

Guide de l'utilisateur

Borland

C++ pour Windows[®]

Version 4.0

Les fichiers suivants peuvent être redistribués librement, comme spécifié dans la licence d'utilisation du présent logiciel :

- BC40RTL.DLL
- BIDS40.DLL
- BIDS40D.DLL
- BIDS40DF.DLL
- BIDS40F.DLL
- BIVBX10.DLL
- BW320007.DLL
- BW320009.DLL
- BW32000C.DLL
- BWCC.DLL
- BWCC0007.DLL
- BWCC0009.DLL
- BWCC000C.DLL
- BWCC32.DLL
- COMPRESS.EXE
- CTL3D.DLL
- CTL3D32.DLL
- CW32.DLL
- CW32MT.DLL
- CX32.DLL
- CX32MT.DLL
- DIB.DRV
- LOADBWCC.EXE
- LOCALE..DLL
- MARS.DLL
- MARS.MOB
- MSMOUSE.DRV
- OWL200.DLL
- OWL200D.DLL
- OWL200DF.DLL
- OWL200F.DLL
- PENWIN.DLL
- PENWIN.INI
- REGLOAD.EXE
- STRESS.DLL
- TOOLHELP.DLL
- VGAP.DRV
- YESMOUSE.DRV

Les applications mentionnées dans ce manuel sont brevetées ou en attente de brevet. Ce document ne donne aucun droit sur ces brevets.

Copyright© 1988-1993 Borland international. Tous droits réservés. Tous les produits Borland sont des marques déposées de Borland International, Inc. Tous les autres noms de produits sont des marques déposées de leurs fabricants respectifs.

10 9 8 7 6 5 4 3 2 1

Table des matières

Partie 1 : Utilisation de Borland C++ pour Windows

Chapitre 1 Pour commencer

Installation de Borland C++	5
Configuration matérielle et logicielle requise	5
Procédure d'installation	6
Lancement de Borland C++	8
Comment activer l'aide	10
Configuration de l'EDI	10
Modification des barres d'icônes	11
Choix de préférences de l'EDI	12
Enregistrement des paramètres de l'EDI	12
Utilisation de l'Editeur	13
Configuration de l'éditeur de l'EDI	13
Mise en surbrillance de la syntaxe	14
La fenêtre de messages	15
Parcourir votre code	16
Visualisation des objets (vue générale sur les classes)	17
Filtres	17
Affichage des déclarations des symboles listés	18
Déplacement dans les symboles globaux	18
Utilisation des expressions régulières dans le Browser	19
Visualisation des symboles dans votre code	20
Utilisation des outils en ligne de commande	20
DPMI et les outils en ligne de commande	21
Mémoire vive et MAKESWAP.EXE	21
Gestionnaire de run-times et outils en ligne de commande	22
Contrôle de la mémoire utilisée par le gestionnaire de run-times	22
Lancement d'autres programmes depuis l'EDI ..	23
Chapitre 2 Utilisation du gestionnaire de projet	
Qu'est-ce que la gestion de projets ?	25
Construction d'un projet	27
Création d'un projet multi-cible	29
Conversion des anciens projets	29
Conversion de projets en fichiers MAKE	29

Modification de la visualisation du projet	30
Création d'un projet	31
Construction d'une partie de projet	31
Modification de l'arborescence du projet	32
Modification des attributs de cible avec TargetExpert	32
Modification des attributs de noeud	33
Ajout et suppression de noeuds	34
Ajout et suppression de cibles	34
Déplacement des noeuds et des cibles	35
Copie de noeuds	35
Utilisation des Source Pools	36
Définition des options d'un projet	37
Déroation locale	37
Utilisation des feuilles de style	38
Association d'une feuille de style à un noeud	39
Création d'une feuille de style	39
Modification d'une feuille de style	40
Partage de feuille de styles	40
Visualisation des options d'un projet	41
Traducteurs	42
Installation d'un traducteur	42
Utilisation de Special dans le TurboMenu	44
Installation de visualiseurs et d'outils	44

Chapitre 3 Compilation

Compilation dans l'EDI	46
Utilisation des options de compilation dans l'EDI	46
Utilisation des compilateurs en ligne de commande	47
Fichiers de configuration	48
Fichiers de recherche	49
Règles de priorité des options	49
Options de compilation	50
Répertoires	60
Algorithmes de recherche de fichiers	61
Compiler Defines	62
Compiler Code-generation	63
Compiler Floating Point	64

Compiler Compiler Output	65	Options du menu MDI Child/View	104
Compiler Source	65	MDI Child/View Basic Options	104
Compiler Debugging	66	Chapitre 5 Utilisation de ClassExpert	
Compiler Precompiled headers	68	Lancement de ClassExpert	107
16-bit Compiler Processor	68	ClassExpert : notions de base	107
16-bit Compiler Calling Convention	69	Volet Classes	108
16-bit Compiler Memory Model	70	Volet Events	108
16-bit Compiler Segment Names Data	72	Volet Edit	108
16-bit Compiler Segment Names Far Data	73	Ajouter une classe	109
16-bit Compiler Segment Names Code	74	Créer des types de documents	110
16-bit Compiler Entry/Exit Code	74	Ajout et suppression de gestionnaires	
32-bit Compiler Processor	76	d'événements	112
32-bit Compiler Calling Convention	76	Ajout et suppression de variables	
C++ Options Member Pointer	77	d'instance	113
C++ Options C++ Compatibility	78	Accès au code source des classes	114
C++ Options Virtual Tables	79	Utilisation de Resource Workshop avec	
C++ Options Templates	80	ClassExpert	114
C++ Options Exception handling/RTTI	80	Exécution de Resource Workshop à	
Optimisations	81	partir de l'EDI	115
Optimizations Specific	81	Utilisation de Rescan	116
Optimizations Size	83	Supprimer une classe	116
Optimizations Speed	84	Déplacer une classe	117
Messages	87	Renommer un élément d'AppExpert	117
Messages Portability	87	Importer une classe	117
Messages ANSI Violations	87	Reconstitution du fichier de base de	
Messages Obsolete C++	88	données .APX	118
Messages Potential C++ Errors	88	Chapitre 6 Débogage dans l'EDI	
Messages Inefficient C++ Coding	89	Types de bugs	119
Messages Potential errors	89	Erreurs à la compilation	119
Messages Inefficient Coding	89	Erreurs à l'exécution	120
Messages General	90	Erreurs de logique	120
Make	90	Création d'informations de débogage	121
Options en ligne de commande	90	Indication des arguments du programme	121
Chapitre 4 Création d'applications avec		Contrôle de l'exécution du programme	121
AppExpert		Vérification de la sortie générée par le	
Notions de base sur AppExpert	95	programme	122
Création d'une application avec AppExpert	96	Mode pas à pas	122
Applications AppExpert par défaut	98	Tracé du code en mode approfondi	124
Options du menu Application	98	Mode pas à pas et mode approfondi	
Application Basic Options	99	appliqués aux fonctions membres	
Application Advanced Options	99	d'une classe	124
Application Code Gen Control	100	Mode pas à pas et mode approfondi	
Application Admin Options	101	appliqués au code externe	124
Options du menu Main Window	101	Contourner une partie du code	124
Main Window Basic Options	101	Recherche d'une zone précise	125
Main Window SDI Client	102		
Main Window MDI Client	104		

Recherche d'une fonction	125	Tracé des fenêtres	146
Retour au point d'exécution	125	Choix des messages à tracer	146
Remonter dans le code	125	Utilisation du volet Tracé de messages	147
Arrêt du programme	126	Autres options de tracé	147
Recommencer depuis le début	126		
Visualisation des valeurs	126	Chapitre 8 WinSpector	
Qu'est-ce qu'une expression ?	126	Utilisation de WinSpector	153
Visualisation des expressions	127	Configuration de WINSPCTR.LOG	154
Ajout d'un suivi	127	WINSPCTR.LOG Références	155
Formatage des expressions de suivi	127	Section Disassembly	157
Désactivation d'un suivi	128	Section Stack Trace	157
Suppression d'un suivi	129	Section Registers	158
Modification d'un suivi	129	Message Queue :	159
Evaluation et modification des expressions	129	Section Tasks	159
Evaluation des expressions	129	Section Modules	160
Modification des variables	130	Section des tas USER et GDI	160
Visualisation des éléments de données	131	System Info	161
Examen des valeurs de registres	132	Traitement des données WinSpector	161
Points d'arrêt	133	Sortie de DFA	162
Pose de points d'arrêt	133	Utilisation de DFA avec le fichier	
Utilisation des points d'arrêt	133	WINSPCTR.LOG	162
Suppression de points d'arrêt	134	Utilisation de DFA avec WINSPCTR.BIN	162
Désactiver et activer les points d'arrêt	134	Autres outils de WinSpector	164
Visualisation et modification du code		Utilisation de EXEMAP.EXE	164
sous un point d'arrêt	135	Utilisation de TMAPSYM	164
Réinitialisation de points d'arrêt		Utilisation de BUILDSYM	165
incorrects	135		
Visualisation et modification des		Chapitre 9 Utilisation de l'éditeur de liens	
propriétés d'un point d'arrêt	135	TLINK	
Consignation des expressions	136	Notions de base sur TLINK	167
Personnalisation des points d'arrêt et du		TLINK.CFG	169
point d'exécution	137	Fichiers de recherche	170
Gestion des erreurs GPF	137	Utilisation de TLINK avec BCC.EXE	171
Utilisation de la fenêtre Event Log	138	Edition de liens pour les bibliothèques	171
Débogage des DLL	138	Options de TLINK	173
Débogage en mode soft et en mode hard	139	Fichiers de définition de modules	183
		Instruction CODE	183
Chapitre 7 WinSight		Instruction DATA	184
Démarrage de WinSight	141	Instruction DESCRIPTION	185
Activation et désactivation de la mise à jour	142	Instruction EXETYPE	185
Désactivation du tracé des messages	142	Instruction EXPORTS	185
Choix d'une vue	143	Instruction IMPORTS	186
Liste des classes	143	Instruction LIBRARY	187
Utilisation du volet Liste de classes	143	Instruction NAME	188
Tracé des classes	144	Instruction SEGMENTS	188
Arborescence de fenêtres	144	Instruction STACKSIZE	189
Recherche d'une fenêtre	145		
Sortie du mode de recherche de fenêtres	146		

Instruction STUB	189	Préfixes des commandes	217
Valeurs par défaut des fichiers de définition de modules	190	Utilisation de @	218
Chapitre 10 Les outils de compilation de ressources		Utilisation de - <i>valeur</i> et de -	218
BRCC : le compilateur de ressources	192	Utilisation de &	218
RLINK : l'éditeur de liens pour ressources	193	Opérateurs de commandes	218
BRC : le shell de ressources	195	Débogage avec fichiers temporaires	219
Chapitre 11 Les bibliothèques		Les macros de MAKE	220
IMPLIB : le bibliothécaire d'import	197	Définition des macros	220
IMPDEF : le gestionnaire de fichiers de définition de module	199	Utilisation des macros	221
Les classes d'une DLL	199	Substitution de chaînes dans les macros	221
Les fonctions d'une DLL	200	Macros par défaut de MAKE	222
TLIB : le Turbo bibliothécaire	201	Modification des macros par défaut	222
Pourquoi utiliser des bibliothèque de modules objet ?	201	Les directives de MAKE	223
Ligne de commande de TLIB	202	.autodepend	224
Les fichiers de recherche	203	!error	224
Distinction des majuscules et des minuscules dans les symboles : l'option /C	203	Résumé des contrôles d'erreurs	224
Création d'un dictionnaire étendu : l'option /E	204	!if et autres directives conditionnelles	225
Choix du format de la page : l'option /P	204	!include	226
Suppression des enregistrements de commentaire : l'option /O	205	!message	227
Liste d'opérations	205	.path.ext	227
Exemples	206	.precious	227
Chapitre 12 Utilisation de MAKE		.suffixes	227
Généralités sur MAKE	207	!undef	228
BUILTINS.MAK	209	Utilisation de macros dans les directives	229
Utilisation de TOUCH.EXE	209	Macros vides	229
Options de MAKE	210	Partie 2 : Utilisation de Resource Workshop	
Définir des options par défaut	211	Chapitre 13 Présentation de Resource Workshop	
Compatibilité avec NMAKE de Microsoft	211	Ressources Windows	233
Utilisation des fichiers MAKE	212	Types de fichiers ressources	235
Cibles symboliques	213	Fichiers ressources bitmap	236
Règles pour cibles symboliques	213	Chapitre 14 Projets, ressources et identificateurs	
Règles explicites et implicites	213	Création d'un projet	239
Syntaxe des règles explicite	214	Ouverture d'un projet existant	240
Cibles uniques avec règles multiples	215	Comment utiliser la fenêtre projet	241
Syntaxe des règles implicites	215	Ressources intégrées ou liées	241
Règles explicites utilisées avec des commandes implicites	216	Affichage d'informations dans la fenêtre projet	242
Syntaxe des commandes	217	Show Identifiers	242
		Show Resources	242
		Show Items	242
		Show Unused Types	243

