experts, exemples d'applications DLL ainsi que d'autres éléments pouvant simplifier le développement. Choisissez Fichier | Nouveau | Autre pour afficher la boîte de dialogue Nouveaux éléments par laquelle débute un projet. La boîte de dialogue Nouveaux éléments est également appelée le référentiel d'objets. Recherchez dans le référentiel un objet qui ressemble à celui que vous voulez créer.

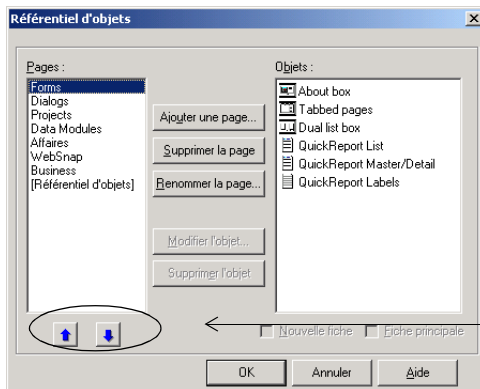
Les pages à onglets du référentiel d'objets contiennent des objets tels des fiches, des cadres, des unités, ainsi que des experts pour créer des éléments spécialisés.

Quand vous créez un élément à partir d'un élément du référentiel, vous pouvez copier, hériter ou utiliser l'élément.

Copier (par défaut) crée une copie de l'élément dans votre projet. *Hériter* signifie que les modifications apportées à l'objet du référentiel sont héritées par l'objet de votre projet. *Utiliser* signifie que les modifications apportées à l'objet de votre projet sont héritées par l'objet du référentiel.



Pour modifier ou supprimer des objets dans le référentiel d'objets, choisissez Outils | Référentiel ou cliquez avec le bouton droit dans la boîte de dialogue Nouveaux éléments et choisissez Propriétés.



Vous pouvez ajouter, supprimer ou renommer un onglet dans le référentiel d'objets.

Cliquez sur les flèches pour modifier l'ordre dans lequel apparaissent les onglets de la boîte de dialogue Nouveaux éléments.

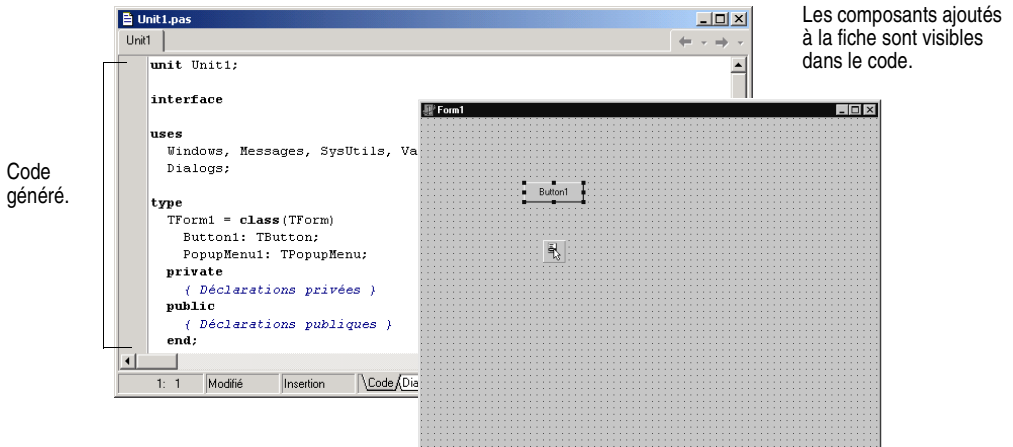
Pour ajouter des modèles de projet ou de fiche au référentiel d'objets, voir ["Ajout de modèles au référentiel d'objets"](#) à la page 4-9.

Pour plus d'informations...

Voir "référentiel d'objets" dans l'index de l'aide en ligne. Les objets disponibles dépendent de l'édition de Delphi dont vous disposez.

L'éditeur de code

A mesure que vous construisez l'interface utilisateur de votre application, Delphi génère le code sous-jacent. Lorsque vous sélectionnez et modifiez les propriétés des fiches et des composants, ces modifications sont automatiquement répercutées dans les fichiers source. Vous pouvez ajouter directement du code à vos fichiers source en utilisant l'éditeur de code intégré, un éditeur ASCII complet. Delphi propose diverses aides à l'écriture de code, dont les outils d'audit de code (Code Insight), l'achèvement de classe et la navigation dans le code.



Audit de code

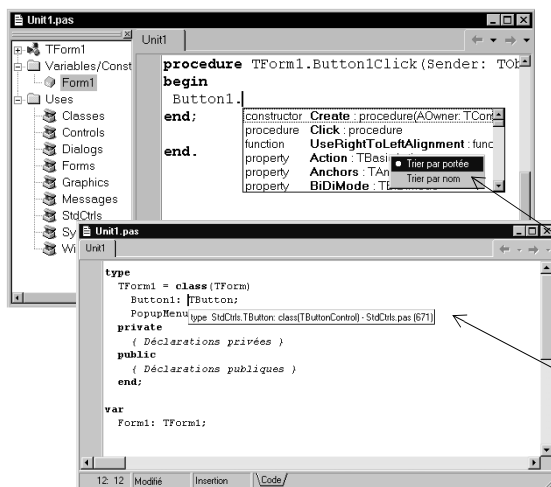
Les outils d'audit de code (Code Insight) affichent des fenêtres surgissantes contextuelles.

Tableau 2.1 Outils de l'audit de code

Outil	Fonctionnement de l'outil
Achèvement du code	Entrez le nom d'une classe suivi d'un point (.) pour afficher la liste des propriétés, méthodes et événements appropriés à cette classe ; sélectionnez un élément et appuyez sur <i>Entrée</i> . Dans la section interface du code, vous pouvez sélectionner plusieurs éléments. Tapez le début d'une instruction d'affectation puis appuyez sur <i>Ctrl+espace</i> pour afficher une liste d'arguments valides pour la variable. Tapez un nom de procédure, de fonction ou de méthode pour faire apparaître une liste d'arguments.
Paramètres du code	Tapez un nom de méthode et une parenthèse ouvrante pour afficher la syntaxe des arguments de la méthode.
Evaluation d'expression dans conseil	Pendant que votre programme est suspendu au cours du débogage, pointez sur une variable pour afficher sa valeur courante.

Tableau 2.1 Outils de l'audit de code

Outil	Fonctionnement de l'outil
Audit de symbole dans conseil	Pendant l'édition du code, pointez sur un identificateur pour afficher sa déclaration.
Modèles de code	Appuyez sur <i>Ctrl+J</i> pour lister les instructions de programmation courantes que vous pouvez insérer dans votre code. Vous pouvez créer vos propres modèles qui viendront s'ajouter à ceux disponibles.



Grâce à l'achèvement du code, quand vous entrez le point de `Button1`. Delphi affiche la liste des propriétés, méthodes et événements de cette classe. Au fur et à mesure que vous tapez, la liste filtre automatiquement la sélection relative à cette classe. Sélectionnez un élément de la liste et appuyez sur *Entrée* pour ajouter cet élément à votre code.

Les procédures et les propriétés sont colorées en brun et les fonctions en bleu.

Vous pouvez trier cette liste dans l'ordre alphabétique en cliquant avec le bouton droit de la souris puis en cliquant sur *Trier par nom*.

L'audit de symbole dans conseil affiche les informations de déclaration pour tout identificateur lorsque vous faites passer le curseur de la souris au-dessus de lui.

Pour activer ou désactiver ces outils, choisissez Outils | Options de l'éditeur et sélectionnez l'onglet Audit de code. Cochez ou non les outils dans la zone Fonctionnalités automatiques.

Achèvement de classe

L'achèvement de classe génère le code squelette des classes. Placez le curseur n'importe où dans la déclaration d'une classe dans la section **interface** d'une unité et appuyez sur *Ctrl+Maj+C* ou cliquez avec le bouton droit et choisissez Compléter la classe sous le curseur. Delphi ajoute automatiquement les spécificateurs privés **read** et **write** aux déclarations de toutes les propriétés qui l'exigent, puis crée le code squelette de toutes les méthodes de la classe. Vous pouvez aussi utiliser l'achèvement de classe pour remplir les déclarations des méthodes que vous avez déjà implémentées.

Pour activer l'achèvement de classe, choisissez Outils | Options d'environnement, cliquez sur l'onglet Explorateur et vérifiez que l'option Finir les propriétés incomplètes est cochée.

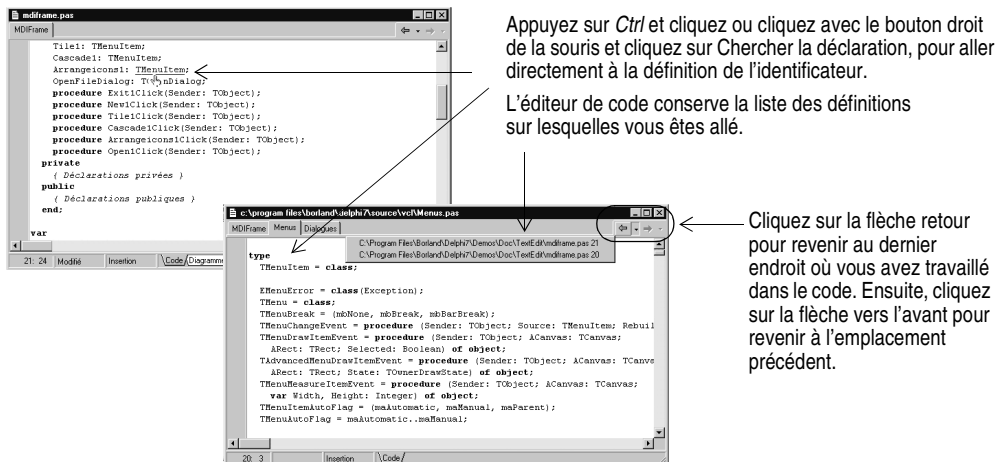
Pour plus d'informations...

Voir "audit de code" et "achèvement de classe" dans l'index de l'aide en ligne.

Navigateur de code

Quand vous faites passer la souris au-dessus d'un nom de classe, de variable, de propriété de méthode ou de tout autre identificateur, le menu surgissant appelé **Audit de symbole** dans conseil indique où l'identificateur est déclaré. Appuyez sur **Ctrl**: le curseur prend la forme d'une main, l'identificateur devient bleu et vous pouvez cliquer dessus pour atteindre sa définition.

L'éditeur de code dispose de boutons **Avant** et **Arrière** similaires à ceux d'un navigateur Web. Quand vous atteignez une définition, l'éditeur de code se souvient de vos déplacements dans le code. Vous pouvez cliquer sur les flèches placées à côté des boutons **Avant** et **Arrière** pour parcourir l'historique de ces références.



Vous pouvez aussi vous déplacer entre la déclaration d'une procédure et son implémentation, en appuyant sur **Ctrl+Maj+↑** ou sur **Ctrl+Maj+↓**.

Pour personnaliser l'environnement de l'éditeur de code, voir ["Personnalisation de l'éditeur de code"](#) à la page 4-11.

Pour plus d'informations...

Voir "éditeur de code" dans l'index de l'aide en ligne.

La page Diagramme

En bas de l'éditeur de code, il peut y avoir un ou plusieurs onglets, en fonction de l'édition de Delphi que vous utilisez. La page **Code** dans laquelle vous écrivez votre code apparaît par défaut au premier plan. La page **Diagramme** affiche des icônes reliées par des lignes représentant les relations entre les composants que vous avez placés dans une fiche ou un module de données. Ces relations s'établissent entre enfants, de parents à enfants ou entre les composants et les propriétés.

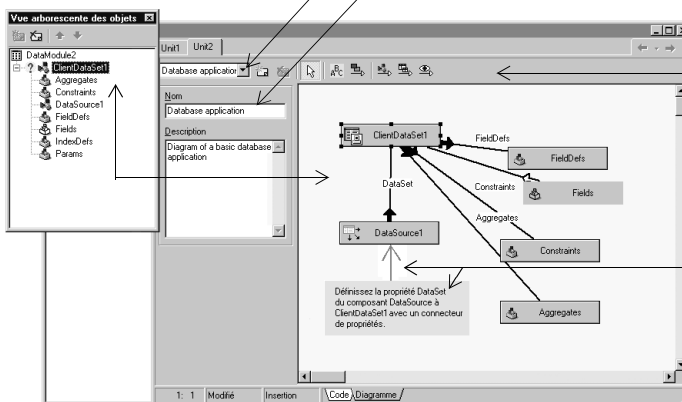
Pour créer un diagramme, cliquez sur la page Diagramme. Depuis l'arborescence d'objets, faites glisser vers la page Diagramme une ou plusieurs icônes pour les organiser verticalement. Pour les organiser horizontalement, appuyez sur *Maj* tout en déplaçant les icônes. Quand vous déplacez des icônes ayant des dépendances de type parent-enfant ou composant-propriété, des lignes (ou *connecteurs*) affichant la relation de dépendance sont automatiquement ajoutées. Si, par exemple, vous ajoutez un composant ensemble de données à un module de données, puis déplacez dans la page Diagramme l'icône ensemble de données plus ses icônes de propriété, le connecteur de propriété relie automatiquement les icônes de propriétés à l'icône de l'ensemble de données.

Si vous voulez afficher une relation de dépendance entre des composants n'ayant pas, utilisez les boutons de la barre d'outils en haut de la page Diagramme pour ajouter l'un des quatre types de connecteurs (bloc de texte, propriété, maître/détail et référence). Vous pouvez également ajouter des blocs de commentaire s'interconnectant ou reliés à une icône particulière.

Depuis l'arborescence des objets, faites glisser les icônes des composants dans la page Diagramme.

Pour voir d'autres diagrammes nommés du projet en cours, choisissez dans la liste déroulante.

Entrez le nom et la description de votre diagramme.



Utilisez les boutons de la barre d'outils de la page Diagramme (Propriété, Maître /détail, Référence) pour désigner la relation entre des composants ou entre des composants et leurs propriétés. Le type de trait change selon le type de relation.

Cliquez sur le bouton Bloc de commentaire pour ajouter un bloc de commentaire et sur le bouton Connecteur de blocs de texte pour établir une connexion avec un autre bloc de commentaire ou une icône.

Vous pouvez attribuer un nom et une description à votre diagramme, l'enregistrer et l'imprimer une fois qu'il est complet.

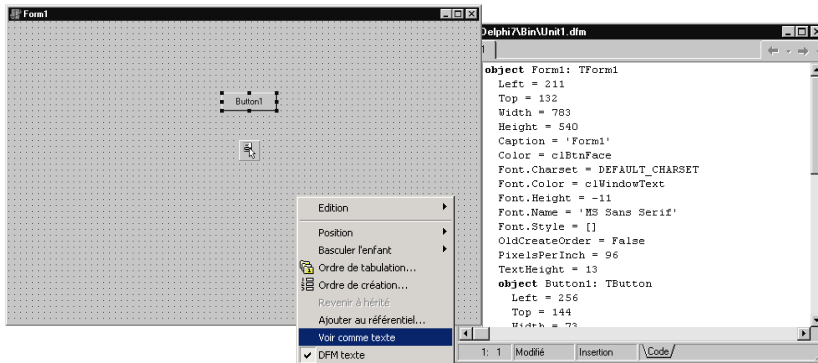
Pour plus d'informations...

Voir "page Diagramme" dans l'index de l'aide en ligne.

Visualisation du code des fiches

Les fiches constituent la partie visible de la majorité des projets Delphi. C'est là que vous concevez l'interface utilisateur d'une application. Normalement, vous concevez les fiches en utilisant les outils visuels de Delphi, et Delphi les stocke dans des fichiers. Les fichiers des fiches (.dfm ou .xfm pour une application CLX) décrivent chacun des composants d'une fiche, y compris la valeur de toutes les propriétés persistantes. Pour voir et modifier le fichier d'une fiche dans

l'éditeur de code, cliquez avec le bouton droit sur la fiche et sélectionnez Voir comme texte. Pour revenir à la vue graphique de la fiche, cliquez avec le bouton droit et choisissez Voir comme fiche.



Utilisez Voir comme texte pour voir dans l'éditeur de code une description des attributs de la fiche au format texte.

Vous pouvez enregistrer les fichiers des fiches soit en format texte (par défaut) soit en format binaire. Choisissez Outils | Options d'environnement, cliquez sur la page Concepteur, et cochez ou non l'option Nouvelles fiches comme texte pour préciser le format à utiliser pour les fiches nouvellement créées.

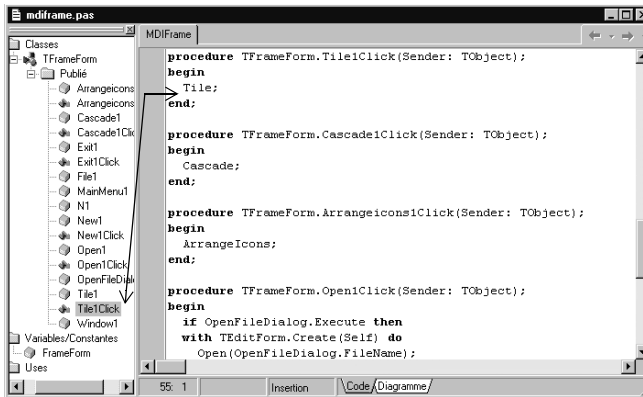
Pour plus d'informations...

Voir "fichiers fiche" dans l'index de l'aide en ligne.

L'explorateur de code

A l'ouverture de Delphi, l'explorateur de code est ancré à gauche de l'éditeur de code (sa disponibilité dépend de l'édition de Delphi dont vous disposez). L'explorateur de code affiche la table des matières sous la forme d'une arborescence du code source ouvert dans l'éditeur de code, en énumérant les types, classes, propriétés, méthodes, variables globales et les routines définies dans l'unité. Il montre également les autres unités citées dans la clause **uses**.

Vous pouvez utiliser l'explorateur de code pour parcourir le contenu de l'éditeur de code. Si par exemple, vous double-cliquez sur un nom de méthode dans l'explorateur de code, le curseur passe dans l'éditeur de code à sa définition dans la partie interface de la déclaration de classe de l'unité.



Double-cliquez sur un élément dans l'explorateur de code, et le curseur se déplace dans l'éditeur de code sur l'implémentation de l'élément. Appuyez sur **Ctrl+Maj+E** pour déplacer le curseur en arrière et en avant entre le dernier emplacement où vous étiez dans l'explorateur de code et dans l'éditeur de code.

Chaque élément de l'explorateur de code est associé à une icône qui représente son type.

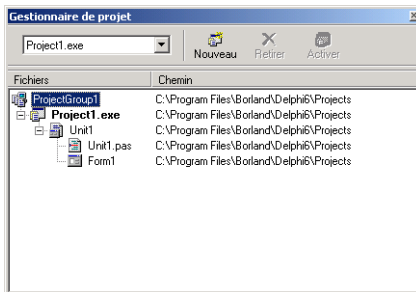
Pour configurer l'affichage du contenu de l'explorateur de code, choisissez Outils|Options d'environnement et sélectionnez l'onglet Explorateur.

Pour plus d'informations...

Voir "explorateur de code" dans l'index de l'aide en ligne.

Le gestionnaire de projet

Lorsque vous démarrez Delphi pour la première fois, il ouvre automatiquement un nouveau projet. Un projet comprend plusieurs fichiers qui constituent l'application ou la DLL que vous développez. Vous pouvez voir et organiser ces fichiers, comme les fichiers fiche, unité, ressource, objet et bibliothèque, dans un outil de gestion des projets appelé le gestionnaire de projet. Pour afficher le gestionnaire de projet, choisissez Voir|Gestionnaire de projet.



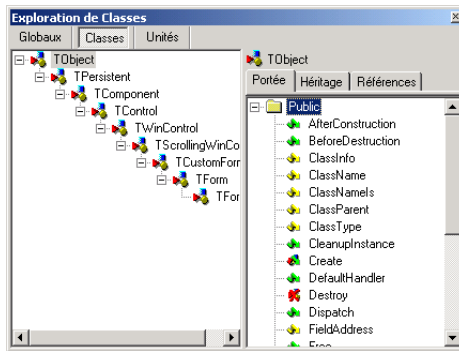
Vous pouvez utiliser le gestionnaire de projet pour combiner et afficher des informations concernant des projets qui sont associés en un *groupe de projets*. En organisant ainsi des projets, par exemple plusieurs exécutables, vous pouvez les compiler en même temps. Pour modifier les options des projets, comme celles de la compilation, reportez-vous à ["Définition des options des projets"](#) à la page 4-8.

Pour plus d'informations...

Voir “gestionnaire de projet” dans l'index de l'aide en ligne.

L'explorateur de projet

L'explorateur de projet examine un projet en détail. L'explorateur affiche sous forme d'arborescence les classes, unités et symboles globaux (types, propriétés, méthodes, variables et routines) déclarés ou utilisés dans votre projet. Choisissez Voir | Explorateur pour afficher l'explorateur de projet.



L'explorateur de projet comporte deux volets redimensionnables : le volet Inspecteur (à gauche) et le volet Détails. L'inspecteur possède trois onglets affichant les symboles globaux, les classes et les unités.

Globaux affiche les classes, types, propriétés, méthodes variables et routines.

Classes montre dans un diagramme la hiérarchie des classes.

Unités affiche les unités, les identificateurs déclarés dans chaque unité et les autres unités qui utilisent chaque unité ou y sont utilisées.