Sélection d'une ressource	243	Sélection d'un style de cadre	266
Comment travailler avec les ressources	243	Spécification du style de dialogue	266
Chargement d'une ressource	243	Sélection de police	268
Éditeurs de ressources	244	Comment inclure un menu	268
L'éditeur de texte interne	244	Comment affecter une classe personnalisée à une boîte de dialogue	269
Ajout d'une ressource	244	Définition de la position de la boîte de dialogue	269
Ajout d'une ressource intégrée	245	Comment travailler avec les contrôles	269
Ajout d'une ressource liée	245	Familles de contrôles	270
Comment déplacer une ressource	247	Palette Tools	270
Comment copier une ressource entre projets	247	Sélection de contrôles	273
Comment supprimer une ressource	248	Ajout de contrôles	274
Comment renommer une ressource	248	Ajout de plusieurs copies d'un contrôle	275
Spécification des options de mémoire ressource	248	Propriétés des contrôles (contrôles .VBX)	275
Comment utiliser les identificateurs	250	Déplacement des contrôles	276
Composants des identificateurs de ressource	250	Redimensionnement des contrôles	276
Fichiers identificateurs	251	Placement et dimensionnement des contrôles en une seule opération	276
Création des fichiers identificateurs	252	Alignement des contrôles avec une grille	277
Fichiers en-tête C	252	Edition de contrôles	277
Gestion automatique des identificateurs	253	Comment ajouter des libellés aux contrôles	279
Comment travailler sans fichier d'identificateurs	254	Modification d'une classe de contrôle	279
Ajout d'identificateurs	254	Comment définir des contrôles comme taquets de tabulation	280
En renommant les ressources	254	Regroupement de contrôles liés	281
A partir de la boîte de dialogue Identifiers	255	Comment réordonner les contrôles (séquence de tabulation)	281
Edition d'identificateurs	256	Agencement, alignement, et redimensionnement des contrôles	282
Suppression d'identificateurs	256	Comment aligner plusieurs contrôles	282
Liste des identificateurs	257	Comment disposer les contrôles en colonnes et rangées	284
Lancement d'un éditeur de ressources	257	Comment redimensionner plusieurs contrôles	285
Préférences de configuration	258	Options de dimensionnement pour un contrôle unique	286
Undo Levels	258	Contrôles boutons	287
Text Editor	258	Valeurs d'identificateur des contrôles bouton-poussoir	288
Multi-Save	258	Contrôles barre de défilement	289
Target Windows Version	259	Contrôles boîte liste	290
Comment travailler avec des fichiers binaires	259	Contrôles boîte de saisie	292
Comment créer des ressources 32 bits	261	Contrôles statiques	294
Chapitre 15 Création de boîtes de dialogue		Contrôles statiques icône	297
Lancement de l'éditeur Dialog	263	Contrôles boîtes à options combinées	297
Création d'une boîte de dialogue	263	Contrôles personnalisés	299
Edition d'une boîte de dialogue existante	264	Création de contrôles personnalisés	299
Utilisation de l'éditeur Dialog	265		
Sélection d'une boîte de dialogue	265		
Définition des attributs d'une boîte de dialogue	265		
Ajout d'un libellé	266		

Installation d'une bibliothèque de contrôles (.DLL ou .VBX)	300	Pour créer une nouvelle table de raccourcis . . .	325
Affichage des contrôles personnalisés	300	Pour éditer une table de raccourcis existante . .	325
Ajout d'un contrôle personnalisé	300	Exécution de l'éditeur Menu en même temps .	325
Comment tester une boîte de dialogue	301	Comment utiliser l'éditeur Accelerator	326
Enregistrement d'une boîte de dialogue	302	Volet Outline	326
Enregistrement du projet	302	Volet Attribute	326
Enregistrement d'une boîte de dialogue dans un fichier	302	Edition d'une table de raccourcis	328
Visualisation de deux boîtes de dialogue	303	Ajout d'un raccourci	328
Personnalisation de l'éditeur Dialog	304	Sélection d'un raccourci	328
Chapitre 16 Création de menus		Comment utiliser le volet Attribute	329
Terminologie	307	Paramétrage de la valeur de commande	329
Lancement de l'éditeur Menu	308	Spécification du raccourci	329
Création d'un nouveau menu	308	Fonction Flash	330
Edition d'un menu existant	309	Recherche des combinaisons de touches redondantes	330
Ecran de l'éditeur Menu	309	Création d'un exemple de table de raccourcis . .	331
Volet Attribute	309	Chapitre 18 Création d'une table de chaînes	
Volet Test Menu	309	Lancement de l'éditeur String	336
Volet Outline	310	Création d'une nouvelle table de chaînes	336
Edition de menus	311	Edition d'une table de chaînes existante	336
Ajout de nouvelles instructions	311	Comment travailler avec des tables de chaînes . .	337
Ajout d'éléments de menu et de séparateurs .	312	Windows et les chaînes	338
Edition des éléments de menu	312	Entrée d'une nouvelle chaîne	339
Utilisation du volet Attribute	312	Edition de chaînes existantes	340
Saisie du texte de l'élément	314	Modification d'une chaîne	340
Saisie des identificateurs d'éléments	315	Edition du script ressource d'une table de chaînes	340
Déplacement et copie d'instructions	315	Modification de chaînes	340
Suppression d'instructions de menu	316	Création d'un exemple de table de chaînes	341
Création de menus flottants	316	Chapitre 19 Comment utiliser l'éditeur Bitmap	
Test des menus	317	Lancement de l'éditeur Bitmap	345
Edition de scripts ressource menu	318	Couleurs de premier plan et de fond, et pixels . .	346
Exemple de menu	319	Utilisation de la palette Tools (Outils)	346
Création du menu exemple avec l'éditeur Menu	320	Outil Rectangle de sélection	348
Ajout de commandes au menu Widgets	320	Outil Ciseaux	348
Ajout de commandes au menu Arrange List	321	Outil Zoom	348
Test du menu	322	Outil Gomme	349
Chapitre 17 Création de raccourcis		Outil Stylo	349
Combinaisons de touches de la table des raccourcis	324	Outil Pinceau	350
Touches ASCII	324	Outil Aérographe	350
Touches virtuelles	324	Outil Boîte de peinture	350
Lancement de l'éditeur Accelerator	325	Outil Ligne	351
		Outil Texte	352
		Dessin de cadres vides	352

Dessin de cadres pleins	353	Ajout d'un effet à trois dimensions	376
Outil Main	353	Dessin de la page du livre de comptes	377
Sélections de style	354	Chapitre 21 Création de curseurs	
Utilisation des deux volets de la fenêtre	354	Création d'un nouveau curseur	380
Lecture de la ligne d'état	355	Ajout d'un curseur dans un fichier projet	380
Comment travailler avec les couleurs	356	Création d'un fichier curseur autonome	380
Choix du nombre de couleurs d'une		Option de format Binary	381
ressource	356	Fichier projet curseur	381
Utilisation des couleurs de premier plan et		Edition des curseurs	381
de fond.	357	Palette Colors pour les curseurs	381
Zones couleur transparentes et inversées	357	Comment travailler avec les zones	
Définition des couleurs transparentes et		transparentes et inversées	382
inversées	358	Réglage du point actif du curseur	382
Comment afficher et cacher la palette Colors	359	Chapitre 22 Création de polices	
Personnalisation des couleurs	359	Types de polices	386
Edition de couleurs dans la palette Colors	360	Pourquoi placer des bitmaps dans une police	386
Index de palette	360	Création d'une nouvelle ressource police	387
Edition d'une couleur	360	Ajout d'une police dans un fichier projet	387
Ajout de texte à une ressource	361	Création d'un fichier police autonome	388
Alignement du texte	362	Option de format Binary	388
Choix des polices, de la taille et du style de		Fichier projet police	388
texte	362	Edition d'une ressource police	389
Choix des formes de pinceaux	363	Définition et ajout de caractères pour une	
Choix des motifs de dessin	363	police	389
Choix d'un style de ligne	364	Définition de la taille de police	389
Alignement d'une zone sélectionnée	365	Définition du nombre de caractères	390
Redimensionnement d'une zone sélectionnée	365	Projection du jeu de caractères	390
Choix des options globales de l'éditeur Bitmap	366	Création de polices proportionnelles	391
Draw on both images	366	Définition de la largeur d'un caractère ou	
Grid on zoomed windows	367	d'une image	391
Save with default device colors	367	Définition d'un en-tête pour une ressource	
Chapitre 20 Création d'icônes		police	392
Création d'une nouvelle icône	369	Modification de la taille et des attributs	394
Ajout d'une icône dans un fichier projet	369	Utilisation de vos polices dans vos applications	394
Création d'un fichier icône autonome	370	Chapitre 23 Création des ressources définies par	
Option de format Binary	371	l'utilisateur	
Fichier projet icône	371	Création d'un type de ressource	396
Edition des icônes	371	Ajout d'une ressource définie par l'utilisateur	397
Visualisation d'autres résolutions	371	Edition d'une ressource définie par l'utilisateur	397
Utilisation des zones couleur transparentes et		Intégration des données ressources dans le	
inversées	372	fichier projet	398
Ajout d'une image à une ressource icône	372	Comment entrer des données dans le script	
Modification des attributs d'une icône	374	ressource	399
Affichage des informations périphérique	374		
Création d'une icône exemple	375		
Dessin de la calculatrice	375		

Gestion des données stockées dans un fichier distinct	400	Modifier des applications existantes avec BWCC	476
Utilisation du type de ressource RCDATA	401	Chargement de BWCC pour activer les contrôles Borland	476
Annexe A Messages d'erreur		Utilisation de BWCC dans des programmes C et C++	477
Types de messages	403	Astuces sur les modifications de ressources	477
Erreurs fatales	403	Accélérateurs	477
Erreurs	404	Bitmaps, curseurs et icônes	477
Mises en garde	404	Boîtes de dialogue	477
Messages du compilateur d'aide	405	Menus	478
Listes de messages	405	Tables de chaînes	478
Description des messages	406	Annexe C En-têtes précompilés	
Annexe B Contrôles personnalisés Windows de Borland		Principe	479
Utilisation de la classe Borland de dialogues personnalisés	469	Inconvénients	480
Utilisation des contrôles Borland	470	Comment utiliser des en-têtes précompilés	480
Perfectionnements des boutons et cases à cocher	472	Création des noms de fichiers	481
Utilisation des boîtes de dialogue de style BWCC	472	Contrôle d'identité	481
Boîte de dialogue Borland Button Style	473	Optimisation des en-têtes précompilés	482
Boîte de dialogue Borland Radio Button Style	475	Annexe D Utilisation d'EasyWin	
Boîte de dialogue Borland Check Box Style	475	Convertir des applications DOS en applications Windows : un jeu d'enfant !	485
Boîte de dialogue Borland Shade Style	475	_InitEasyWin()	486
Boîte de dialogue Borland Static Text Style	476	Fonctions supplémentaires	487
		Index	489

Tableaux

1.1	Symboles littéraux du Browser	19	12.5	Macros par défaut	222
1.2	Expressions de recherche du Browser	19	12.6	Autres macros par défaut	222
1.3	Variables d'environnement contrôlant l'allocation mémoire de RTM.	22	12.7	Modificateurs de noms de fichiers	223
3.1	Récapitulatif des options.	50	12.8	Les directives de MAKE.	223
4.1	Client/view class avec Document/view	103	12.9	Opérateurs conditionnels.	226
4.2	MDI client/view class avec Document/view	104	14.1	Options mémoire des ressources	249
6.1	Spécificateurs de format des expressions du débogueur	130	14.2	Préfixes des identificateurs	254
6.2	Drapeaux CPU de la fenêtre Registers	132	15.1	Options communes aux boîtes de dialogue Style.	278
6.3	TurboMenu de la fenêtre Event Log.	138	15.2	Attributs de contrôle.	279
7.1	Messages souris	149	15.3	Options d'alignement	283
7.2	Messages fenêtres.	149	15.4	Options Size	286
7.3	Messages clavier.	149	15.5	Types de bouton	287
7.4	Messages système	149	15.6	Contrôles boutons Windows prédéfinis	289
7.5	Messages d'initialisation.	149	15.7	Options d'alignement pour barre de défilement.	290
7.6	Messages du presse-papiers.	150	15.8	Options Owner Drawing	290
7.7	Messages DDE	150	15.9	Options List Box	291
7.8	Messages non clients	150	15.10	Options de la boîte de dialogue Edit Text Style	293
7.9	Messages d'impression	150	15.11	Styles Windows 3.1	294
7.10	Messages de contrôle.	151	15.12	Options Control Type.	295
7.11	Messages stylo	152	15.13	Types de boîtes à options combinées.	298
7.12	Messages multimédia	152	15.14	Options Owner Drawing	298
7.13	Autres messages.	152	15.15	Attributs de la boîte à options combinées.	299
7.14	Messages non documentés par Microsoft	152	15.16	Unités ligne d'état	304
8.1	Types d'exceptions.	156	15.17	Options Selection Border	304
8.2	Options de DFA	163	15.18	Options Drawing Type.	304
9.3	Options TLINK.	173	15.19	Options Selection.	305
9.4	Mises en garde TLINK32 s	181	16.1	Sélections du menu View.	310
10.1	BRCC (le compilateur de ressources de Borland).	192	16.2	Sélections du volet Attribute de l'éditeur Menu	313
10.2	Les options de RLINK.	193	17.1	Sélections dans le volet Attribute de l'éditeur Accelerator	327
10.3	Les options de BRC	195	19.1	Commendes de zoom.	349
11.1	Options IMPLIB	198	22.1	Options de taille de police	390
11.2	Options de TLIB	202	22.2	Options Character	391
11.3	Symboles d'action de TLIB.	206	22.3	Options d'en-tête.	393
12.1	Options MAKE.	210	B.1	Boutons de contrôles BWCC prédéfinis	473
12.2	Préfixes des commandes.	217	B.2	Décalages Bitmap	474
12.3	Opérateurs MAKE.	218			
12.4	Différences entre les deux types de macros	221			

Figures

1.1	Modules de l'EDI	9	15.7	Contrôle boîte de saisie dans la boîte de dialogue File Open.	292
1.2	Affichage des classes d'une application	17	15.8	Boîte à options combinées dans la boîte de dialogue Open File.	297
1.3	Fenêtre de déclaration de symbole	18	16.1	Menu exemple	319
1.4	Affichage des symboles globaux	18	18.1	Editeur String avec des entrées de table de chaînes	337
2.1	Arborescence de projet	26	18.2	Editeur String avec quatre chaînes définies	342
5.1	Fenêtre ClassExpert	108	19.1	Palette Tools de l'éditeur Bitmap	347
6.1	Fenêtre de suivi	127	19.2	Palette Colors à 16 couleurs	356
6.2	Boîte de dialogue Watch Properties	128	19.3	Index de la palette de 16 couleurs	360
6.3	Fenêtre des points d'arrêt	133	19.4	Alignement de texte	362
6.4	Boîte de dialogue Breakpoint Properties	134	20.1	Fenêtre icône	373
14.1	Fenêtre projet montrant les ressources par fichier	242	20.2	Calculatrice avant l'ajout de l'ombre portée	376
15.1	Exemple de boîte de dialogue	263	20.3	Calculatrice avec effet d'ombre	377
15.2	Editeur Dialog avec boîte de dialogue vide	264	20.4	Icône Home Budget terminée	378
15.3	Palette Tools	271	B.1	Boîte de dialogue avec contrôles Borland	470
15.4	Coordonnées d'une boîte de dialogue	277			
15.5	Options d'ordre des contrôles	285			
15.6	Boîte de dialogue Size Controls	285			

Utilisation de Borland C++ pour Windows

Cette section du *Guide de l'utilisateur* indique comment installer et utiliser Borland C++. Elle vous apprend à vous servir de ses divers composants dans l'environnement de développement intégré (EDI) : le débogueur intégré, Browser, AppExpert, ClassExpert et le gestionnaire de projet. Elle fournit en outre une documentation sur les outils en ligne de commande : le compilateur, l'éditeur de liens, le bibliothécaire et l'outil MAKE.

Borland C++ possède les caractéristiques suivantes :

- Développement intégré d'applications DOS, Windows, WIN32s et Windows NT. Vous pouvez concevoir plusieurs applications à partir d'un seul fichier projet.
- Création rapide et facile d'applications ObjectWindows grâce à AppExpert. Lorsque vous avez créé votre application, ClassExpert vous aide à effectuer la maintenance en assurant le suivi des classes et des événements, et se combine avec Resource Workshop pour gérer les ressources que vous y utilisez.
- Aide au débogage des applications sans passer par un débogueur séparé.
- Editeur personnalisable. Vous pouvez vous servir des raccourcis clavier fournis avec Borland C++ ou créer les vôtres.

Cette section aborde également deux programmes Windows d'aide au débogage d'applications : WinSight et WinSpector.

Deux fichiers en ligne sont disponibles à propos de Borland C++ ; ils contiennent des informations complémentaires qui ne figurent ni dans les manuels, ni dans l'aide en ligne.

- **INSTALL.TXT** Présente des informations complètes sur l'installation à partir de disquettes ou de CD-ROM.

- **UTILS.TXT** Décrit les outils et utilitaires en ligne de commande non présentés dans les manuels et l'aide en ligne.

Pour commencer

Borland C++ est un ensemble professionnel de développement contenant des outils Windows, des outils en ligne de commande et des bibliothèques. Il vous permettra de créer des applications pour DOS, Windows, Win32s et Windows NT. Ce chapitre vous explique comment utiliser Borland C++ : l'EDI, le gestionnaire de projets, AppExpert, les outils et les divers utilitaires.

Dans ce chapitre, vous apprendrez à

- Installer et configurer Borland C++
- Utiliser l'éditeur
- Utiliser la mise en surbrillance de la syntaxe dans votre code
- Utiliser la barre d'icônes
- Utiliser la fenêtre de messages
- Vous déplacer dans votre code
- Utiliser les outils de l'EDI

Les autres utilitaires et les outils en ligne de commande ne sont décrits ici que succinctement. Pour de plus amples informations sur les autres modules de Borland C++, consultez le *Roadmap*, qui vous indiquera où trouver vos informations dans la documentation.

Installation de Borland C++

Borland C++ contient des applications DOS et Windows. Avant de l'installer, vérifiez que votre configuration répond aux besoins indiqués ci-dessous.

Configuration matérielle et logicielle requise

Pour utiliser Borland C++, il vous faut :

- DOS version 4.01 ou ultérieure
- Windows 3.1 ou toute version ultérieure exécutée en mode 386 étendu

- Un disque dur avec 40 Mo de libres (80 Mo pour l'installation complète)
- Un lecteur de disquette de 1,44 Mo ou un lecteur de CD-ROM (pour l'installation)
- 4 Mo de mémoire étendue
- Une souris compatible Windows

Les éléments ci-dessous, bien que non indispensables, vous permettront de tirer le meilleur parti de votre nouveau logiciel :

- 8 Mo de mémoire vive
- Un coprocesseur 80x87 (si vous utilisez les bibliothèques de calcul en virgule flottante). Notez cependant que Borland C++ émule ce composant s'il ne détecte pas sa présence.

Procédure d'installation

Le programme d'installation installe l'environnement Borland C++ (l'EDI, les outils en ligne de commande, ObjectWindows et Turbo Debugger) ainsi que Win32s, qui vous permet d'exécuter des applications 32 bits sous Windows 16 bits. Ce programme d'installation fonctionne sous Windows, Win32s et Windows NT ; toutes les applications, cependant, ne tournent pas sous Windows NT.

Pour de plus amples informations sur l'installation de Borland C++, consultez le fichier INSTALL.TXT, sur la première disquette (ce fichier n'étant pas compressé, il est lisible par tout éditeur de texte).

L'installation depuis disquettes ou depuis CD-ROM est sensiblement identique, mais nous vous recommandons de lire le fichier INSTALL.TXT ou les notes du CD si vous installez depuis un CD-ROM.

Pour installer Borland C++ depuis les disquettes :

1. Placez la première disquette dans le lecteur.
2. Depuis Windows, choisissez Fichier | Exécuter dans le Gestionnaire de programmes.

Si vous utilisez un utilitaire de compression de fichiers, lisez attentivement le fichier INSTALL.TXT avant de procéder à l'installation de Borland C++.

3. Tapez `a:\install` (ou `b:\install`) et appuyez sur *Entrée*. Une boîte de dialogue s'ouvre, dans laquelle s'affiche la quantité d'espace disque nécessaire pour l'installation complète (Target Requirements), ainsi que l'espace disponible sur votre machine. Si vous utilisez la compression de fichiers, lisez le fichier `INSTALL.TXT` : il se peut que vous ayez besoin de plus de place que ce qui est signalé comme disponible.
4. Cliquez sur *Customize BC4.0 Installation* si vous désirez n'installer qu'une partie du package. Une seconde boîte de dialogue s'ouvre alors, contenant une description des modules installables. Cliquez le bouton correspondant à la partie que vous souhaitez personnaliser. Dans la boîte de dialogue qui s'affiche, vous pouvez cocher les fichiers qui vous intéressent (par défaut, tous les fichiers sont copiés sur le disque de destination). Cliquez sur *OK* et répétez cette opération pour tous les modules que vous souhaitez personnaliser. Cliquez sur *OK* pour revenir à la première boîte de dialogue de l'installation.
5. Les répertoires dans lesquels les fichiers vont être copiés s'affichent à l'écran. Vous pouvez les modifier.
 - **Borland C++ destination directory** est le répertoire principal, à partir duquel tous les fichiers seront copiés (il s'agit, par défaut, de `C:\BC4`).
 - **Borland C++ Working directory** est le répertoire dans lequel les applications de Borland C++ sont installées (`C:\BC4\BIN`).
6. Par défaut, l'installation crée un groupe Windows dans lequel sont placées les icônes de Borland C++. Si vous ne souhaitez pas créer ce groupe, désélectionnez l'option *Create Borland C++ Group*.
7. `Win32s` est également installé par défaut. Désélectionnez cette option si vous ne souhaitez pas copier les fichiers correspondants sur votre disque dur.
8. Ne cochez *LAN Windows Configuration* que si votre machine est équipée de LAN Windows.
9. Cliquez sur *Install* pour lancer la copie. A la fin de l'installation, lisez le fichier `README`, qui contient des informations de dernière minute sur votre logiciel. Pour de plus amples informations sur les icônes créées lors de l'installation, lisez le fichier `INSTALL.TXT` copié dans le répertoire `C:\BC4`.

Vérifiez que votre `CONFIG.SYS` contient au moins la valeur 40 pour les paramètres `FILES` et `BUFFERS`.

La procédure d'installation modifie certains fichiers de votre disque dur :

- AUTOEXEC.BAT contient maintenant le paramètre C:\BC4\BIN (ou toute autre répertoire spécifié par vous) dans l'instruction PATH.
- WIN.INI contient une section [BCW4.0 INSTALL] utilisée par TASM (si celui-ci est installé) pour savoir où Borland C++ est placé sur votre machine. De plus, dans la section [EXTENSIONS], l'extension EDI est associée à l'Environnement de Développement Intégré BCW.EXE.
- SYSTEM.INI contient les lignes suivantes :

```
device = c:\bc4\bin\tddebug.386  
device = c:\bc4\bin\windpml.386
```

- Si vous avez Windows NT, la section [BORLAND INSTALL] est insérée dans CONFIG.NT.

Le fichier FILELIST.DOC liste tous les fichiers composant Borland C++. Lisez-le attentivement avant de supprimer certains fichiers pour libérer de l'espace disque.

Lancement de Borland C++

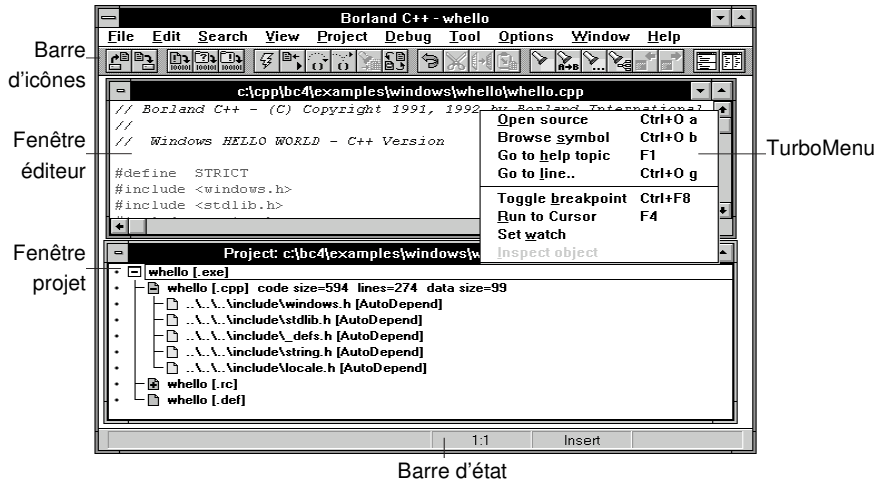


Pour lancer l'EDI, faites un double-clic sur l'icône de Borland C++ (voir marge gauche) sous Windows. L'EDI vous permet de saisir, de modifier, de compiler, de linker (éditer les liens), de déboguer et de gérer vos projets de développement. Il offre :

- Un éditeur et un Browser, décrits dans le présent chapitre
- Un gestionnaire de projets, décrit au chapitre 2
- Un débogueur, décrit au chapitre 6

La figure 1.1 montre certains des modules de l'EDI.

Figure 1.1
Modules de l'EDI



L'EDI intègre des TurboMenus contextuels qui vous permettent de modifier rapidement les objets. Pour voir un TurboMenu, cliquez, avec le bouton droit, dans une fenêtre (celle-ci doit avoir été sélectionnée avec le bouton gauche) ou sur un élément d'une fenêtre, ou appuyez sur *Alt+F10* (le TurboMenu change en fonction de l'objet sélectionné). Par exemple, pour passer rapidement à une ligne précise dans la fenêtre d'édition, cliquez avec le bouton droit dans la fenêtre d'édition, choisissez *Go to line*, puis entrez le numéro de la ligne à atteindre. L'option de menu "Go to line" n'apparaît que si une fenêtre d'édition est sélectionnée. Si vous ouvrez un TurboMenu dans une fenêtre de projet, les menus qui s'afficheront seront totalement différents.

La barre d'icônes change aussi en fonction de la fenêtre sélectionnée. Il existe une barre d'icônes configurable pour l'éditeur, le Browser, le débogueur, le gestionnaire de projets, la fenêtre de messages, le bureau et le ClassExpert (voir la section "Modifications des barres d'icônes", plus loin dans ce chapitre). Lorsque le pointeur est sur un bouton de la barre d'icônes, une ligne décrivant ce bouton s'affiche dans la ligne d'état de l'EDI.