Par défaut, l'explorateur de projet n'affiche que les symboles des unités appartenant au projet en cours. Vous pouvez changer la portée pour afficher tous les symboles disponibles dans Delphi. Choisissez Outils | Options d'environnement et, sur la page Explorateur, cochez Tous les symboles.

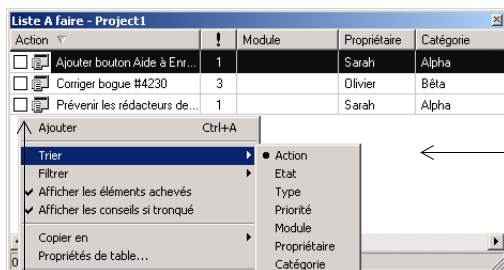
Pour plus d'informations...

Voir “explorateur de projet” dans l'index de l'aide en ligne.

Listes A faire

Les listes A faire enregistrent les éléments à terminer pour un projet. Vous pouvez inclure dans une liste des éléments du niveau du projet en les ajoutant directement à la liste ou vous pouvez ajouter des éléments spécifiques directement dans le code source.

Choisissez Voir | Liste A faire pour ajouter ou voir des informations associées à un projet.



Cliquez avec le bouton droit sur une liste A faire pour afficher les commandes permettant de trier ou de filtrer cette liste.

Cliquez ici lorsque vous en avez terminé avec un élément.

Pour plus d'informations...

Voir "liste A faire" dans l'index de l'aide en ligne.

Programmation avec Delphi

Les sections suivantes fournissent une analyse du développement de logiciels avec Delphi : création d'un projet, travail sur les fiches, écriture de code, compilation, débogage, déploiement et internationalisation des programmes. Elles présentent également les types de projets que vous pouvez développer.

Création d'un projet

Un projet est une collection de fichiers qui sont soit créés au moment de la conception soit générés lorsque vous compilez le code source du projet. Lorsque vous démarrez Delphi pour la première fois, un nouveau projet est ouvert. Cela génère automatiquement, entre autres fichiers, un fichier projet (Project1.dpr), un fichier unité (Unit1.pas) et un fichier ressource (Unit1.dfm, Unit1.xfm pour les applications CLX).

Si un projet est déjà ouvert et que vous voulez en ouvrir un nouveau, choisissez Fichier | Nouveau | Application, ou choisissez Fichier | Nouveau | Autre et double-cliquez sur l'icône Application. Fichier | Nouveau | Autre ouvre un référentiel d'objets qui fournit les fiches, les cadres et les modules, ainsi que les modèles pré-fabriqués comme les boîtes de dialogue, que vous pouvez ajouter à votre projet. Pour en savoir plus sur le référentiel d'objets, voir ["Le référentiel d'objets"](#) à la page 2-6.

Lorsque vous démarrez un projet, vous devez savoir ce que vous voulez développer, par exemple une application ou une DLL. Pour des informations sur les types de projets pouvant être développés avec Delphi, voir ["Types de projets"](#) à la page 3-9.

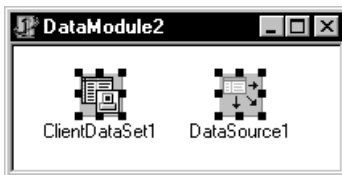
Pour plus d'informations...

Voir "projets" dans l'index de l'aide en ligne.

Ajout de modules de données

Un module de données est une sorte de fiche ne contenant que des composants non visuels. Les composants non visuels d'un module de données *peuvent* être placés sur des fiches normales, à côté de composants visuels. Mais, si vous prévoyez de réutiliser des groupes de bases de données et d'objets système, ou si vous voulez séparer les parties de votre application concernant la connexion aux bases de données et les règles de gestion, les modules de données offrent un outil d'organisation pratique.

Pour créer un module de données, choisissez Fichier | Nouveau | Module de données. Delphi ouvre un module de données vide, ce qui affiche dans l'éditeur de code un fichier unité supplémentaire, et ajoute le module au projet en cours en tant que nouvelle unité. Pour ajouter des composants non visuels à un module de données, procédez comme avec une fiche.



Double-cliquez sur un composant non visuel de la palette de composants pour le placer dans le module de données.

Lorsque vous rouvrez un module de données existant, Delphi affiche ses composants.

Pour plus d'informations...

Voir “modules de données” dans l'index de l'aide.

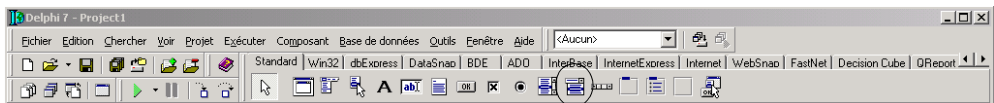
Construction de l'interface utilisateur

Avec Delphi, vous devez d'abord créer une interface utilisateur (IU) en sélectionnant des composants dans la palette de composants et en les plaçant sur la fiche principale.

Positionnement des composants sur une fiche

Pour placer des composants sur une fiche :

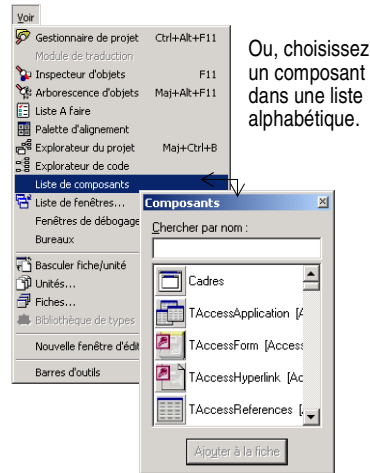
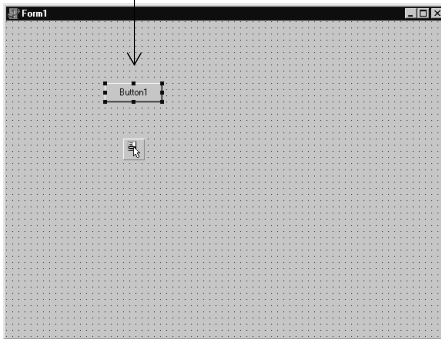
- 1 Double-cliquez sur le composant ou
- 2 Cliquez une seule fois sur le composant puis sur la fiche à l'endroit où vous voulez le faire apparaître.



Cliquez sur un composant dans la palette de composants

Sélectionnez le composant et faites-le glisser là où vous voulez le faire apparaître sur la fiche.

Cliquez ensuite sur l'endroit où vous voulez le positionner sur la fiche.

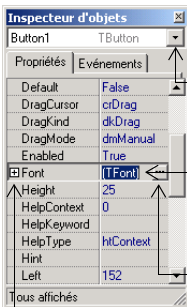


Pour plus d'informations...

Voir "palette de composants" dans l'index de l'aide en ligne.

Définition des propriétés des composants

Après avoir placé des composants sur une fiche, définissez leurs propriétés et codez leurs gestionnaires d'événements. En définissant les propriétés d'un composant, vous changez la façon dont celui-ci apparaît et se comporte dans votre application. Lorsqu'un composant est sélectionné sur une fiche, ses propriétés et ses événements sont affichés dans l'inspecteur d'objets.



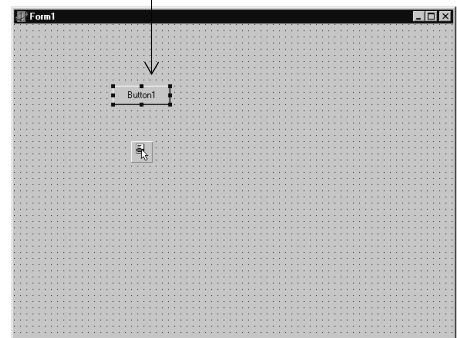
Ou utilisez cette liste déroulante pour sélectionner un objet. Ici, Button1 est sélectionné et ses propriétés sont affichées.

Sélectionnez une propriété et changez sa valeur dans la colonne de droite.

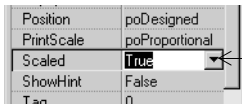
Cliquez sur les points de suspension pour ouvrir une boîte de dialogue permettant de modifier les propriétés d'un objet utilitaire.

Vous pouvez aussi cliquer sur le signe plus pour ouvrir une liste de détails.

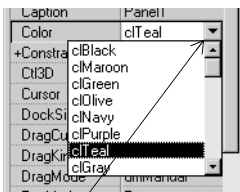
Vous pouvez sélectionner un composant, ou objet, sur la fiche en cliquant dessus.



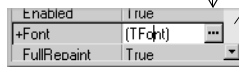
De nombreuses propriétés possèdent des valeurs simples, comme les noms des couleurs, *True* ou *False*, et des entiers. Pour les propriétés booléennes, vous pouvez double-cliquer sur le mot pour passer alternativement de *True* à *False*. Certaines propriétés sont associées à des éditeurs permettant de définir des valeurs plus complexes. Lorsque vous cliquez sur la valeur d'une de ces propriétés, des points de suspension apparaissent. Pour certaines propriétés, comme la taille, il faut entrer une valeur.



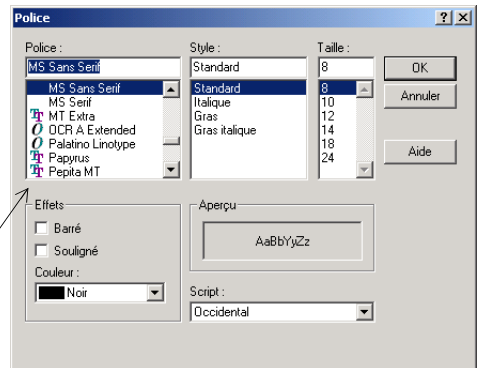
Double-cliquez ici pour changer la valeur *True* en *False*.



Cliquez sur les points de suspension pour afficher l'éditeur de cette propriété.

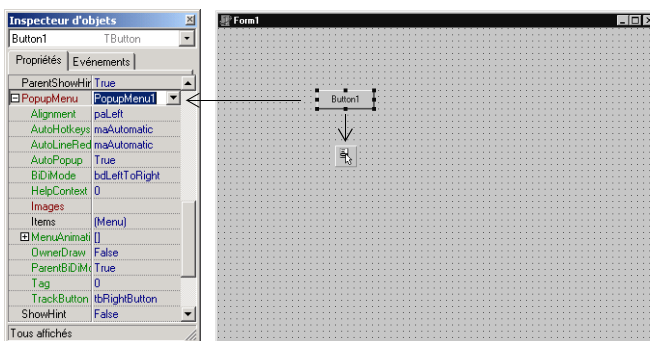


Cliquez sur la flèche vers le bas pour faire votre choix parmi les valeurs valides de la propriété.



Lorsque plusieurs composants sont sélectionnés dans la fiche, l'inspecteur d'objets affiche toutes les propriétés partagées par ces composants.