Certains des boutons de la barre d'icônes sont parfois estompés. Ceci signifie que la commande correspondante n'est pas disponible dans le contexte courant. Par exemple, si une fenêtre d'édition est ouverte, le bouton *Coller le texte du Presse-papiers* est estompé si le Presse-papiers ne contient aucun texte.

Comment activer l'aide

L'Aide en ligne de Borland C++ contient des informations détaillées sur Borland C++. Par rapport au manuel, l'Aide en ligne traite des sujets supplémentaires suivants :

- Les menus de l'EDI
- Le KEYMAPPER de l'éditeur
- Exemple de code des bibliothèques run-time
- L'API Windows

Pour activer l'Aide en ligne :

- Depuis l'EDI, choisissez Help dans la barre de menus ou appuyez sur *F1*.
- Dans les boîtes de dialogue, cliquez sur le bouton Help ou appuyez sur *F1*.
- Pour obtenir des informations sur une option de menu, sélectionnez-la et appuyez sur *F1*.

Configuration de l'EDI

Vous pouvez paramétrer l'EDI de telle sorte qu'il effectue seul certaines tâches (créer une sauvegarde des fichiers dans les fenêtres d'édition, par exemple) ou qu'il gère des événements. Cette section explique comment procéder.

La boîte de dialogue Environment Options permet de configurer l'éditeur, le Browser, le débogueur, les fenêtres de projets et d'autres éléments de l'EDI (ces options sont enregistrées dans le fichier BCCONFIG.BCW).

Pour afficher la boîte de dialogue Environment Options, choisissez Options | Environment. La liste des sujets s'affiche à gauche de la boîte de dialogue. Certains comprennent des sous-sujets (le sujet Editeur, par exemple, comporte les sous-sujets Options, File et Display). Lorsqu'un sujet comporte des sous-sujets qui ne sont pas affichés, le signe "+" est placé en regard de son nom. Si vous cliquez dessus, les sous-sujets s'affichent et le signe "+" devient un "-" (si vous cliquez sur le "-", la liste des sous-sujets se rétracte à nouveau). Les sujets n'ayant pas de sous-sujets sont précédés d'un point. Lorsque vous cliquez sur un sujet, ses caractéristiques s'affichent à droite de la boîte de dialogue.

Ce chapitre ne détaille pas toutes les sujets de la boîte de dialogue Options | Environment. Reportez-vous à l'Aide en ligne pour de plus amples informations sur les sujets et les options.

Pour étendre ou condenser la liste des options, vous pouvez aussi appuyer sur + ou sur - (sur le pavé numérique).

Certains sujets associés à des tâches ou des modules de l'EDI sont décrits plus loin dans ce manuel (les options de projet, par exemple, sont décrites au chapitre 2). Consultez l'index en fin d'ouvrage pour localiser les informations spécifiques aux sujets.

**Modification des
barres d'icônes**

Il existe une barre d'icônes configurable pour l'éditeur, le Browser, le débogueur, le gestionnaire de projets, la fenêtre de messages, le bureau et le ClassExpert. Lorsque vous sélectionnez l'une de ces fenêtres, la barre d'icônes correspondante s'affiche. Toutes ces barre d'icôness sont configurables : vous pouvez ne leur attribuer que les boutons dont vous vous servez le plus souvent.

Pour ajouter ou supprimer des boutons dans une barre d'icônes :

1. Choisissez Options | Environment dans le menu principal.
2. Choisissez le sujet barre d'icônes à gauche. La partie droite de la boîte de dialogue liste les options générales de toutes les barre d'icôness.
Ces options vous permettent de modifier la position de la barre d'icônes (en haut ou en bas de l'EDI) et son mode d'utilisation (cochez Use flyby help pour que les lignes d'aide s'affichent dans la ligne d'état lorsque vous placez le pointeur sur un bouton).
3. Choisissez Customize, le sujet sous barre d'icônes. Les options à droite contiennent des informations sur les barre d'icôness.
4. Choisissez le type de barre d'icônes à modifier (Editeur, Browser, Debugger, Project, Message, Desktop ou classex) dans la liste déroulante Window.

La colonne de gauche (Available Buttons) vous présente les boutons disponibles (non utilisés) et leur description. La colonne de droite (Active Buttons) ne présente que les boutons de la barre d'icônes sélectionnée.

5. Pour ajouter un bouton, faites un double-clic sur son icône dans la liste Available Buttons ou sélectionnez-le et cliquez sur la flèche orientée vers la droite. Le bouton passe dans la liste Active Buttons.
6. Pour supprimer un bouton, faites un double-clic sur son icône dans la liste Active Buttons ou sélectionnez-le et cliquez sur la flèche orientée vers la gauche. Le bouton passe dans la liste Available Buttons.

Pour changer l'ordre des boutons d'une barre d'icônes, utilisez les flèches haute et basse. Le bouton sélectionné dans la liste Active Buttons se déplace vers le haut ou vers le bas (le bouton du haut est placé à gauche de la barre d'icônes, et le dernier est placé à droite).

Pour donner le même aspect à toutes vos barre d'icôness, sélectionnez une barre d'icônes dans la liste Window puis cliquez sur Copy Layout. Une boîte de dialogue s'ouvre, dans laquelle vous cochez toutes les barre d'icôness auxquelles vous désirez donner l'aspect de la barre d'icônes sélectionnée.

Pour redonner son aspect par défaut à une barre d'icônes, sélectionnez-la dans la liste Window et cliquez sur Restore Layout.

Les séparateurs placent un espace entre deux boutons. Pour placer des séparateurs dans une barre d'icônes, sélectionnez un bouton dans la liste Active Buttons et cliquez sur Separator. Le séparateur s'insère devant le bouton sélectionné.

Choix de préférences de l'EDI

Les préférences vous permettent de définir précisément ce que vous voulez enregistrer de façon automatique, et de spécifier le mode de fonctionnement de certaines fenêtres.

Pour définir vos préférences,

1. Choisissez Options | Environment | Preferences.
2. Cochez les options de votre choix. Chacune est décrite dans l'Aide en ligne.
3. Cliquez sur OK.

Enregistrement des paramètres de l'EDI

Lorsque vous quittez l'EDI, générez un projet, utilisez un outil de transfert, lancez le débogueur intégré, ouvrez, fermez ou enregistrez un projet, certaines informations sont automatiquement enregistrées sur disque. Il est possible de paramétrer cette fonctionnalité en choisissant Preferences dans la boîte de dialogue Environment Options (qui s'affiche lorsque vous choisissez Options | Environment dans le menu principal).

Pour enregistrer manuellement votre environnement de travail :

1. Choisissez Options | Save.
2. Cochez Environment pour enregistrer les paramètres de l'Editeur, de la Mise en surbrillance de la syntaxe, des barre d'icôness, du Browser et de vos Préférences définies dans la boîte de dialogue Environment Options. L'enregistrement se fait dans le fichier BCCONFIG.BCW.

3. Cochez Desktop pour enregistrer les informations concernant les fenêtres ouvertes et leur position. Ces données sont stockées dans le fichier <projet>.DSW. Si aucun projet n'est ouvert, elles sont stockées dans BCWDEF.DSW.
4. Cochez Project pour enregistrer les modifications apportées au fichier du projet (.IDE), notamment les options de "build" et les attributs de noeuds.

Utilisation de l'Editeur

Vous pouvez ouvrir 32 fenêtres d'édition simultanément dans l'EDI.

Les fenêtres d'édition servent à créer et à éditer le code source de vos programmes. Lorsque vous éditez un fichier, la barre d'état de l'EDI affiche le numéro de la ligne et de la colonne sur lesquelles se trouve le curseur (1 :1 ou 68 :23, par exemple). La barre d'état indique aussi si le curseur est en mode Insertion ou en mode Surfrappe (appuyez sur *Inser* pour passer d'un mode à l'autre). De plus, si vous avez modifié le fichier affiché, le mot Modified vient se placer dans la barre d'état.

L'Editeur vous permet d'annuler les dernières modifications en choisissant Edit | Undo ou en appuyant sur *Alt+Retour arrière*. Si, par exemple, vous effacez une ligne puis en collez une autre, vous pouvez annuler ces deux opérations : l'action de coller la ligne est annulée en premier, puis l'effacement de la ligne initiale. Pour spécifier le nombre d'opérations pouvant être annulées, choisissez Options | Environment | Editor | Options.

Configuration de l'éditeur de l'EDI

Les fichiers .CKB configurent aussi les autres fenêtres de l'EDI. Voir l'Aide en ligne pour de plus amples informations.

Vous pouvez configurer l'éditeur de telle façon qu'il émule d'autres éditeurs de texte (Brief ou Epsilon, par exemple). L'éditeur de l'EDI utilise des fichiers de mappe du clavier (.CKB) qui définissent les raccourcis clavier de l'éditeur (ces fichiers modifient également les raccourcis clavier des autres fenêtres).

Pour utiliser l'un des quatre fichiers .CKB, choisissez Options | Environment | Editor et cliquez sur un SpeedSetting (Default keymapping, EDI classic, BRIEF emulation ou Epsilon). Pour de plus amples informations sur la création de vos propres fichiers .CKB, voir l'Aide en ligne (mot clé "Keymapper").

La mise en surbrillance de la syntaxe vous permet de définir une couleur et un attribut de caractère (gras, par exemple) pour certaines parties de votre code. Ceci permet, par exemple, d'afficher les commentaires en bleu et les chaînes en rouge. Par défaut, la mise en surbrillance est activée.

Pour la désactiver :

1. Choisissez Options | Environment | Syntax Highlighting.
2. Désélectionnez Use Syntax Highlighting.

La mise en surbrillance fonctionne pour les fichiers dont l'extension figure dans la liste Syntax Extensions (.CPP, .C, .H et .HPP par défaut). Vous pouvez supprimer ou ajouter des extensions dans cette liste, mais vous devez les séparer par un point-virgule.

La section Syntax Highlighting affiche les couleurs par défaut ainsi que quatre jeux de couleurs prédéfinis (des boutons).

Pour utiliser des couleurs prédéfinies :

1. Choisissez Options | Environment | Syntax Highlighting.
2. Choisissez l'un des quatre jeux de couleurs en cliquant sur le bouton correspondant. A partir de ces jeux prédéfinis, vous pouvez créer vos propres associations de couleurs pour la mise en surbrillance de la syntaxe.

Pour sélectionner manuellement les couleurs de mise en surbrillance :

1. Choisissez Options | Environment | Syntax Highlighting | Customize. Des éléments et un listing type s'affichent en haut à droite de la boîte de dialogue Environment Options.
2. Sélectionnez l'élément à modifier dans la liste ou cliquez sur l'élément en question dans l'exemple de code (que vous pouvez faire défiler). L'exemple utilise la police sélectionnée dans la section Editor | Display section de la boîte de dialogue Environment Options.
3. Sélectionnez une couleur pour cet élément. Utilisez le bouton gauche de la souris pour sélectionner la couleur du premier plan (FG - "Foreground" - s'inscrit dans la couleur). Utilisez le bouton droit pour sélectionner la couleur d'arrière-plan (BG - "Background" - s'inscrit). Si FB s'inscrit dans la couleur, cela signifie que la couleur est utilisée à la fois dans le premier plan et dans l'arrière-plan.
4. Choisissez un attribut, gras par exemple.

5. Vous pouvez cocher Default FG (premier plan) ou BG (arrière-plan) pour utiliser les couleurs par défaut de Windows pour un élément.
6. Répétez les étapes 2, 3 et 4 pour tous les éléments à modifier.

La fenêtre de messages

Vous pouvez personnaliser certaines des fonctionnalités des fenêtres de messages en utilisant la section Preferences de la boîte de dialogue Environment Options.

La fenêtre de messages affiche les erreurs et des avertissements lorsque vous compilez vos programmes. Lorsque vous sélectionnez un message dans la fenêtre de messages, l'éditeur place le curseur à l'endroit de votre code qui a généré ce message. Si le fichier contenant l'erreur n'est pas en mémoire, appuyez sur la *Barre d'espacement* pour le charger (vous pouvez aussi appuyer sur *Alt+F10* et choisir View source dans le TurboMenu). La fenêtre de messages reste active pour que vous puissiez passer de message en message.

Pour voir le code associé à une erreur ou à un avertissement, sélectionnez le message dans la fenêtre de messages et appuyez sur *Entrée*, double-cliquez sur le message, ou appuyez sur *Alt+F10* et choisissez Edit source dans le TurboMenu. Le curseur se place sur le source qui a généré l'erreur (la fenêtre de messages passe à l'arrière-plan). Utilisez *Alt+F7* pour voir le message d'erreur suivant, *Alt+F8* pour voir le précédent.

Vous pouvez aussi vous déplacer dans les messages dans la fenêtre de messages. Lorsque vous sélectionnez un message, le curseur de la fenêtre d'édition se place automatiquement sur le code fautif (ceci s'appelle la recherche automatique d'erreur). La recherche automatique d'erreur ne fonctionne que si le fichier en question est ouvert dans une fenêtre d'édition. Si le message suivant concerne un fichier source autre que celui qui est dans la fenêtre d'édition active, vous devez sélectionner la fenêtre d'édition en question pour poursuivre la recherche automatique d'erreur.

Vous pouvez effacer la fenêtre de messages en choisissant Remove all messages dans le TurboMenu (cliquez avec le bouton droit ou appuyez sur *Alt+F10* pour afficher le TurboMenu).