L'inspecteur d'objets permet également de développer en ligne les références à un composant. Cela permet d'accéder aux propriétés et événements d'un composant référencé sans avoir à le sélectionner. Si, par exemple, vous ajoutez un bouton et un composant menu surgissant dans une fiche, vous pouvez sélectionner le composant bouton et, dans l'inspecteur d'objets, initialiser sa propriété *PopupMenu* à *PopupMenu1* pour afficher toutes les propriétés du menu surgissant.



Initialisez la propriété *PopupMenu* du composant bouton à *PopupMenu1* : toutes les propriétés du menu surgissant apparaissent quand vous cliquez sur le signe plus (+).

Les références en ligne aux composants sont affichées en rouge et les sous-propriétés en vert.

Pour plus d'informations...

Voir "inspecteur d'objets" dans l'index de l'aide en ligne.

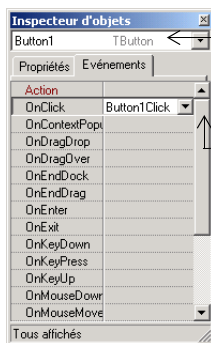
Ecriture du code

La partie fondamentale de toute application est le code associé à chaque composant. Bien que l'environnement RAD de Delphi apporte la majorité des blocs de construction de vos applications, comme les composants visuels et non visuels pré-empaquetés, il vous faudra fréquemment écrire des gestionnaires d'événements et, sans doute, certaines de vos propres classes. Pour vous aider dans cette tâche, les milliers d'objets de la bibliothèque de classes sont à votre disposition. Pour apprendre à voir et à modifier votre code source, voir ["L'éditeur de code" à la page 2-7](#).

Ecriture de gestionnaires d'événements

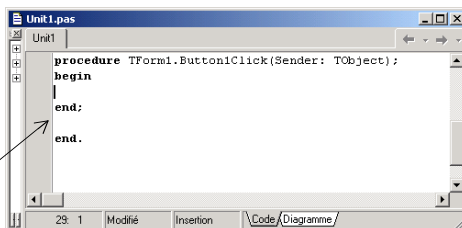
Votre code doit répondre aux événements qui se produisent pour un composant au moment de l'exécution. Un événement est le lien établi entre une occurrence du système, comme le clic d'un bouton, et une portion de code qui répond à cette occurrence. Le code qui répond est le gestionnaire de l'événement. Ce code modifie les valeurs des propriétés et appelle des méthodes.

Pour voir les gestionnaires d'événements prédéfinis d'un composant de votre fiche, sélectionnez le composant et, dans l'inspecteur d'objets, cliquez sur l'onglet Événements.



Ici, Button1 est sélectionné et son type est affiché : *TButton*. Cliquez sur l'onglet Événements dans l'inspecteur d'objets pour voir les événements que le composant Button peut gérer.

Sélectionnez un gestionnaire d'événement existant dans la liste déroulante. Ou double-cliquez sur la colonne de la valeur pour que Delphi génère le squelette du code d'un nouveau gestionnaire.



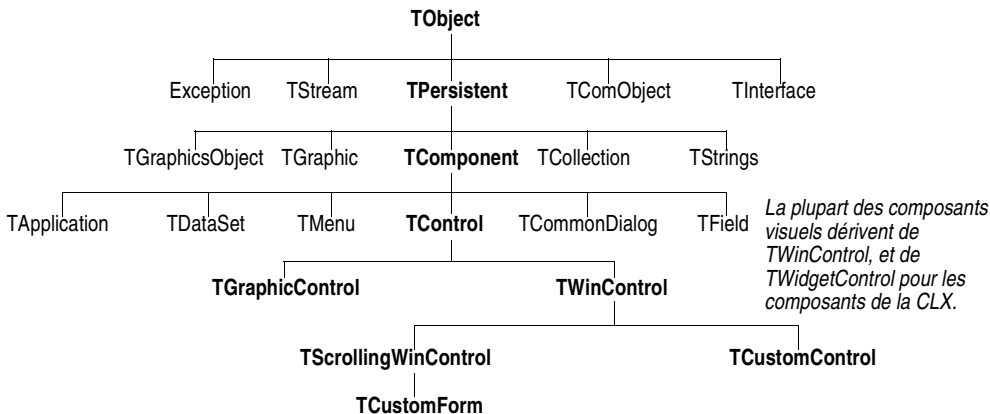
Pour plus d'informations...

Voir "événements" dans l'index de l'aide en ligne.

Utilisation de la bibliothèque de composants

Delphi est fourni avec une bibliothèque constituée d'objets, certains étant des composants ou des contrôles, que vous pouvez utiliser pour écrire du code. Vous pouvez utiliser les composants de la VCL pour les applications Windows et ceux de la CLX pour les applications Windows et Linux. La bibliothèque de composants contient des objets visibles à l'exécution, comme les contrôles de saisie, les boutons et les autres éléments d'interface utilisateur, ainsi que des contrôles non visuels comme les ensembles de données ou les timers.

Le diagramme suivant montre certaines des classes principales constituant la hiérarchie de la VCL. La hiérarchie de la CLX est similaire.



Les objets dérivés de *TComponent* possèdent les propriétés et les méthodes autorisant leur installation dans la palette de composants et leur ajout aux fiches et aux modules de données Delphi. Les composants étant intégrés à l'EDI, vous pouvez utiliser des outils comme le concepteur de fiche pour développer des applications très rapidement.

Les composants apportent un haut degré d'encapsulation. Par exemple, les boutons sont pré-programmés pour répondre aux clics de souris en déclenchant des événements *OnClick*. Si vous utilisez un contrôle bouton, vous n'avez pas à écrire de code gérant les événements générés lors d'un clic sur le bouton ; vous n'êtes chargé que de la logique de l'application qui doit s'exécuter en réponse au clic lui-même.

La plupart des éditions de Delphi sont livrées avec le code source des composants de la bibliothèque et avec des exemples de techniques de programmation Delphi.

Pour plus d'informations...

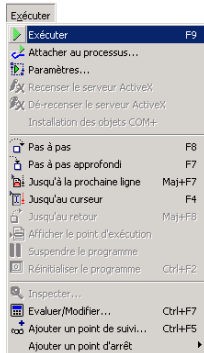
Voir "Référence de la bibliothèque de composants visuels" et "Référence CLX" dans le sommaire de l'aide et "VCL" dans l'index de l'aide en ligne. Voir à l'adresse <http://www.borland.com/delphi> des options pour CLX à sources libres ou sous licence.

Compilation et débogage des projets

Après avoir écrit votre code, vous devez compiler et déboguer votre projet. Avec Delphi, vous pouvez soit compiler votre projet en premier et le déboguer ensuite indépendamment, soit le compiler et le déboguer en une seule étape en utilisant le débogueur intégré. Pour compiler votre programme en incluant les informations de débogage, choisissez *Projet | Options*, cliquez sur la page *Compilateur*, cochez *Informations de débogage*.

Delphi utilise un débogueur intégré qui permet de contrôler l'exécution du programme, suivre les variables et modifier les valeurs des données. Vous pouvez avancer dans votre code ligne par ligne, en examinant l'état du programme à chaque point d'arrêt. Pour vous servir du débogueur intégré, choisissez Outils | Options du débogueur, cliquez sur l'onglet Général et vérifiez que la case Débogage intégré est cochée.

Vous pouvez commencer une session de débogage dans l'EDI en cliquant sur le bouton Exécuter, en choisissant Exécuter | Exécuter ou en appuyant sur **F9**.

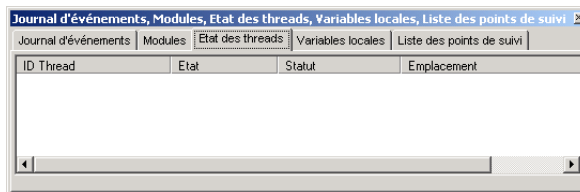


Choisissez une des commandes de débogage dans le menu Exécuter. Certaines commandes sont aussi accessibles dans la barre d'outils



Bouton Exécuter

Le débogueur intégré met à votre disposition de nombreuses fenêtres de débogage, notamment les fenêtres Points d'arrêt, Pile d'appels, Points de suivi, Variables locales, Threads, Modules, CPU et Journal d'événements. Affichez-les en choisissant Voir | Fenêtres de débogage. Toutes les vues du débogueur ne sont pas disponibles dans toutes les éditions de Delphi.



Pour simplifier l'utilisation, vous pouvez combiner plusieurs fenêtres de débogage.

Pour savoir comment combiner les fenêtres de débogage pour en faciliter l'utilisation, voir [“Ancrage des fenêtres d'outils” à la page 4-2](#).

Lorsque vous avez défini votre bureau tel que vous voulez qu'il fonctionne pendant le débogage, vous pouvez enregistrer ces paramètres en tant que bureau de débogage ou d'exécution. Cette disposition du bureau sera utilisée chaque fois que vous déboguerez une application. Pour plus de détails, voir [“Enregistrement des dispositions du bureau” à la page 4-4](#).

Pour plus d'informations...

Voir “débogage” et “utilisation du débogueur intégré” dans l'index de l'aide.

Déploiement des applications

Si vous voulez que votre application puisse être installée et exécutée par d'autres personnes, vous devez préalablement la déployer. Pour déployer une application, créez un paquet d'installation incluant non seulement les fichiers indispensables comme les exécutables, mais aussi tous les fichiers de support, les DLL, les fichiers paquet ou les applications complémentaires. Delphi est fourni avec un utilitaire d'installation, InstallShield Express, qui vous permet de créer un programme d'installation avec ces fichiers. Pour installer InstallShield Express, choisissez l'édition personnalisée de InstallShield Express pour Delphi, dans l'écran d'installation de Delphi.

Pour plus d'informations...

Voir "déploiement, applications" dans l'index de l'aide en ligne.

Internationalisation d'applications

Delphi offre de nombreuses fonctionnalités consacrées à l'internationalisation et à la localisation des applications. L'EDI et la VCL apportent le support des éditeurs de méthodes de saisie (IME) et des jeux de caractères étendus pour internationaliser votre projet. Avec certaines éditions de Delphi, vous disposez d'un environnement de traduction qui permet la localisation des logiciels et le développement simultané dans plusieurs langues. L'environnement de traduction vous permet de gérer plusieurs versions localisées d'une application comme des parties d'un même projet.

L'environnement de traduction comprend trois outils intégrés :

- L'expert DLL de ressources, un expert DLL qui génère et gère les DLL de ressources.
- Le module de traduction, une grille pour visualiser et modifier les ressources traduites.
- Le référentiel de traduction, base de données partageable contenant les traductions.

Pour ouvrir l'expert DLL de ressources, choisissez Fichier | Nouveau | Autre et double-cliquez sur l'icône Expert DLL de ressources. Pour configurer les outils de traduction, choisissez Outils | Options des outils de traduction.

Pour plus d'informations...

Voir "applications internationales" dans l'index de l'aide en ligne.

Types de projets

Toutes les éditions de Delphi gèrent la programmation générale Windows 32 bits, les DLL, les paquets, les composants personnalisés, le multithreading, les contrôleurs COM et automation, ainsi que le débogage multiprocesseurs. Certaines éditions supportent les applications serveur comme les applications serveur Web, les applications de bases de données, les serveurs COM, les applications multiniveaux, CORBA et les systèmes d'aide à la décision.

Pour plus d'informations...

Pour connaître les outils gérés par l'édition dont vous disposez, voir la liste des caractéristiques à l'adresse www.borland.fr/delphi.

Applications CLX

Vous pouvez utiliser Delphi pour développer des applications 32 bits multiplates-formes qui s'exécuteront sur les systèmes d'exploitation Windows et Linux. Pour développer une application CLX, choisissez Fichier | Nouveau | Application CLX. L'EDI est similaire à l'environnement d'une application Delphi normale, à cette différence que n'apparaissent dans la palette des composants et dans le référentiel d'objets que les éléments utilisables dans une application CLX. Les fonctionnalités Windows gérées par Delphi ne peuvent pas être portées directement dans un environnement Linux.

Pour plus d'informations...

Pour connaître les composants disponibles dans les applications multiplates-formes, voir "Référence CLX" dans le sommaire de l'aide.

Applications serveur Web

Une application serveur Web fonctionne avec un serveur Web en traitant les requêtes d'un client et en renvoyant un message HTTP sous la forme d'une page Web. Pour publier des données sur le Web, Delphi propose deux technologies différentes, selon l'édition de Delphi dont vous disposez.

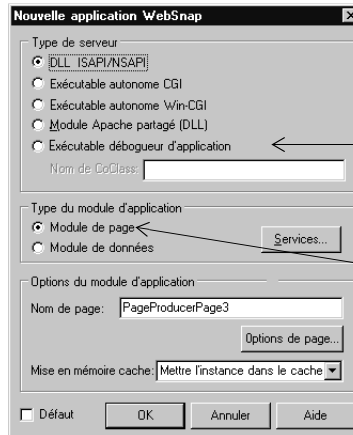
La technologie d'application serveur Web la plus ancienne de Delphi est l'agent Web (Web Broker). Les applications agent Web peuvent répartir des requêtes, effectuer des actions et renvoyer des pages Web aux utilisateurs. La plus grande partie de la logique métier d'une application est définie dans des gestionnaires d'événements écrits par le développeur de l'application. Pour créer une application serveur agent Web, choisissez Fichier | Nouveau | Autre et double-cliquez sur l'icône Application Serveur Web. Vous pouvez ajouter à votre module Web des composants issus des pages Internet et InternetExpress de la palette de composants.

WebSnap apporte en plus des adaptateurs, des répartiteurs et des générateurs de page supplémentaires, la gestion des sessions et des modules de page Web.

Ces fonctionnalités supplémentaires ont été conçues pour gérer automatiquement les tâches courantes des applications serveur Web. Le développement WebSnap est plus simple et plus visuel que le développement Web Broker. Un développeur d'applications WebSnap peut consacrer davantage de temps à la conception de la logique métier de l'application, plutôt qu'à l'écriture de gestionnaires d'événements pour les tâches communes de transfert de pages. Pour créer une nouvelle application serveur WebSnap, choisissez **Fichier | Nouveau | Autre**, sélectionnez la page WebSnap et double-cliquez sur l'icône Application WebSnap. Vous pouvez ajouter des composants WebSnap à partir de la page WebSnap de la palette de composants.



Vous pouvez également accéder au module de données d'application WebSnap en choisissant **VoirlBarres d'outils | Internet**, et en cliquant sur l'icône Nouvelle application WebSnap.



Vous pouvez créer une application s'exécutant sur divers types de serveurs Web, y compris un serveur de test qui vous permet de déboguer votre application serveur Web. Décidez si vous souhaitez un module de données ou un module de page qui affiche votre page HTML.

Pour plus d'informations...

Voir "applications Web" dans l'index de l'aide en ligne.

Applications de bases de données

Delphi apporte une grande variété d'outils de bases de données et de connectivité qui simplifient le développement des applications de bases de données.

Pour créer une application de base de données, commencez par concevoir l'interface d'une fiche en utilisant les composants de page ContrôleBD. Puis, ajoutez une source de données dans un module de données en utilisant la page AccèsBD. Enfin, pour vous connecter à divers serveurs de bases de données, ajoutez un composant ensemble de données ou connexion de données au module de données depuis les pages déjà citées de la palette ou depuis celles associées aux outils de connectivité suivants :

- dbExpress est un ensemble de pilotes utilisés par des applications multiplates-formes donnant un accès rapide à des serveurs SQL comme DB2, Informix, InterBase, MSSQL, MySQL et Oracle. Avec un pilote dbExpress,

vous pouvez accéder aux bases de données en utilisant des ensembles de données unidirectionnels.

- Le BDE (Borland Database Engine, moteur de bases de données) est un ensemble de pilotes supportant de nombreux formats de bases de données parmi les plus utilisés, y compris dBASE, Paradox, FoxPro, Microsoft Access, et toute source de données ODBC.
- ADO (ActiveX Data Objects) est une interface Microsoft de haut niveau pour toute source de données, dont les bases de données relationnelles ou non, les systèmes de messagerie électronique et les systèmes de fichiers, le texte et les graphiques, ainsi que les objets de gestion personnalisés.
- Les composants InterBase Express (IBX) sont basés sur l'architecture de composants d'accès aux données personnalisés de Delphi. Les applications IBX donnent accès aux fonctionnalités avancées d'InterBase et offrent une interface de composants ultra-performante pour InterBase 5.5 et ses versions ultérieures. IBX est compatible avec la bibliothèque Delphi de composants orientés données.

Certains outils de connectivité aux bases de données ne sont pas proposés dans toutes les éditions de Delphi.

Pour plus d'informations...

Voir "applications de bases de données" dans l'index de l'aide en ligne.

Administrateur BDE

L'administrateur BDE (BDEAdmin.exe) permet de configurer les pilotes BDE et de définir les alias employés par les contrôles orientés données de la VCL pour se connecter aux bases de données.

Pour plus d'informations...

A partir du menu Démarrer de Windows, choisissez Programmes | Borland Delphi | Administrateur BDE. Choisissez ensuite Aide | Index.

Explorateur SQL (explorateur de base de données)

L'explorateur SQL (DBExplor.exe) vous permet de parcourir et de modifier les bases de données. Vous pouvez l'utiliser pour créer des alias, voir les informations de schéma, exécuter des requêtes SQL et maintenir les dictionnaires de données ou les jeux d'attributs.

Pour plus d'informations...

Depuis le menu principal de Delphi, choisissez Base de données | Explorateur pour ouvrir l'explorateur. Choisissez ensuite Aide | Index. Ou recherchez "explorateur de base de données" dans l'index principal de l'aide.

Module Base de données

Le module Base de données (DBD32.exe) vous permet de créer, de voir et de modifier les tables des bases de données Paradox et dBase dans divers formats.

Pour plus d'informations...

A partir du menu Démarrer de Windows, choisissez Programmes | Borland Delphi | Module Base de données. Puis choisissez Aide | Guide de l'utilisateur.

Dictionnaire de données

Le dictionnaire de données fournit un espace de stockage personnalisable indépendant de vos applications dans lequel vous pouvez créer des jeux d'attributs de champs étendus décrivant le contenu et l'aspect des données. Le dictionnaire de données peut résider sur un serveur distant pour partager encore plus d'informations.

Pour plus d'informations...

Choisissez Aide | Outils Delphi pour voir "Dictionnaire des données".

Composants personnalisés

Les composants livrés avec Delphi sont pré-installés dans la palette de composants et proposent des fonctionnalités suffisamment étendues pour répondre à la majorité des besoins de développement. Vous pourriez programmer avec Delphi pendant des années sans avoir besoin d'installer de nouveaux composants. Mais, parfois pour résoudre certains problèmes spéciaux ou encapsuler des types de comportement très particuliers, la contribution des composants personnalisés devient nécessaire. Les composants personnalisés favorisent la réutilisation du code et la cohérence des applications.

Vous pouvez installer des composants personnalisés fournis par des tiers ou créer les vôtres. Pour créer un nouveau composant, choisissez Composant | Nouveau composant pour afficher l'expert Nouveau composant. Pour installer des composants fournis par un tiers, voir "[Installation de paquets de composants](#)" à la page 4-6.

Pour plus d'informations...

Voir le *Guide du concepteur de composants* et "composants, création" dans l'index de l'aide en ligne.

DLL

Les DLL (bibliothèques de liaison dynamique) sont des modules compilés contenant des routines pouvant être appelées par les applications et d'autres DLL. Une DLL contenant du code et des ressources partageables, elle est habituellement utilisée par plusieurs applications. Pour créer un modèle de DLL, choisissez Fichier | Nouveau | Autre et double-cliquez sur l'icône Expert DLL.

Pour plus d'informations...

Voir "DLL" dans l'index de l'aide en ligne.

COM et ActiveX

Delphi supporte le standard COM de Microsoft et offre des experts facilitant la création des contrôles ActiveX. Choisissez Fichier | Nouveau | Autre et cliquez sur l'onglet ActiveX pour accéder aux experts. Des exemples de contrôles ActiveX sont installés dans la page ActiveX de la palette de composants. De nombreux composants serveur COM sont proposés dans l'onglet Serveurs de la palette de composants. Vous pouvez les utiliser comme s'il s'agissait de composants de la VCL. Par exemple, vous pouvez placer un des composants Microsoft Word sur une fiche, afin d'offrir une instance de Microsoft Word à l'interface d'une application.

Pour plus d'informations...

Voir "COM" et "ActiveX" dans l'index de l'aide en ligne.

Bibliothèques de types

Les bibliothèques de types sont des fichiers contenant des informations sur les types de données, les interfaces, les fonctions membres et les classes d'objets présentées dans un contrôle ou un serveur ActiveX. En incluant une bibliothèque de types à votre application COM ou à votre bibliothèque ActiveX, vous mettez ces informations à la disposition d'autres applications et d'autres outils de programmation. Delphi possède un éditeur de bibliothèques de types qui permet la création et la maintenance de celles-ci.

Pour plus d'informations...

Voir "bibliothèque de types" dans l'index de l'aide en ligne.

Personnalisation du bureau

Ce chapitre explique quelques-unes des méthodes permettant de personnaliser l'EDI de Delphi.

Organisation de votre espace de travail

L'EDI disposant de nombreux outils pour le développement, vous aurez envie de réorganiser votre aire de travail pour un confort maximal. Vous pouvez agencer les menus et les barres d'outils, combiner les fenêtres d'outils et enregistrer la nouvelle disposition du bureau.

Agencement des menus et des barres d'outils

Dans la fenêtre principale, vous pouvez réorganiser le menu, les barres d'outils et la palette de composants en cliquant sur la poignée située à leur gauche et en les faisant glisser jusqu'à un autre emplacement.