Parcourir votre code

Le Browser permet de visualiser la hiérarchie de vos objets, classes, fonctions, variables, types et constantes de votre programme. Il vous permet aussi de :

- Afficher sous forme graphique les hiérarchies de votre application, puis de sélectionner un objet et voir les fonctions et symboles qu'il contient et dont il hérite.
- Lister les variables de votre programme, de les sélectionner et de voir leur déclaration, de lister leurs références au sein du programme, ou d'éditer leur déclaration dans le code source.

Avant d'utiliser le Browser, définissez ces options dans la boîte de dialogue Project Options (choisissez Options | Project) et compilez votre application :

- Choisissez Compiler | Debugging et cochez Debug information in OBJs
- Choisissez Compiler | Debugging et cochez Browser reference information in OBJs
- Choisissez Linker | General et cochez Include debug information.

Le Browser possède une barre d'icônes personnalisable (voir plus haut la section "Modifications des barres d'icônes" pour de plus amples informations).

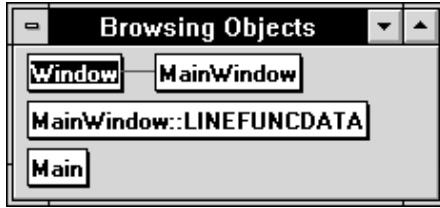
Pour lancer le Browser, choisissez Search | Browse Symbol, View | Classes, ou View | Globals. Vous pouvez aussi placer le curseur sur un symbole dans le code et choisir Search | Browse symbol. Si le programme dans la fenêtre d'édition active n'a pas été compilé, vous devez le compiler et le linker avec les informations de débogage. Si vous tentez de visualiser, avec le Browser, une définition de classe (ou tout symbole ne possédant pas d'informations symboliques de débogage), un message d'erreur s'affiche.

Vous pouvez définir des options du Browser dans la boîte de dialogue Environment Options. Choisissez Options | Environment, cliquez sur Browser et sélectionnez des options. Single window signifie que vous ne pouvez ouvrir qu'une fenêtre de Browser à la fois ; Multiple windows ouvre une fenêtre de Browser à chaque fois que vous effectuez une action liée au Browser (si vous choisissez View | Globals dans le menu principal, par exemple). Les symboles visibles sont décrits à la section "Déplacement dans les symboles globaux plus loin dans ce chapitre.

Visualisation des objets (vue générale sur les classes)

Choisissez *View | Classes* pour "prendre du recul" et voir les hiérarchies des objets de votre application, ainsi que les moindres détails. Lorsque vous choisissez *View | Classes*, le Browser dessine vos objets et montre leur relation ancêtre-descendant dans une arborescence horizontale. Les traits rouges vous aident à visualiser la relation ancêtre-descendant immédiate de l'objet sélectionné. La figure 1.2 montre la structure de l'application WHELLO.

Figure 1.2
Affichage des classes d'une application



Pour en savoir plus sur un objet, double-cliquez dessus ou appuyez sur *Entrée* après l'avoir sélectionné.

Filtres

Vous pouvez cocher des options du Browser dans la boîte de dialogue Environment Options pour sélectionner un type de symboles, mais vous devez définir ses options avant d'ouvrir des fenêtres du Browser.



Lorsque vous visualisez un symbole, les lettres qui l'identifient apparaissent dans la matrice Filters, au bas de la fenêtre du Browser. Vous pouvez vous servir de ces filtres pour sélectionner le type des symboles à visualiser (voir le tableau 1.1 pour de plus amples informations sur ces codes).

La matrice des filtres compte une colonne pour chaque lettre. Cliquez sur la ligne du haut ou du bas pour déplacer la lettre. Une lettre dans la ligne du haut signifie que le Browser doit afficher les symboles ayant cet identificateur ; une lettre dans la ligne du bas signifie que le Browser doit ignorer les symboles ayant cet identificateur).

Pour ne pas afficher certains types de symboles, cliquez sur la cellule inférieure de la colonne de la lettre (voir marge gauche). Ainsi, pour ne pas visualiser les variables de l'objet sélectionné, cliquez sur la dernière cellule de la colonne V.

Parfois, plusieurs lettres s'affichent à côté d'un symbole. La seconde lettre, placée juste après la lettre identifiant le type du symbole, complète sa description. Voir le tableau 1.1 pour connaître la liste des identificateurs.

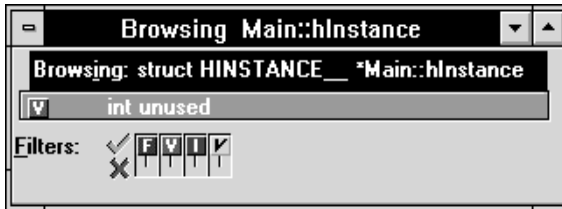
Affichage des déclarations des symboles listés

Il existe plusieurs méthodes pour voir la déclaration d'un symbole affiché dans une liste :

- Double-cliquez sur un symbole.
- Sélectionnez un symbole et appuyez sur *Entrée*.
- Sélectionnez un symbole, appuyez sur *Alt+F10* pour ouvrir le TurboMenu, puis choisissez Browse Symbol.

La déclaration s'affiche dans une fenêtre, comme le montre la figure 1.3.

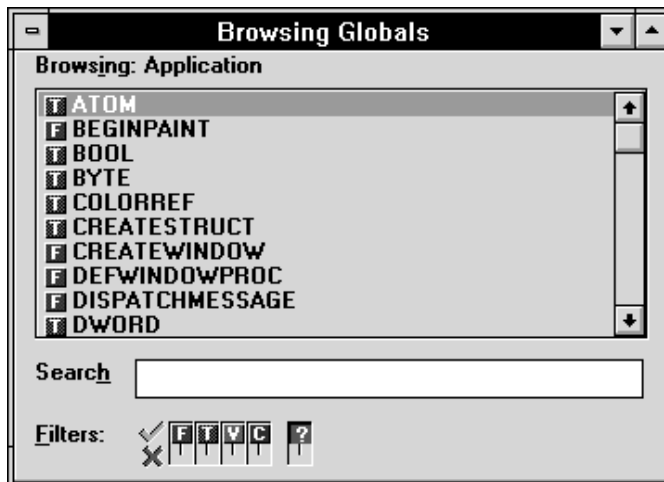
Figure 1.3
Fenêtre de déclaration de symbole



Déplacement dans les symboles globaux

Choisissez View | Globals pour ouvrir une fenêtre listant tous les symboles globaux de votre application, triés par ordre alphabétique. Le Browser liste les symboles (fonctions, variables, etc.) utilisés dans l'objet. La figure 1.4 montre les symboles globaux de WHELLO.

Figure 1.4
Affichage des symboles globaux



Une ou plusieurs lettres, symbolisant le type de l'objet, s'affichent à gauche des symboles de l'objet. Pour exclure certains symboles, servez-vous de la liste des filtres au bas de la fenêtre du Browser. Voir la section précédente, "Filtres", pour de plus amples informations.

Tableau 1.1
Symboles littéraux
du Browser

Littéral	Symbole
F	Fonctions
T	Types
V	Variables
C	Constantes intégrales
?	Débogable
I	Hérité d'un ancêtre
v	Méthode virtuelle

Vous pouvez aussi entrer des expressions régulières pour effectuer une recherche (?, * et +, par exemple).

Pour de plus amples informations sur un symbole, cliquez dessus ou utilisez les touches fléchées pour le sélectionner. Une zone de critère, au bas de la fenêtre, vous permet de taper les premières lettres du nom du symbole pour une recherche rapide. Pour voir la déclaration, sélectionnez le symbole et appuyez sur *Entrée*. Voir la section précédente pour de plus amples informations.

Utilisation des expressions régulières dans le Browser

Vous pouvez utiliser des expressions dans la case de recherche de certaines fenêtres du Browser. Le tableau 1.2 contient la liste des symboles autorisés.

Tableau 1.2
Expressions de
recherche du
Browser

Caractère	Fonction
.	Egal à un caractère, quel qu'il soit.
*	Egal à aucun ou plusieurs caractères précédents. Par exemple : * est incorrect car aucun caractère ne précède *. cré* trouve tout texte commençant par "cr". cré*e trouve "cre", "crée", "créée"

Tableau 1.2
Expressions de
recherche du
Browser
(suite)

Caractère	Fonction
+	Egal à un ou plusieurs caractères précédents. Par exemple : + est une erreur cré+ trouve tout ce qui commence par "cré" cré+e trouve "créé" et "créée"
?	Egal à aucun ou un caractère précédent. Par exemple : ? est une erreur cré? trouve tout texte commençant par "cr" cré?e trouve seulement "créé" ou "cre"

Visualisation des symboles dans votre code

Vous pouvez visualiser tout symbole dans votre code sans visualiser les hiérarchies des objets ou les listes de symboles. Choisissez l'une des méthodes ci-dessous :

- Mettez en surbrillance le symbole dans votre code et choisissez Search | Browse Symbol.
- Cliquez sur le bouton droit ou appuyez sur *Alt+F10* lorsqu'une fenêtre d'édition est sélectionnée pour afficher le TurboMenu, puis choisissez Browse Symbol.

Utilisation des outils en ligne de commande

Borland C++ contient plusieurs outils en ligne de commande vous permettant de faire la même chose que depuis l'EDI. Borland C++ inclut un compilateur, un éditeur de liens, un compilateur de ressources, un bibliothécaire, un générateur de projets (appelé MAKE), et d'autres outils en ligne de commande. La plupart de ces outils sont décrits dans ce manuel. Certains le sont dans les fichiers en ligne. Tous le sont dans l'Aide en ligne.

Vous devez utiliser les outils en ligne de commande si vous travaillez avec un éditeur DOS (Brief, par exemple). Voici la liste des outils en ligne de commande, ce qu'ils font et où en trouver une description complète :

- BCC.EXE et BCC32.EXE sont les compilateurs 16 bits et 32 bits. Ils sont décrits au chapitre 3.
- TLINK.EXE et TLINK32.EXE lient les fichiers .OBJ et .LIB pour créer des fichiers .EXE et .DLL. Ils sont décrits au chapitre 9.

- IMPLIB.EXE et TLIB.EXE vous assistent dans la création et l'utilisation des bibliothèques. Ils sont décrits au chapitre 11.
- HC31.EXE compile les fichiers de l'Aide en ligne et crée le fichier .HLP au format standard de l'aide sous Windows. Il est décrit dans l'Aide en ligne.
- BRCC.EXE, BRCC32.EXE, BRC.EXE, BRC32.EXE et RLINK.EXE sont des outils de compilation des ressources de vos applications. Ils sont décrits au chapitre 10.
- MAKE.EXE et MAKER.EXE vous assistent dans la gestion de vos projets en ne générant que les fichiers qui ont été modifiés depuis le précédent "build". Ils sont décrits au chapitre 12.

DPMI et les outils en ligne de commande

Le compilateur en ligne de commande utilise DPMI (Dos Protected Mode Interface) pour s'exécuter en mode protégé sur les machines 286, 386, i486 ou Pentium ayant au moins 640 ko de mémoire conventionnelle et 1 Mo de mémoire étendue.

Bien que Borland C++ s'exécute en mode protégé, il génère des applications capables de s'exécuter en mode réel. L'avantage du mode protégé est qu'il laisse plus de place en mémoire au compilateur Borland C++, ce qui vous permet de compiler de gros projets sans échange sur disque.

Mémoire vive et MAKESWAP.EXE

Si vous recevez des messages "Mémoire insuffisante" sous DOS (DOS n'étant pas exécuté depuis Windows) lorsque vous exécutez les outils 32 bits en ligne de commande, créez un fichier d'échange avec l'utilitaire MAKESWAP, en indiquant la taille désirée en ko :

```
MAKESWAP 12000
```

crée un fichier d'échange de 12 Mo du nom de EDPMI.SWP dans le répertoire courant, auquel les outils en ligne de commande accèdent s'ils manquent de mémoire vive. Pour paramétrer un fichier d'échange, utilisez la variable d'environnement DPMIMEM à l'invite du DOS, ou insérez la ligne suivante dans votre AUTOEXEC.BAT :

```
set DPMIMEM=SWAPFILE <emplacement du fichier>\EDPMI.SWP
```

Vous devez supprimer cette variable d'environnement avant d'utiliser les outils en ligne de commande de Borland C++ 3.1 ou d'autres exécutables 16 bits en mode DPMI (Paradox, par exemple). Entrez la commande suivante à l'invite du DOS :

```
set DPMIMEM=
```

MAKESWAP ne fonctionne que pour les sessions DOS standards, non exécutées dans une fenêtre Windows. Consultez le fichier INSTALL.TXT pour de plus amples informations.

Gestionnaire de run-times et outils en ligne de commande

Les applications Borland C++ en mode protégé (telles que BCC et BCC32) utilise le gestionnaire de run-times RTM.EXE et 32RTM.EXE. Les outils qui utilisent le gestionnaires de run-times chargent d'abord le gestionnaire, accomplissent leur tâche puis le déchargent de la mémoire. Si vous faites de nombreux appels aux outils 32 bits en ligne de commande utilisant le gestionnaire de run-times (depuis un fichier MAKE, par exemple), vous pouvez gagner du temps en chargeant le gestionnaire de run-times une fois puis en appelant les outils. Pour charger le gestionnaire, tapez 32RTM depuis la ligne de commande. Pour le décharger, tapez 32RTM -u.

Contrôle de la mémoire utilisée par le gestionnaire de run-times

Par défaut, le gestionnaire de run-times utilise toute la mémoire lorsqu'il se charge, puis l'alloue à ses clients lorsqu'ils la demandent par le biais des routines API du gestionnaire de mémoire.

Pour indiquer la mémoire utilisable par le gestionnaire de run-times, tapez la commande suivante à l'invite de DOS :

```
SET RTM=[option nnnn]
```

Le tableau ci-dessous liste les options utilisables. Le paramètre "nnnn" peut être une valeur décimale ou hexadécimale au format xAB54 ou xab54.

Tableau 1.3
Variables d'environnement contrôlant l'allocation mémoire de RTM

Option	Description
EXTLEAVE nnnn	Toujours laisser au moins nnnn ko de mémoire étendue disponible. La valeur par défaut est 640 ko.
EXTMAX nnnn	Ne pas allouer plus de nnnn ko de mémoire étendue. La valeur par défaut est 4 Go. Sous Windows, la valeur par défaut est la moitié de la mémoire disponible.
EXTMIN nnnn	Si moins de nnnn ko sont disponibles après application de EXTMAX et de EXTLEAVE, quitter l'application et afficher le message "Mémoire insuffisante". La valeur par défaut est zéro.
REALLEAVE nnnn	Toujours laisser au moins nnnn paragraphes de mémoire réelle disponible. La valeur par défaut est 64 ko (4096 paragraphes).
REALMAX nnnn	Ne pas allouer plus de nnnn paragraphes de mémoire réelle. La valeur par défaut est 1 Mo (65535 paragraphes).
REALMIN nnnn	Si moins de nnnn paragraphes sont disponibles après application de REALMAX et de REALLEAVE, quitter l'application et afficher le message "Mémoire insuffisante". La valeur par défaut est zéro.

Lancement d'autres programmes depuis l'EDI

Vous pouvez lancer des programmes, outils et utilitaires externes (Turbo Debugger, Resource Workshop, GREP, WinSight, WinSpector et Keymapper) sans quitter l'EDI. Pour ce faire, choisissez Tools | *Nom du programme* (par exemple, Tools | GREP).

Pour placer des programmes dans le menu Tools :

1. Choisissez Options | Tools. La boîte de dialogue Tools s'affiche ; elle vous montre les Outils, les Visualiseurs et les Traducteurs.
2. Cliquez sur New. Pour ajouter un outil existant (listé dans la boîte de dialogue Tools), cliquez sur Edit.
3. Entrez le nom du programme, son chemin d'accès et toute option de ligne de commande nécessaire. Vous pouvez utiliser les macros de transfert sur cette ligne de commande. Voir l'Aide en ligne pour de plus amples informations.
4. Entrez le texte du menu, qui pourra s'afficher dans les TurboMenus et dans le menu principal Tools. Si vous désirez associer un raccourci clavier à cette option, faites précéder l'une des lettres du signe "&" ; elle apparaîtra ainsi soulignée dans le menu. Par exemple, si le raccourci de l'option Fichier est F, elle apparaît sous la forme &Fichier. Pour afficher un signe perluète, entrez-le en deux ; ainsi, &&Test deviendrait &Test dans le menu final.
5. Entrez un éventuel texte d'aide, qui s'affichera dans la ligne d'état lorsque l'option sera sélectionnée.
6. Cliquez sur Advanced. La boîte de dialogue Tool Advanced Options s'ouvre.
7. Cochez Translator si le programme utilise un type de fichier pour en créer un autre (ce que fait un compilateur, par exemple). Cochez Viewer si le programme permet de visualiser un fichier (un éditeur, par exemple).
8. Cochez Place on Tools menu. Cochez Place on SpeedMenu pour que le programme apparaisse dans le TurboMenu de la fenêtre du projet (voir le chapitre 2 pour de plus amples informations sur les projets).
9. Si votre programme est un Traducteur, entrez l'extension des fichiers qu'il peut traiter (BCC traduit les fichiers .C et .CPP, . Par exemple). Utilisez le point-virgule pour séparer chaque extension, et placez le caractère ":" à la fin de la liste.

10. Entrez l'extension du fichier généré par le traducteur (".OBJ", par exemple).
11. Cliquez sur OK dans toutes les boîtes de dialogue ouvertes.
12. Choisissez Tool dans le menu principal pour vérifier que votre programme a bien été ajouté.

Utilisation du gestionnaire de projet

Borland C++ Version 4.0 intègre un nouveau gestionnaire de projet aux fonctionnalités étendues. Ce chapitre explique comment utiliser Borland C++ pour créer des applications et des projets à partir de vos fichiers source et comment utiliser les fichiers .PRJ des versions précédentes du gestionnaire de projet. Si vous utilisiez déjà une version antérieure du gestionnaire de projet, lisez attentivement ce chapitre.

Le gestionnaire de projet sait traiter des applications issues de composants divers et variés. Les applications peuvent contenir plusieurs modules source modules qui doivent être compilés avec des options différentes. Par exemple, pour créer un .EXE, les scripts de ressources doivent être compilés avec le compilateur de ressources, des bibliothèques d'import doivent être créées et les fichiers .OBJ doivent être liés avec l'éditeur de liens.

Qu'est-ce que la gestion de projets ?

Le gestionnaire de projet lit les fichiers .PRJ des versions précédentes.

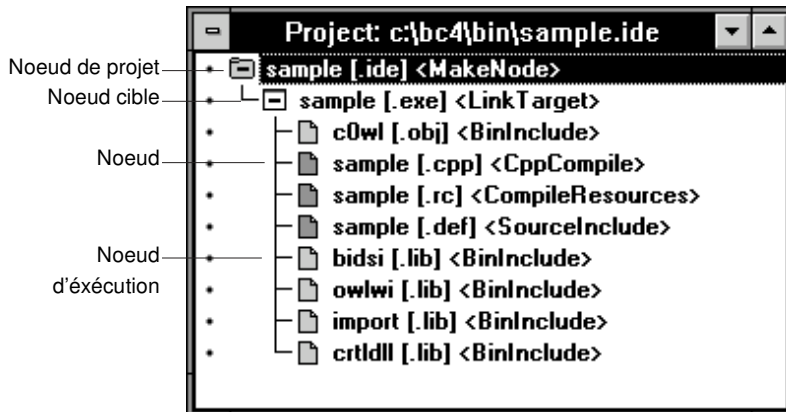
Il structure et met à jour des applications complexes en consignnant le nom de tous les fichiers et leur interdépendance dans un fichier projet ayant l'extension .IDE.

L'utilisation du gestionnaire de projet est une façon efficace de créer des projets car il ne traduit que les fichiers qui ont été modifiés depuis la dernière création du projet. Le terme *traduire* désigne la lecture d'un fichier dans un format spécifique et son écriture dans un format différent. Par exemple, le compilateur C++ est un traducteur de fichiers .CPP car il les utilise pour générer des fichiers .OBJ (voir en fin de chapitre la section traitant des traducteurs pour de plus amples informations).

Un projet peut être considéré comme un ensemble de fichiers dépendant les uns des autres. Certains fichiers contiennent du code source que vous créez ; d'autres, tels que les fichiers .OBJ, .EXE ou .DLL, sont générés par le compilateur, l'éditeur de liens ou d'autres outils dépendant de vos fichiers de code source.

Dans le gestionnaire de projet, les dépendances sont indiquées de façon graphique : il s'agit de l'arborescence de projet. A chaque niveau, les fichiers montrés dans un projet sont dépendants des fichiers placés en retrait en dessous d'eux, comme le montre la figure 2.1.

Figure 2.1
Arborescence de projet



Dans l'arborescence de projet, chaque noeud est représenté par une icône différente.

- Un noeud de projet représente l'intégralité du projet. Tous les fichiers servant à créer ce projet sont placés en dessous (un noeud de projet est similaire à une cible symbolique dans un fichier MAKE). Un projet peut contenir de nombreux noeuds cibles. Vous pouvez, par exemple, avoir un projet que vous utilisez pour créer deux applications et une DLL (trois cibles).
- Un noeud cible représente un fichier qui est généré lorsque ses noeuds dépendants sont créés (une cible est généralement le fichier .EXE ou .DLL que vous générez à partir de votre code source). Vous pouvez condenser un noeud cible pour que les noeuds qui en dépendent ne soient pas affichés.
- Un noeud désigne généralement un fichier servant à générer une cible, tel que les fichiers .C, .CPP, .H et .RC.
- Un noeud d'exécution ("runtime") désigne les fichiers communs utilisés lors de l'exécution, tels que du code de démarrage (les fichiers .LIB). Il est possible de ne pas afficher ces fichiers (voir la section "Modification de la visualisation du projet", plus loin dans ce chapitre).

Construction d'un projet

Pour créer un projet :

1. Choisissez Project | New project. Entrez le chemin d'accès et le nom du projet (huit caractères au plus), puis appuyez sur *Tab*. Vous pouvez aussi utiliser le bouton Browse pour sélectionner le fichier du projet.
2. Entrez le nom de la première cible de votre projet. Il s'agit généralement du nom du programme à créer (le fichier .EXE ou .DLL).
3. Choisissez un type de cible :
 - Application est un fichier .EXE normal.
 - Dynamic Library est un fichier .DLL.
 - EasyWin est une application en mode caractère qui s'exécute sous Windows.
 - Static Library est un fichier .LIB.
 - Import Library est un fichier .LIB.
 - Windows Help est un fichier d'aide (.HLP) qui s'active depuis une application Windows (.EXE).
4. Choisissez une plate-forme pour votre cible :
 - Windows 3.x est une application Windows 16 bits.
 - Win32 est une application 32 bits Windows NT.
 - DOS Standard est une application DOS 16 bits.
 - DOS Overlay est une application DOS 16 bits utilisant les fichiers de recouvrement ("overlays").
5. Cochez les bibliothèques que vous souhaitez utiliser dans votre application. Certaines sont cochées par défaut lorsque vous sélectionnez un type de cible (vous ne pouvez alors pas désélectionner certaines, car elles sont indispensables à la création de votre cible). Si des bibliothèques dynamiques et statiques existent, vous pouvez spécifier lesquelles vous allez utiliser (les bibliothèques dynamiques sont en général sélectionnées par défaut).
 - OWL utilise les bibliothèques ObjectWindows. Voir le *Guide du programmeur Object Windows* pour de plus amples informations.
 - Class Library utilise les bibliothèques Borland de classes de conteneur, décrites dans le guide du programmeur.

Vous pouvez utiliser TargetExpert pour modifier ces attributs.

- Runtime utilise les bibliothèques d'exécution ("run-time") listées dans le *Guide de référence*.
 - BWCC utilise les bibliothèques de contrôle personnalisées Windows de Borland. Voir l'annexe B.
 - BGI utilise l'Interface Graphique de Borland (uniquement disponible pour les applications DOS). Voir les fichier UTILS.TXT.
6. Cochez Diagnostic si vous désirez utiliser la version de diagnostic des bibliothèques (disponible pour les Class Libraries et ObjectWindows ; voir la documentation ObjectWindows pour de plus amples informations).
 7. Cochez Multithread si vous désirez utiliser la version multithread de la bibliothèque run-time. Cette option n'est disponible que si vous avez choisi la plate-forme Win32.
 8. Si votre application est pour DOS, cochez
 - Floating point pour éditer les liens dans FP87.lib.
 - Emulation pour éditer les liens dans EMU.LIB.
 - No Math pour éditer les liens dans les bibliothèques mathématiques DOS.
 9. Choisissez un modèle de mémoire pour votre cible (Target Model). Les modèles changent en fonction de votre type de cible.
 10. Cliquez sur OK pour créer le projet. Une représentation graphique s'affiche dans une fenêtre d'édition. Pour modifier les attributs de la cible, définis aux étapes 2 à 8, utilisez le TurboMenu du gestionnaire de projet (cliquez, avec le bouton droit, sur un noeud ou appuyez sur *Alt+F10*).

Voir le *Guide de référence DOS* pour de plus amples informations sur ces bibliothèques.

Le gestionnaire de projet crée une cible avec une ou plusieurs dépendances - les dépendances par défaut varient en fonction de la cible. Pour voir quels dépendances sont ajoutées, cliquez sur le bouton Advanced dans la boîte de dialogue New Project. Vous pouvez sélectionner d'autres noeuds et cliquer sur OK. En cas d'application DOS, il est possible d'indiquer le fonctionnement souhaité des segments de pile et de données.

Après avoir créé la cible initiale du projet, vous pouvez ajouter, supprimer ou réorganiser ses noeuds et ses cibles. Pour plus de détails, voir la section "Modification de l'arborescence du projet" plus loin dans ce chapitre.

Création d'un projet multi-cible

La création de projets multi-cibles est similaire à la création de projets à une seule cible :

1. Créez votre projet comme décrit à la section "Création d'un projet", au début de ce chapitre.
2. Choisissez Project | New target pour ajouter une seconde cible à votre application. La boîte de dialogue New Target s'affiche.
3. Entrez le nom de la seconde cible et choisissez son type (Standard est la valeur par défaut). Cliquez sur OK. Le gestionnaire de projet ajoute cette nouvelle cible à votre projet.

Pour inclure une DLL pour une application, placez le noeud de la DLL sous le noeud .EXE.