Vous pouvez déplacer menus et barres d'outils au sein de la fenêtre principale. Faites glisser la poignée (la double barre située à gauche) d'une barre d'outils pour la déplacer.

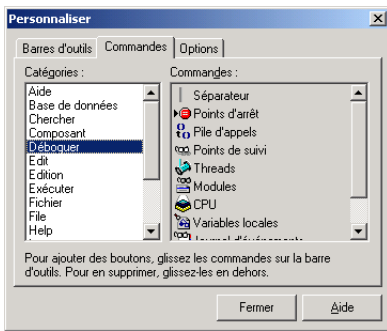


Vous pouvez séparer des éléments de la fenêtre principale et les placer ailleurs sur l'écran ou les supprimer complètement du bureau. C'est utile lorsque vous avez une double configuration de moniteur.



Fenêtre principale organisée différemment.

Vous pouvez ajouter des outils aux barres d'outils, ou en supprimer, en choisissant Voir | Barres d'outils | Personnaliser. Cliquez sur la page Commandes, sélectionnez une catégorie, sélectionnez une commande et faites-la glisser dans la barre d'outils où vous voulez la voir apparaître.



Dans la page Commandes, sélectionnez une commande et faites-la glisser dans une barre d'outils.

Dans la page Options, cochez l'option Afficher les conseils pour que les conseils apparaissent pour les composants et les icônes de la barre d'outils.

Pour plus d'informations...

Voir "barres d'outils, personnaliser" dans l'index de l'aide en ligne.

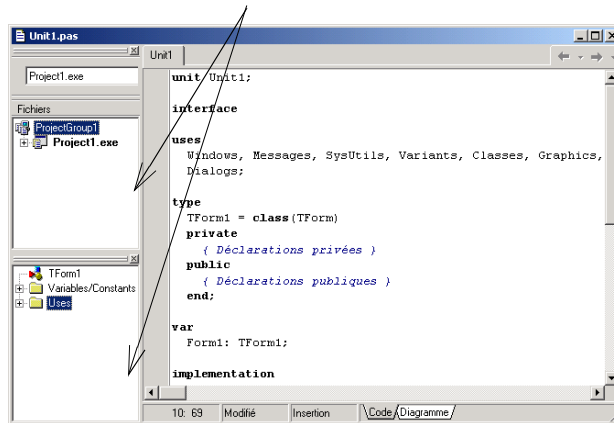
Ancrage des fenêtres d'outils

Vous pouvez ouvrir et fermer les fenêtres d'outils individuellement et les agencer sur le bureau à votre convenance. Nombre d'entre elles peuvent également être *ancrées* à une autre pour faciliter leur manipulation. L'ancrage (qui signifie soit attacher les fenêtres les unes aux autres afin qu'elles se déplacent ensemble, soit combiner plusieurs fenêtres en un "classeur" à onglets unique) vous permet de gérer l'espace de l'écran avec efficacité tout en bénéficiant d'un accès rapide aux différents outils.

Dans le menu Voir, vous pouvez faire apparaître n'importe quelle fenêtre d'outil et l'ancrer directement à une autre. Par exemple, lorsque vous ouvrez pour la première fois Delphi dans sa configuration par défaut, l'explorateur de code est ancré à gauche de l'éditeur de code. Vous pouvez ajouter le gestionnaire de projet aux deux premiers et créer ainsi trois fenêtres ancrées ensemble.

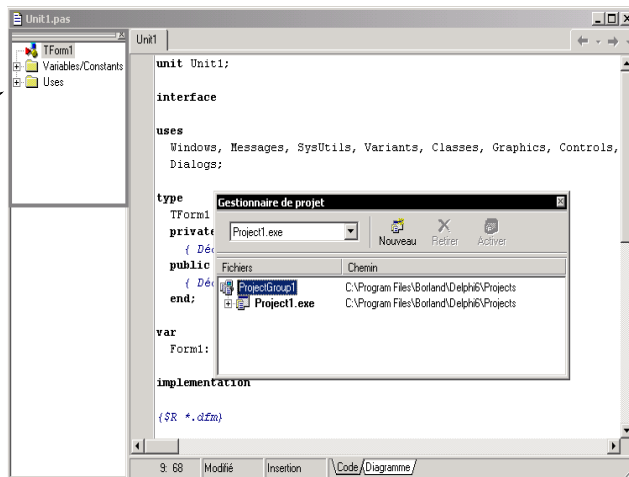
Ici, le gestionnaire de projet et l'explorateur de code sont ancrés à l'éditeur de code.

Vous pouvez combiner, ou "ancrer" des fenêtres, soit avec les poignées, comme à droite, soit avec les onglets, comme à la [page 4-4](#).

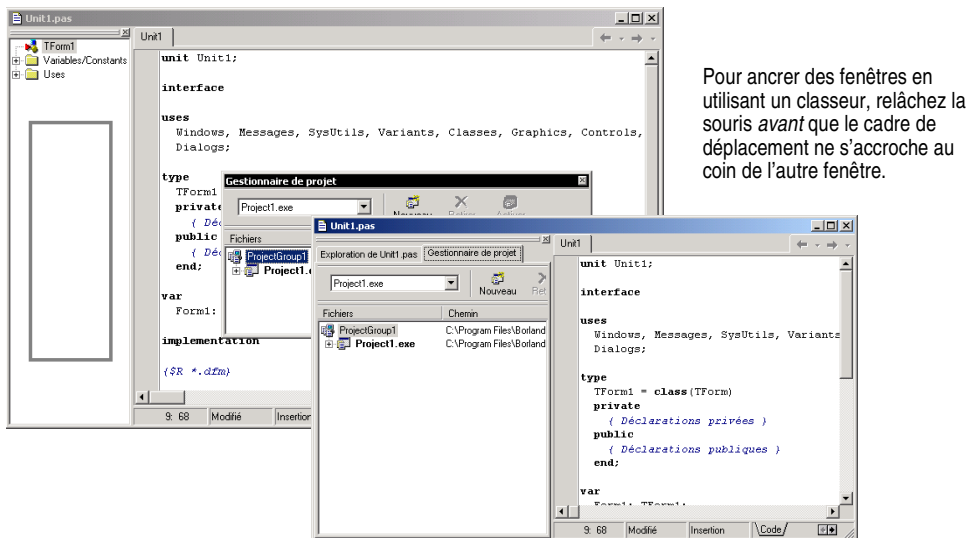


Pour ancrer une fenêtre, cliquez sur sa barre d'outils et faites-la glisser par dessus une autre fenêtre. Lorsque le cadre montrant l'emplacement de la première fenêtre devient un rectangle étroit et s'accroche à un coin de la seconde, relâchez la souris. Les deux fenêtres sont ancrées.

Pour ancrer des fenêtres en utilisant les poignées, relâchez la souris lorsque le cadre de déplacement s'accroche au coin de l'autre fenêtre.



Vous pouvez également ancrer les outils entre eux et former une fenêtre à onglets.



Pour désancrer une fenêtre, double-cliquez sur sa poignée ou sur son onglet.

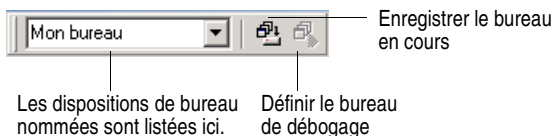
Pour désactiver l'ancrage automatique, appuyez sur la touche *Ctrl* tout en déplaçant une fenêtre ou choisissez Outils | Options d'environnement, sélectionnez la page Préférences et désélectionnez l'option Ancrage automatique par glissement.

Pour plus d'informations...

Voir "ancrage" dans l'index de l'aide.

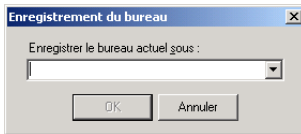
Enregistrement des dispositions du bureau

Vous pouvez personnaliser et enregistrer les dispositions du bureau. La barre d'outils Bureau, dans l'EDI, offre une liste de sélection des dispositions de bureau disponibles et deux icônes facilitant la personnalisation du bureau.



Agencez le bureau à votre convenance : affichez, redimensionnez et ancrez certaines fenêtres, et placez-les à l'endroit voulu sur l'écran.

Dans la barre d'outils Bureaux, cliquez sur l'icône Enregistrer le bureau en cours ou choisissez Voir | Bureaux | Enregistrer le bureau, et entrez un nom pour la nouvelle disposition.



Entrez un nom pour la disposition de bureau à enregistrer et cliquez sur OK.

Pour plus d'informations...

Voir "disposition du bureau" dans l'index de l'aide.

Personnalisation de la palette de composants

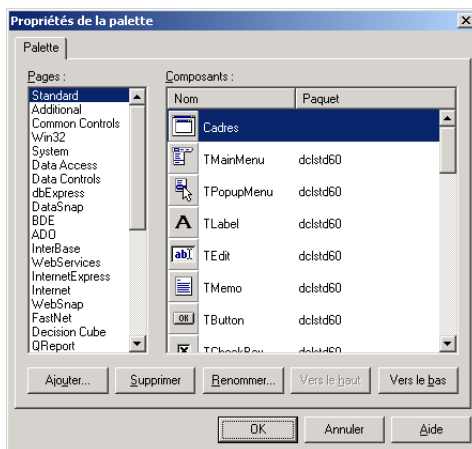
Dans sa configuration par défaut, la palette de composants affiche de nombreux objets utiles, regroupés de manière fonctionnelle en plusieurs pages à onglets. Vous pouvez personnaliser la palette de composants et :

- Cacher ou réorganiser des composants.
- Ajouter, supprimer, réorganiser ou renommer des pages.
- Créer des modèles de composants et les ajouter à la palette.
- Installer de nouveaux composants.

Organisation de la palette de composants

Pour ajouter, supprimer, réorganiser ou renommer des pages, ou pour cacher ou réorganiser des composants, utilisez la boîte de dialogue Propriétés de la palette. Vous pouvez ouvrir cette boîte de dialogue de différentes façons :

- Choisissez Composant | Configurer la palette.
- Choisissez Outils | Options d'environnement et cliquez sur l'onglet Palette.
- Cliquez avec le bouton droit sur la palette de composants et sélectionnez Propriétés.



Vous pouvez réorganiser la palette et lui ajouter de nouvelles pages.

Pour plus d'informations...

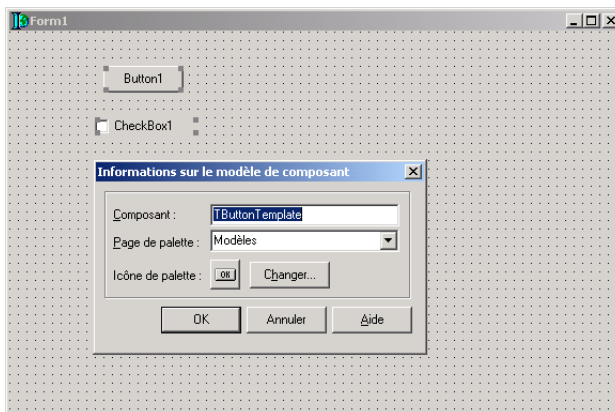
Cliquer sur le bouton Aide de la boîte de dialogue Propriétés de la palette.

Création de modèles de composants

Les modèles de composants sont des groupes de composants que vous ajoutez à une fiche en une seule opération. Les modèles vous permettent de configurer des composants sur une fiche, puis de sauvegarder leur agencement, leurs propriétés par défaut et leurs gestionnaires d'événements dans la palette de composants, afin de les réutiliser dans d'autres fiches.

Pour créer un modèle de composants, disposez simplement un ou plusieurs composants dans une fiche, définissez leurs propriétés dans l'inspecteur d'objets, puis sélectionnez-les tous en faisant glisser la souris autour d'eux. Choisissez ensuite Composant | Créer un modèle de composant. Lorsque s'ouvre la boîte de dialogue Informations sur le modèle de composant, vous pouvez choisir un nom pour le modèle, la page de la palette sur laquelle il doit apparaître et l'icône devant le représenter dans celle-ci.

Après avoir positionné le modèle sur une fiche, vous pouvez repositionner les composants indépendamment les uns des autres, réinitialiser leurs propriétés, et créer ou modifier des gestionnaires d'événements comme si vous aviez positionné chaque composant séparément.



Pour plus d'informations...

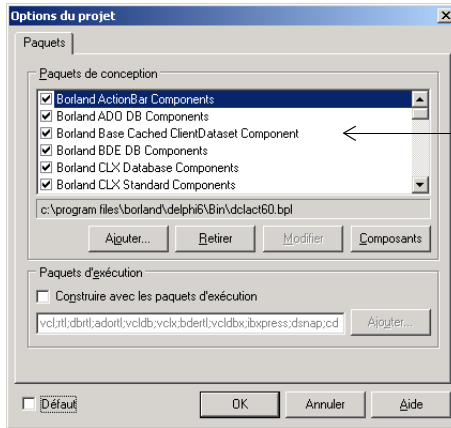
Voir "modèles de composants" dans l'index de l'aide en ligne.

Installation de paquets de composants

Que vous écriviez vos propres composants personnalisés ou les obteniez d'un fournisseur extérieur, les composants doivent être compilés en *paquet* avant d'être installés dans la palette de composants.

Un paquet est une DLL spéciale contenant du code partageable par les applications Delphi, l'EDI, ou les deux. Les *paquets d'exécution* fournissent des fonctionnalités lorsqu'un utilisateur exécute une application. Les *paquets de conception* sont utilisés pour installer des composants dans l'EDI. Les paquets Delphi ont l'extension .bpl.

Lorsque des composants de développeurs tiers sont déjà compilés en paquet, suivez les instructions de ce fournisseur ou choisissez Composant | Installer des paquets.



Ces composants sont pré-installés dans Delphi. Lorsque vous installez de nouveaux composants provenant de fournisseurs tiers, leur paquet apparaît dans cette liste.

Cliquez sur Composants pour voir les composants contenus dans le paquet.

Pour plus d'informations...

Voir "installation des composants" et "paquets" dans l'index de l'aide.

Utilisation des cadres

Un cadre (*TFrame*) sert, comme une fiche, de conteneur à des composants que vous voulez réutiliser. Un cadre ressemble en fait plus à un composant personnalisé qu'à une fiche. Il est possible d'intégrer un cadre à la palette des composants afin de le réutiliser plus facilement. Les cadres peuvent être imbriqués dans une fiche, dans d'autres cadres ou dans d'autres objets conteneur. Après qu'un cadre ait été créé et enregistré, il continue de fonctionner en tant qu'unité et à hériter des modifications apportées aux composants (y compris les autres cadres) qu'il contient. Lorsqu'un cadre est incorporé dans un autre cadre ou dans une fiche, il continue d'hériter des modifications apportées au cadre dont il dérive.

Pour ouvrir un nouveau cadre, choisissez Fichier | Nouveau | Cadre.



Vous pouvez ajouter au cadre les composants visuels ou non visuels de votre choix. Une nouvelle unité est automatiquement ajoutée dans l'éditeur de code.

Pour plus d'informations...

Voir “cadres” et “TFrame” dans l’index de l’aide en ligne.

Ajout de contrôles ActiveX

Vous pouvez ajouter des contrôles ActiveX à la palette de composants et les utiliser dans vos projets Delphi. Choisissez Composant | Importer un contrôle ActiveX pour ouvrir la boîte de dialogue Importation d’ActiveX. Vous pouvez alors recenser de nouveaux contrôles ActiveX ou en sélectionner un déjà recensé en vue de son installation dans l’EDI. Lorsque vous installez un contrôle ActiveX, Delphi crée et compile un fichier unité “enveloppe” pour ce contrôle.

Pour plus d'informations...

Choisissez Composant | Importer un contrôle ActiveX et cliquez sur le bouton Aide.

Définition des options des projets

Si vous devez gérer les répertoires de votre projet ou spécifier les options de fiche, application, compilateur et lieu de votre projet, choisissez Projet | Options. Lorsque vous effectuez des modifications dans la boîte de dialogue Options du projet, elles n’affectent que le projet en cours ; vous pouvez cependant enregistrer vos sélections comme paramètres de fonctionnement par défaut des nouveaux projets.

Définition des options de projet par défaut

Pour enregistrer vos sélections comme paramètres de fonctionnement par défaut de tous les nouveaux projets, cochez Défaut, dans le coin inférieur gauche de la boîte de dialogue Options du projet. L’activation de la case Défaut écrit les paramètres en cours du dialogue dans le fichier d’options defproj.dof, placé dans le répertoire Delphi7\Bin. Pour restaurer les paramètres par défaut de Delphi, supprimez ou renommez le fichier defproj.dof.

Pour plus d'informations...

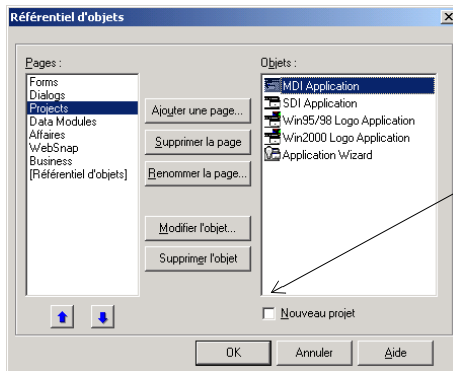
Voir “Options de projet, boîte de dialogue” dans l’index de l’aide en ligne.

Spécification des modèles de projet et de fiches par défaut

Lorsque vous choisissez Fichier | Nouveau | Application, un nouveau projet est ouvert dans l’EDI. Delphi crée une nouvelle application avec une fiche vide, sauf si vous spécifiez un *modèle* de projet comme projet *par défaut*. Vous pouvez enregistrer vos propres projets en tant que modèles dans le référentiel d’objets dans la page Projets en choisissant Projet | Ajouter au référentiel (voir [“Ajout de modèles au référentiel d’objets” à la page 4-9](#)). Vous pouvez aussi utiliser l’un

des modèles de projet du référentiel d'objets Delphi (voir "[Le référentiel d'objets](#)" à la page 2-6).

Pour spécifier votre modèle de projet comme projet par défaut, choisissez Outils | Référentiel. Dans la boîte de dialogue Référentiel d'objets, sélectionnez Projets sous Pages. Si vous avez enregistré un projet en tant que modèle dans la page Projets, il apparaît dans la liste Objets. Sélectionnez le nom du modèle et cochez Nouveau projet, puis choisissez OK.



Les pages du référentiel d'objets contiennent des modèles de projets, des modèles de fiches ou les deux.

Pour définir un modèle de projet comme projet par défaut, sélectionnez un élément dans la liste des objets et cochez Nouveau projet.

Pour définir un modèle de fiche comme fiche par défaut, sélectionnez un élément dans la liste des objets et cochez Nouvelle fiche ou Fiche principale.

Lorsque vous avez spécifié un modèle de projet comme projet par défaut, Delphi l'ouvre automatiquement chaque fois que vous choisissez Fichier | Nouveau | Application.

De la même façon que vous spécifiez un projet par défaut, vous pouvez spécifier une *nouvelle fiche par défaut* et une *fiche principale par défaut* en les choisissant dans la liste des modèles de fiches existant dans le référentiel d'objets. La *nouvelle fiche par défaut* est la fiche créée lorsque vous choisissez Fichier | Nouveau | Fiche pour ajouter une fiche supplémentaire à un projet ouvert. La *fiche principale par défaut* est la fiche créée lorsque vous ouvrez une nouvelle application. Si vous n'avez spécifié aucun projet par défaut, le modèle de projet vierge est le modèle par défaut.

Vous pouvez toujours remplacer le projet ou les fiches par défaut en choisissant Fichier | Nouveau | Autre et en sélectionnant un modèle différent dans la boîte de dialogue Nouveaux éléments.

Pour plus d'informations...

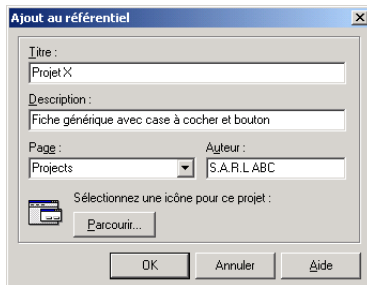
Voir "modèles, ajout au référentiel d'objets", "projets, spécification des valeurs par défaut" et "fiches, spécification des valeurs par défaut" dans l'index de l'aide.

Ajout de modèles au référentiel d'objets

Vous pouvez ajouter vos propres objets au référentiel d'objets afin de les utiliser comme *modèles* que vous réutilisez ou partagez avec d'autres développeurs via un réseau. La réutilisation d'objets vous permet de concevoir des familles

d'applications ayant une interface utilisateur et des fonctionnalités communes, ce qui réduit les temps de développement tout en améliorant la qualité.

Par exemple, pour ajouter un projet comme modèle dans le référentiel, commencez par enregistrer le projet, puis choisissez **Projet | Ajouter au référentiel**. Complétez alors la boîte de dialogue **Ajout au référentiel**.



Saisissez le titre, la description et l'auteur. Dans la liste **Page**, choisissez **Projets** pour que votre projet apparaisse dans la page **Projets** du référentiel d'objets.

À la prochaine ouverture de la boîte de dialogue **Nouveaux éléments**, votre modèle de projet apparaît dans la page **Projets** (ou dans la page spécifiée). Pour que votre modèle soit le modèle par défaut à la prochaine ouverture de Delphi, voir ["Spécification des modèles de projet et de fiches par défaut"](#) à la page 4-8.

Pour plus d'informations...

Voir "modèles, ajouter au référentiel d'objets" dans l'index de l'aide en ligne.