Pour visualiser un exemple de projet contenant deux cibles, ouvrez le fichier MULTITRG.IDE dans le répertoire EXAMPLES\IDE\MULTITRG. Ce projet crée deux versions du programme WHELLO (une version 16 bits et une version 32 bits). Ce fichier de projet contient un fichier texte qui explique comment utiliser deux (ou plus) cibles dans un projet.

Si votre projet contient plusieurs cibles, vous pouvez créer une seule cible, plusieurs cibles ou tout le projet. Voir la section "Construction d'un projet" plus loin dans ce chapitre, pour plus de précision.

Conversion des anciens projets

Le gestionnaire de projet de cette nouvelle version peut utiliser des projets issus des versions précédentes de Borland C++ pour Windows. Choisissez Project | Open project, puis entrez le nom du projet créé avec une version précédente. Il est possible de changer les attributs de recherche de * .IDE en * .PRJ pour voir les projets des versions 3.0 et 3.1.

Le gestionnaire de projet convertit l'ancien projet au nouveau format. Pensez à sauvegarder l'ancienne version si vous souhaitez continuer à utiliser une version antérieure de Borland C++. Pour enregistrer le projet, choisissez Options | Save. Cochez l'option Project, puis cliquez sur OK. Le nouveau projet conserve son nom, auquel est ajoutée l'extension .IDE.

Conversion de projets en fichiers MAKE

Vous pouvez convertir les fichiers de projets Borland C++ (.IDE) en fichiers MAKE (.MAK) depuis l'EDI. Pour ce faire :

1. Ouvrez le projet (fichier .IDE) à convertir.
2. Choisissez Project | Generate Makefile. L'EDI génère un fichier MAKE du même nom et lui ajoute l'extension .MAK. L'EDI affiche ce nouveau fichier MAKE dans une fenêtre de l'éditeur.

Modification de la visualisation du projet

Par défaut, la fenêtre d'édition affiche le noeud, la cible et les dépendances. Pour modifier l'affichage, utilisez la boîte de dialogue Options | Environment.

1. Choisissez Options | Environment.
2. Choisissez Project View. Une liste d'options s'affiche.
3. Cochez les options de votre choix. Le noeud-type `WHELLO` est modifié en fonction des options que vous sélectionnez. Il vous montre l'aspect que prendront les noeuds dans la fenêtre d'édition. La liste ci-dessous décrit chaque option :
 - **Build translator** indique le traducteur utilisé pour le noeud.
 - **Code size** indique la taille totale, en octets, des segments de code. Cette information n'apparaît que lorsque le noeud a été compilé.
 - **Data size** indique la taille totale, en octets, des segments de données. Cette information n'apparaît que lorsque le noeud a été compilé.
 - **Description** affiche une description du noeud. Pour entrer une description, utilisez la boîte de dialogue Edit Node attributes du TurboMenu.
 - **Location** indique le chemin d'accès du fichier source associé au noeud.
 - **Name** indique le nom du noeud.
 - **Number of lines** indique le nombre de lignes de code dans le fichier associé au noeud. Cette information n'apparaît que lorsque le code a été compilé.
 - **Node type** indique le type du noeud (`.cpp` ou `.c`, par exemple).
 - **feuille de style** indique quelle feuille de style est associée au noeud.
 - **Output** indique quel fichier (avec son chemin d'accès) est créé par le noeud lors de sa traduction. Ainsi, par exemple, un noeud `.CPP` crée un fichier `.OBJ`.
 - **Show runtime nodes** indique quels noeuds le gestionnaire de projet utilise lors de la création du projet. Il peut contenir, par exemple, le code et les bibliothèques de démarrage.
 - **Show project node** affiche le noeud de l'intégralité du projet. Le noeud est créé lorsque vous choisissez Project | Build all. Notez que toutes les cibles sont dépendantes du noeud de projet.

Création d'un projet

Procédez comme suit pour générer un projet :

1. Ouvrez le projet à générer en choisissant Project | Open dans le menu principal.
2. Choisissez Project | Build all pour générer tous les noeuds du projet, même s'ils sont à jour. Si vous choisissez Project | Make all, seuls les noeuds dont les dépendances ont été modifiés seront générés.

Le gestionnaire de projet génère le projet en utilisant la feuille de style des options par défaut (la "Default Project Options StyleSheet"), sauf si vous associez une feuille de style spécifique à un noeud ou si vous avez choisi d'autres options locales. Voir la section "Utilisation des feuilles de style" plus loin dans ce chapitre pour de plus amples informations.

Le gestionnaire de projet analyse la première cible, puis descend dans le projet jusqu'à ce qu'il rencontre un noeud sans dépendances. Le gestionnaire de projet crée d'abord ce noeud (et les autres noeuds de même niveau), puis remonte l'arborescence du projet.

Par exemple, si vous avez un projet avec un fichier cible .EXE dépendant d'un fichier .CPP, le gestionnaire de projet compile le fichier .CPP en un fichier .OBJ puis utilise ce fichier .OBJ pour créer le fichier .EXE.

Si vous choisissez Make all, le gestionnaire de projet vérifie, à l'aide de sa date et de son heure, si le fichier a été modifié. Si c'est le cas, le gestionnaire de projet génère à nouveau ce fichier puis remonte dans l'arborescence du projet et vérifié la date et l'heure de modification du fichier suivant. Le gestionnaire de projet analyse tous les noeuds d'un projet et génère tous ceux qui sont obsolètes.

Construction d'une partie de projet

Vous pouvez générer une partie d'un projet de trois façons différentes :

- En générant un noeud et ses dépendances,
 1. Sélectionnez le noeud à générer.
 2. Cliquez, avec le bouton droit, sur le noeud (ou appuyez sur *Alt+F10*) et choisissez Build node dans le TurboMenu. Tous les noeuds dépendants sont générés, qu'ils soient obsolètes ou non.

- Pour générer un projet en utilisant MAKE :
 1. Sélectionner le noeud à générer.
 2. Cliquez, avec le bouton droit, sur le noeud (ou appuyez sur *Alt+F10*) et choisissez Make node dans le TurboMenu. MAKE ne génère que les noeuds qui ne sont pas à jour. Pour de plus amples informations sur MAKE et ses critères de sélection des fichiers à générer, voir le chapitre 12.
- Pour traduire un seul noeud :
 1. Sélectionnez le noeud à traduire.
 2. Choisissez Project | Compile dans le menu principal ou sélectionnez la commande de traduction par défaut dans le TurboMenu. Ainsi, si vous avez sélectionné un fichier .CPP, le TurboMenu contient la commande C++ Compile, qui ne compile que le noeud sélectionné.

La commande Project | Compile traduit le noeud en cours si la fenêtre d'édition est sélectionnée. Si une fenêtre d'éditeur est sélectionnée, la commande Project | Compile traduit le texte affiché dans l'éditeur.

Modification de l'arborescence du projet

Vous pouvez modifier l'arborescence du projet à l'aide du clavier ou en utilisant les menus. Certaines commandes n'apparaissent que dans le TurboMenu. Pour afficher un TurboMenu dans la fenêtre d'édition, sélectionnez un noeud et appuyez sur *Alt+F10*. Les options disponibles dans le TurboMenu varient en fonction du type du noeud sélectionné.

Lorsque vous modifiez un projet, vous pouvez ajouter, supprimer et déplacer des cibles et des noeuds, et vous pouvez copier des noeuds. Il est également possible de modifier les attributs des noeuds et des cibles.

Modification des attributs de cible avec TargetExpert

Les attributs de cible définissent un type de cible. Il peut s'agir, par exemple, d'une DLL Windows 16 bits à transformer en 32 bits. Il est possible de modifier les attributs des cibles Standard et AppExpert, mais pas ceux des cibles Source Pools (voir la section "Utilisation des Source Pools", plus loin dans ce chapitre). De plus, vous ne pouvez pas modifier le type de la cible (de Source Pool en AppExpert, par exemple).

Pour modifier les attributs d'une cible Standard ou AppExpert :

1. Sélectionnez une cible dans la fenêtre d'édition.
2. Appuyez sur *Alt+F10* ou cliquez, à l'aide du bouton droit, sur le noeud cible.
3. Choisissez TargetExpert dans le TurboMenu. La boîte de dialogue TargetExpert s'affiche.
4. Modifiez les attributs de la cible puis cliquez sur OK. Vous trouverez plus de détails sur les attributs à la section "Création d'un projet", au début de ce chapitre.

Modification des attributs de noeud

Les attributs de noeud définissent un noeud, ainsi que les options et le traducteur à utiliser pour sa traduction. Pour modifier ces attributs :

1. Sélectionnez un noeud dans la fenêtre d'édition.
2. Appuyez sur *Alt+F10* ou cliquez, à l'aide du bouton droit, sur le noeud.
3. Choisissez Edit node attributes dans le TurboMenu. La boîte de dialogue Node Attributes s'affiche.
4. Modifiez les attributs du noeud, puis cliquez sur OK. Ces attributs, qui s'affichent généralement dans la fenêtre d'édition, sont définis ainsi :
 - **Name** indique le nom du noeud.
 - **Description** est un texte qui décrit le noeud.
 - **Style Sheet** est le nom de la feuille de style que le gestionnaire de projet utilise pour traduire ce noeud. Si <<None>> est affiché, cela signifie que le gestionnaire de projet utilise la feuille de style du parent.
 - **Translator** indique le nom du traducteur utilisé pour ce noeud. Il s'agit généralement du traducteur par défaut pour le type du noeud (Cpp Compile pour un noeud .CPP). Si vous changez de traducteur, les "builds" et les fichiers MAKE de ce noeud seront affectés. Pour plus de détails, voir la section "Traducteurs, plus loin dans ce chapitre".
 - **Node type** définit le noeud et indique les traducteurs disponibles pour ce noeud.

Ajout et suppression de noeuds

Vous pouvez aussi utiliser le TurboMenu pour ajouter des noeuds.

Pour ajouter un noeud au projet :

Appuyez sur *Alt+F10*, puis choisissez Add node. Avec cette commande, il est possible d'ajouter plusieurs noeuds.

1. Sélectionnez le noeud sous lequel vous désirez insérer le nouveau noeud. Pour qu'il s'insère sous la cible, sélectionnez le noeud cible.
2. Appuyez sur *Inser* ou cliquez, à l'aide du bouton droit, sur le noeud sélectionnez et choisissez Add node dans le TurboMenu.
3. Choisissez le(s) fichier(s) à associer à ce nouveau noeud ou entrez le nom du noeud à ajouter (si ce fichier n'existe pas dans le répertoire actif, l'EDI le crée).
4. Cliquez sur OK. Le nouveau noeud s'affiche sous le noeud sélectionné.



Pour ajouter plusieurs noeuds à un projet :

1. Ouvrez le gestionnaire de fichiers de Windows et sélectionnez les fichiers à ajouter sous forme de noeuds à votre projet. Vous devez voir la fenêtre d'édition dans l'EDI.
2. Déplacez les fichier depuis le gestionnaire de fichiers. Le gestionnaire de projet les ajoute automatiquement sous le noeud sélectionné.

Pour supprimer un noeud du projet, sélectionnez-le et appuyez sur *Suppr*, ou choisissez Delete node dans le TurboMenu. Pour supprimer plusieurs noeuds en même temps, sélectionnez-les en appuyant sur *Ctrl* ou *Maj* tout en cliquant dessus - avec le bouton gauche -, puis appuyez sur *Suppr*. Avant d'effacer les noeuds sélectionnés, le gestionnaire de projet vous demande confirmation.

Ajout et suppression de cibles

Pour ajouter une cible au projet :

1. Choisissez Project | New target dans le menu principal.
2. Entrez le nom de la nouvelle cible et choisissez son type :
 - **Standard** (valeur par défaut) peut être un exécutable, une DLL ou un autre fichier.
 - **AppExpert** est une application ObjectWindows. Voir le chapitre AppExpertCh pour de plus amples informations sur ce type de cible.
 - **Source Pool** est un ensemble de fichiers pouvant être référencés dans d'autres cibles (voir la section "Utilisation des Source Pools", plus loin dans ce chapitre).

3. Cliquez sur OK. Si la cible est de type Standard, la boîte de dialogue TargetExpert s'affiche, vous permettant de définir d'autres options (voir la page la section "Création d'un projet" au début de ce chapitre pour de plus amples informations sur les options disponibles). Si la cible est de type AppExpert, reportez-vous au chapitre 4. Si elle est de type Source Pool, elle est ajoutée au projet et vous pouvez lui ajouter des noeuds immédiatement.

Pour supprimer une ou plusieurs cibles :

1. Sélectionnez la cible et activez le TurboMenu (cliquez sur la cible à l'aide du bouton droit, ou appuyez sur *Alt+F10*).
2. Choisissez Delete node.
3. Le gestionnaire de projet vous demande de confirmer la suppression. Cliquez sur OK. *Notez bien que vous ne pouvez pas annuler cette opération.*

Déplacement des noeuds et des cibles

Vous pouvez déplacer des noeuds ou des cibles de plusieurs façons :

- Faites glisser le noeud avec la souris. Il se place sous le noeud sélectionné lorsque vous relâchez le bouton.
- Sélectionnez le noeud et appuyez sur *Alt* et les touches fléchées. Le noeud se déplace vers le haut ou le bas, le long des noeuds *visibles*. Vous pouvez aussi utiliser la touche *Alt* et les touches fléchées *Gauche* et *Droite* pour déplacer un noeud à travers les niveaux de dépendance. Par exemple, si un fichier d'en-tête est dépendant d'un fichier .CPP (le fichier .H est placé sous le fichier .CPP dans la fenêtre d'édition), vous pouvez amener l'en-tête au même niveau que le fichier .CPP en sélectionnant le premier et en appuyant sur *Alt+Gauche*.