Définition des préférences pour les outils

Vous pouvez contrôler un grand nombre des outils déterminant l'aspect et le comportement de l'EDI, comme le concepteur de fiche, l'inspecteur d'objets ou l'explorateur de code. Ces paramètres affectent non seulement le projet en cours mais aussi les projets que vous allez ouvrir et compiler. Pour modifier les paramètres de l'EDI pour tous les projets, choisissez **Outils | Options d'environnement**.

Pour plus d'informations...

Voir "Options d'environnement, boîte de dialogue" dans l'index de l'aide en ligne ou cliquer sur le bouton **Aide** figurant sur l'une des pages de la boîte de dialogue **Options d'environnement**.

Personnalisation du concepteur de fiche

La page **Concepteur** de la boîte de dialogue **Outils | Options d'environnement** propose des options qui affectent le concepteur de fiche. Par exemple, vous pouvez activer ou désactiver la fonction "Aligner sur la grille", qui aligne les composants sur la ligne de grille la plus proche ; vous pouvez également afficher ou masquer le nom, ou *l'intitulé*, des composants non visuels placés dans les fiches.

Pour plus d'informations...

Dans la boîte de dialogue Options d'environnement, cliquez sur le bouton Aide de la page Concepteur.

Personnalisation de l'éditeur de code

Vous souhaiterez sans doute personnaliser l'éditeur de code immédiatement. Plusieurs pages de la boîte de dialogue Outils | Options de l'éditeur proposent des options d'édition du code. Par exemple, vous pouvez choisir les équivalences des touches, les fontes, la largeur des marges, les couleurs, la mise en évidence syntaxique, les tabulations et les styles d'indentation.

Vous pouvez également configurer les outils d'audit de code, utilisables dans l'éditeur, sur la page Audit de code de la boîte de dialogue Propriétés de l'éditeur. Pour savoir comment utiliser ces outils, voir ["Audit de code" à la page 2-7](#).

Pour plus d'informations...

Dans la boîte de dialogue Propriétés de l'éditeur, cliquez sur le bouton Aide figurant sur les pages Général, Affichage, Affectations de touches, Couleurs et Audit de code.

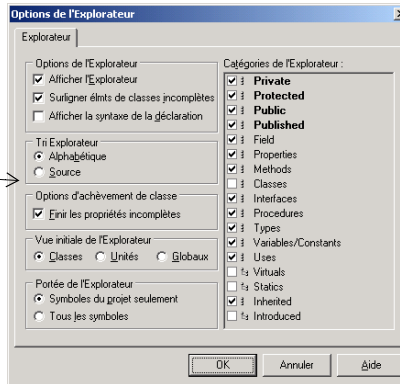
Personnalisation de l'explorateur de code

Au démarrage de l'EDI de Delphi, l'explorateur de code (décrit dans ["L'explorateur de code" à la page 2-11](#)) s'ouvre automatiquement. Si vous ne voulez pas que l'explorateur de code s'ouvre automatiquement, choisissez Outils | Options d'environnement, cliquez sur l'onglet Explorateur et désélectionnez Afficher l'explorateur.

Vous pouvez changer la façon dont le contenu de l'explorateur de code est organisé : cliquez avec le bouton droit de la souris, choisissez Propriétés et, sous Catégories de l'Explorateur, sélectionnez et désélectionnez les cases à cocher. Si une catégorie est cochée, les éléments de cette catégorie sont regroupés sous un même nœud. Si une catégorie n'est pas cochée, chaque élément de cette catégorie est affiché de manière indépendante le long du tronc de l'arborescence.

Si, par exemple, vous ne cochez pas la case Publié, le dossier Publié disparaît mais pas les éléments qu'il contenait.

Dans l'explorateur de code, vous pouvez trier tous les éléments source dans l'ordre alphabétique ou dans l'ordre déclaré dans le fichier source.



Pour afficher le dossier de chaque type d'élément source dans l'explorateur de code, cochez une catégorie Explorateur.

Pour plus d'informations...

Voir “explorateur de code, options d'environnement” dans l'index de l'aide en ligne.

Index

A

- achèvement de classe 2-8
- achèvement du code 2-7
- ActiveX
 - installation des contrôles 4-8
 - page de la palette de composants 3-13
- administrateur BDE 3-11
- ADO 3-11
- affectation des touches 4-11
- agent Web, introduction 3-9
- aide en ligne, fichiers 1-5
- Aide, F1 1-5
- ajout d'éléments au référentiel d'objets 2-6
- ancrage des fenêtres 4-2–4-4
- applications
 - base de données 3-10
 - compilation et débogage 3-6
 - création 3-1, 3-9
 - déploiement 3-8
 - internationalisation 3-8
 - serveur Web 3-9
- applications de bases de données, création 3-10
- applications multiplates-formes 3-9
- applications serveur Web, création 3-9
- arborescence d'objets 2-5
- audit de symbole dans conseil 2-8

B

- barres d'outils 2-3
 - ajout et suppression de composants 4-2
 - organisation 4-1
- BDE 3-11
- bibliothèque de composants 3-5
- bibliothèque de composants Borland multiplate-forme (CLX) 3-5
- bibliothèque de composants visuels (VCL)
 - ajout de composants 2-4
 - utilisation 3-5
- bibliothèque de types, définition 3-13
- boîtes de dialogue, dans le référentiel d'objets 2-6
- bureau
 - enregistrement des dispositions 4-4
 - organisation 4-1–4-5

C

- cadres 4-7
- CLX
 - ajout de composants 2-4
 - applications, création 3-9

- définition 3-5
- code
 - aide à l'écriture 2-7–2-8
 - écriture 3-5
 - gestionnaires d'événements 3-5
 - visualisation et modification 2-7–2-12
- compilation des applications 3-6
- composants
 - agencement dans la palette 4-5
 - ajout à la palette de composants 4-5
 - ajout à une fiche 3-2
 - création de personnalisés 3-12
 - définition des propriétés 3-3
 - installation 3-12, 4-6
 - personnalisation 3-12, 4-6
- concepteur de fiche
 - définition 2-4
 - personnalisation 4-10
- contrôles, ajout à une fiche 3-2
- conventions typographiques 1-7

D

- dbExpress 3-10
- débogage des programmes 3-6–3-7
- débogueur intégré 3-7
- définition des propriétés 3-3
- Delphi
 - démarrage 2-1
 - introduction 1-1
 - personnalisation 4-1–4-11
 - programmation 3-1
- Delphi, enregistrement 1-2
- démarrage de Delphi 2-1
- déploiement d'applications 3-8
- .dfm, fichiers 2-10
- dictionnaire de données 3-12
- DLL
 - création 2-6
 - définition 3-12
 - déploiement 3-8

E

- écriture du code 3-5
- EDI
 - définition 1-1
 - organisation 4-1
 - parcours rapide 2-1
 - personnalisation 4-1–4-11
- éditeur de code
 - combinaison avec d'autres fenêtres 4-2

- personnalisation 4-11
- utilisation 2-7—2-9
- éditeurs de méthodes de saisie 3-8
- éditions de Delphi 3-9
- enregistrement des dispositions du bureau 4-4
- enregistrement du produit 1-2
- environnement de bureau intégré *Voir* EDI
- Évaluation d'expressions de conseil 2-7
- exécutables, déploiement 3-8
- exécution d'une application 3-6
- expert DLL de ressources 3-8
- experts, recherche 2-6
- explorateur 2-13
- explorateur de bases de données 3-11
- explorateur de classes 2-11
- explorateur de code 2-11
 - personnalisation 4-11
 - utilisation 2-11
- explorateur de projet 2-13
- explorateur SQL 3-11

F

- fenêtres à onglets, ancrage 4-4
- fenêtres d'outils, ancrage 4-2
- fenêtres, combinaison 4-2
- fiche principale, définition 4-9
- fiches
 - ajout de composants 3-2
 - principales 4-9
 - recherche 2-6
 - spécification comme défaut 4-9
- fichiers fiche, visualisation du code 2-10

G

- gestionnaire de projet 2-12—2-13
- gestionnaires d'événements, définition 3-5
- groupes de discussion 1-6
- groupes de projets 2-12

I

- IME 3-8
- information, recherche 1-4
- inspecteur d'objets
 - définition 2-4
 - référence de composants en ligne 3-4
 - utilisation 3-3—3-4
- installation des composants personnalisés 4-6
- InterBase 3-11
- interfaces utilisateur, création 3-2
- internationalisation des applications 3-8

J

- jeux de caractères étendus 3-8

K

- Kylix 1-1

L

- listes A faire 2-13
- localisation des applications 3-8

M

- menus
 - contextuels 2-3
 - organisation 2-3, 4-1
- menus clic droit 2-3
- modèles
 - ajout au référentiel d'objets 4-9
 - spécification comme défaut 4-8
- modèles de code 2-8
- modèles de composants création 4-6
- modèles de projet 4-9
- module Base de données 3-12
- modules de données
 - ajout 3-2
 - création 2-6

N

- nouveautés 1-4
- Nouveaux éléments, boîte de dialogue
 - enregistrement des modèles 4-8, 4-10
 - utilisation 2-6
- nouvelle fiche, définition 4-9

O

- objets, définition 3-5
- ODBC 3-11
- Options d'environnement, boîte de dialogue 2-8, 4-10
- Options de l'éditeur, boîte de dialogue 2-8, 4-11
- Options de projet, boîte de dialogue 4-8
- options, définition pour les projets 4-8
- outils de traduction 3-8

P

- page Diagramme 2-9
- palette de composants
 - ajout de composants personnalisés 3-12
 - ajout de pages 4-5
 - définition 2-4
 - personnalisation 4-5—4-7
 - utilisation 3-2
- paquets 4-7
- par défaut
 - modèles de projet et de fiches 4-8
 - options du projet 4-8

- Paradox 3-11
- paramètres du code 2-7
- personnalisation
 - concepteur de fiche 4-10
 - EDI 4-1–4-11
 - éditeur de code 4-11
 - explorateur de code 4-11
 - palette de composants 2-3
- programmation avec Delphi 3-1
- programmes
 - applications CLX 3-9
 - applications serveur Web 3-9
 - compilation et débogage 3-6
 - déploiement 3-8
 - internationalisation 3-8
- projets
 - ajout d'éléments 2-6
 - création 3-1
 - définition des options par défaut 4-8
 - gestion 2-12–2-13
 - spécification comme défaut 4-8
 - types 3-9–3-12
- propriétés, définition 3-3

R

- référentiel d'objets
 - ajout des modèles 4-8, 4-9

- définition 2-6, 3-1
- utilisation 2-6
- relations parent-enfant 2-5

S

- services de support 1-6
- source, aide à l'écriture 2-7–2-8
- support aux développeurs 1-6
- support technique 1-6
- symboles globaux 2-13

T

- tutoriels 1-1

V

- versions de Delphi 3-9

W

- Web, site Borland 1-6
- WebSnap, introduction 3-9

X

- .xfrm, fichiers 2-10