Copie de noeuds

Les noeuds peuvent être copiés complètement ou par référence. Une copie complète vous permet de prendre le noeud et ses attributs et d'en placer une copie conforme, mais distincte, à un autre emplacement du projet. Une telle copie hérite des attributs de son parent, sauf si vous passez outre certaines options.

Une copie de référence vous permet de prendre un noeud et ses dépendances et de les référencer dans un autre emplacement du projet ; une copie de référence n'est pas une copie distincte : si vous ajoutez ou supprimez des dépendances dans l'original, la copie de référence est mise à jour automatiquement. Une telle copie est une copie d'un noeud et de ses dépendances.

Pour effectuer une copie complète d'un noeud :

1. Sélectionnez le(s) noeud(s) à copier (utilisez *Maj* ou *Ctrl* pour étendre la sélection). Si un noeud compte des dépendances, ils sont automatiquement copiés, sans que vous ayez à les sélectionner.
2. Maintenez la touche *Ctrl* enfoncée et faites glisser les noeuds sélectionnés à l'endroit désiré.
3. Lorsque vous relâchez le bouton, les noeuds copiés apparaissent. Si vous modifiez le noeud original, la copie complète n'est pas modifiée.

Pour effectuer une copie de référence,

1. Sélectionnez le noeud à copier. Même si vous ne les sélectionnez pas, les dépendances du noeud sélectionné seront automatiquement copiés.
2. Maintenez la touche *Alt* enfoncée et faites glisser le noeud à l'endroit choisi pour la copie de référence.
3. Lorsque vous relâchez le bouton, le noeud copié apparaît, dans une police maigre (non grasse). Ceci vous rappelle qu'il s'agit d'une copie de référence et non pas d'une copie complète. Si vous modifiez le noeud original, *toutes* les copies de référence sont mises à jour.

Mise en garde ! Si vous supprimez un noeud original, *toutes* les références à ce noeud seront aussi effacées, sans possibilité d'annulation.

Utilisation des Source Pools

Un "Source Pool" (Ensemble de Sources) est un groupe de noeuds. Une cible Source Pool n'est pas générée, mais peut être référencée lors d'une création. Les Source Pools permettent à différentes cibles d'utiliser le même code source. Ils sont par exemple très utiles si vous créez deux applications, l'une 16 bits et la seconde 32. Pour voir un exemple de Source Pools, ouvrez le projet SRCPOOL.IDE dans le répertoire EXAMPLES\IDE\SRCPOOL. Il contient un fichier texte qui explique comment le Source Pool est ici utilisé.

Les Source Pools peuvent contenir plusieurs fichiers que vous souhaitez copier par référence dans votre projet. Par exemple, vous pouvez avoir plusieurs en-têtes à placer à divers endroits du projet. Si vous placez ces fichiers dans un Source Pool, puis copiez ce dernier par référence, il vous suffira de modifier le Source Pool original. Ainsi, si vous ajoutez un en-tête dans le Source Pool, toutes les copies référencées seront immédiatement mises à jour.

Les Source Pools sont particulièrement utiles pour assigner une seule feuille de style à plusieurs cibles. Par exemple, si vous avez trois cibles dans un projet et souhaitez que toutes les cibles utilisent la même feuille de style, vous pouvez soit associer la feuille de style à chaque cible, individuellement, soit placer les cibles sous un Source Pool puis associer la feuille de style au noeud du Source Pool. Si vous désirez réassigner une feuille de style (pour, par exemple, compiler sans informations de débogage), il vous suffira alors de réassigner la feuille de style au Source Pool, et non pas à chaque cible.

Définition des options d'un projet

Lorsque vous avez créé un projet, il se peut que vous souhaitiez modifier les options de création par défaut. Ces options indiquent au gestionnaire de projet comment il doit créer votre projet, en spécifiant, par exemple, si vous désirez inclure les informations de débogage dans votre application.

Pour modifier les options d'un projet :

1. Choisissez Options | Project. Une boîte de dialogue s'affiche.
2. Modifiez certaines options. Voir les chapitres 1 et 3 pour de plus amples informations sur les options.
2. Cliquez sur OK pour valider vos modifications.

Lorsque vous créez votre projet, les options définies s'appliquent à l'intégralité du projet. Si vous créez un nouveau projet, il utilise par défaut les options du dernier projet ouvert.

Vous souhaiterez parfois définir des options spécifiques pour l'un des noeuds du projet (pour, par exemple, compiler un seul fichier sans informations de débogage). Pour activer des options différentes pour un noeud, utilisez Local Override ou une feuille de style.

Dérogation locale

Il est possible, localement, d'utiliser d'autres options que celles qui ont été définies pour l'ensemble du projet. Cette fonctionnalité s'appelle la dérogation locale ("Local Override"). Ceci est utile lorsque vous désirez appliquer des options différentes à un noeud précis. Si vous désirez déroger à de nombreuses options, utilisez une feuille de style séparée (voir la section qui suit pour de plus amples informations sur les feuilles de style).

Pour déroger à une option,

1. Choisissez le noeud auquel vous souhaitez appliquer des options distinctes.
2. Cliquez, à l'aide du bouton droit, sur le noeud (ou appuyez sur *Alt+F10*) et choisissez Edit local options dans le TurboMenu. La boîte de dialogue Style Sheet s'affiche, détaillant les options utilisées pour le noeud.
3. Sélectionnez l'option à modifier. La case Local Override est automatiquement cochée.
4. Cliquez sur OK.

Mise en garde !

Pour annuler une dérogation, désélectionner Local Override. La coche dans Local Override n'est visible que lorsque le curseur se trouve dans l'option à laquelle vous dérogez, ce qui rend difficile son repérage. La case Local Override est gris foncé si aucune des options de cette section n'est remplacée par une autre. Elle devient gris clair si l'une des options est remplacée, mais vous devez quand même les sélectionner l'une après l'autre pour savoir de laquelle il s'agit.

Si vous dérogez à plus de deux options, il est souhaitable d'utiliser une feuille de style propre à ce noeud.

**Utilisation des
feuilles de style**

Les feuilles de style sont un ensemble d'options de création (les "build options") s'appliquant à un projet. Chaque projet utilise un jeu d'options par défaut, qui sont enregistrées dans une feuille de style. Par défaut, un projet utilise la feuille de style nommée "Default Project Options" (Options par défaut pour un projet). Les paramètres contenus dans la feuille de style influent sur la façon dont sera créé le projet. Si tous les composants du projet peuvent être créés avec les mêmes options, vous pouvez les définir à l'aide de la boîte de dialogue Options | Project. Ceci vous permet de modifier la feuille de style "Default Project Options". Si vous ne souhaitez modifier que les options d'un seul noeud, utilisez la fonction Local Override ; si vous vous apercevez que vous utilisez souvent cette dernière, il peut être préférable d'utiliser une feuille de style.

Lors de la création d'un projet, c'est la feuille de style du projet qui est utilisée, sauf si le noeud en cours de création est associé à une feuille de style distincte ou utilise Local Override (voir la section précédente pour de plus amples informations sur Local Override). Il est possible d'utiliser des feuilles de style *et* Local Override, notamment dans le cas où vous avez associé une feuille de style à vos cibles mais souhaitez la modifier légèrement pour traiter un noeud placé sous la cible.

Lorsque le gestionnaire de projet crée un noeud, il utilise la feuille de style du noeud, ainsi que toute option définie dans Local Override. Si aucune feuille de style n'est associée au noeud, le gestionnaire de projet utilise la feuille de style du parent du noeud. Si celui-ci ne comporte pas non plus de feuille de style, le gestionnaire de projet analyse le parent suivant, jusqu'à ce qu'il en trouve une. En dernier ressort, c'est la feuille de style du projet qui est utilisée.

En règle générale, les différents noeuds d'un projet doivent être créés avec des options différentes. Vous compilez, par exemple, les fichiers .C avec certaines options, les fichiers .CPP avec d'autres, ou vous créez une cible au format 16 bits et une autre au format 32 bits. A titre d'exemple, ouvrez le projet STYLESHT.IDE situé dans le répertoire \EXAMPLES\IDE\STYLESHT. Ce fichier utilise une feuille de style différente pour chaque cible (deux versions du programme WHELLO). Ce projet contient en outre un fichier texte décrivant l'utilisation des feuilles de style.

**Association d'une
feuille de style à un
noeud**

Le gestionnaire de projet contient plusieurs feuilles de style utilisables, mais vous pouvez également créer les vôtres. Pour associer une feuille de style à un noeud :

1. Sélectionnez le noeud et cliquez dessus, à l'aide du bouton droit.
2. Choisissez Edit node attributes.
3. Sélectionnez une feuille de style dans la boîte liste qui s'affiche.
4. Cliquez sur OK.

Vous pouvez aussi cliquer sur le bouton Styles pour créer une feuille de style. Reportez-vous à la section qui suit pour de plus amples informations sur la création de vos propres feuilles de style.

**Création d'une
feuille de style**

Pour créer une feuille de style dans un projet :

1. Choisissez Options | Style Sheets dans le menu principal.
2. Cliquez sur Create. Entrez le nom de la feuille de style et appuyez sur *Entrée*.
3. Cliquez sur Edit. La boîte de dialogue Style Sheet s'ouvre.
4. Modifiez, si vous le souhaitez, certaines options de votre nouvelle feuille de style. La plupart de ces options sont décrites au chapitre Options.
5. Cliquez sur OK pour valider vos modifications.

**Modification d'une
feuille de style**

Une feuille de style peut être modifiée, renommée ou copiée. Choisissez Options | Style Sheets pour afficher la boîte de dialogue Style Sheets. Les options disponibles sont les suivantes :

- Compose vous permet de créer une feuille de style contenant des options combinées à partir d'une ou de plusieurs feuilles de style :
 1. Créez une nouvelle feuille de style (cliquez sur New et entrez son nom), puis cliquez sur Compose.
 2. Sélectionnez la feuille de style à inclure dans cette nouvelle feuille, puis cliquez sur Add.
 3. Répétez l'opération aussi souvent que nécessaire, puis cliquez sur OK. Il n'est pas possible de modifier une feuille de style composée, mais vous pouvez à nouveau cliquer sur Compose pour ajouter ou supprimer des feuilles de style.
- Pour copier une feuille de style :
 1. Sélectionnez une feuille de style et cliquez sur Copy.
 2. Entrez le nom de la feuille de style de destination puis cliquez sur OK. Il devient alors possible de cliquer sur Edit pour modifier certaines de ses options. La copie représente la solution la plus rapide pour créer une feuille de style dont les paramètres sont semblables, dans leur majorité, à ceux d'une feuille de style existante.
- Pour modifier une feuille de style :
 1. Sélectionnez-la et cliquez sur Edit.
 2. Modifiez les options de votre choix et cliquez sur OK.
- Pour renommer une feuille de style :
 1. Sélectionnez-la et cliquez sur Rename.
 2. Entrez le nouveau nom de la feuille de style et cliquez sur OK.
- Pour supprimer une feuille de style, sélectionnez-la et cliquez sur Remove.

**Partage de feuille
de styles**

Si vous créez des feuilles de style pour un projet, puis choisissez Project | New project, le nouveau projet hérite des feuilles de style (ainsi que des outils et des options) de l'ancien projet. Cependant, si vous fermez un projet ou relancez l'EDI, vous devrez ouvrir à nouveau le projet avec ses feuilles de style, puis créer un nouveau projet pour que celui-ci hérite des feuilles de style.

Mise en garde !

Il existe une autre façon de partager des feuilles de style entre plusieurs projets. A chaque fois que vous créez un fichier de projet (.IDE), vous créez aussi un fichier .PDL (Project Description Language) qui contient des informations sur les feuilles de style et les outils utilisés dans ce projet. Attention cependant : ne modifiez ce fichier que si vous êtes parfaitement familier avec sa syntaxe.

Lorsque vous ouvrez un fichier de projet, le gestionnaire de projet ouvre le fichier .PDL associé au fichier .IDE.

Pour partager une feuille de style entre plusieurs projets :

1. A l'aide d'un éditeur de texte, ouvrez le fichier .PDL contenant la feuille de style à partager.
2. Localisez le nom de la feuille de style. Par exemple, si vous avez créé la feuille de style MONSTYLE, une section du fichier .PDL commence par la chaîne `StyleSheet= "MONSTYLE"`.
3. Copiez le texte compris entre les accolades de début et de fin. Vous pouvez copier plusieurs feuilles de style.
4. Ouvrez le fichier .PDL dans lequel vous désirez copier la feuille de style.
5. Placez-vous dans la section des feuilles de style et collez le texte à la fin de la liste des feuilles de style.
6. Enregistrez le fichier .PDL modifié. Lorsque vous ouvrirez le projet associé avec ce fichier modifié, les feuilles de style que vous venez de coller seront affichées lorsque vous choisirez Options | Style Sheets.

Visualisation des options d'un projet

Etant donné que chaque noeud peut avoir sa propre feuille de style *et* qu'il est possible de déroger aux options de cette feuille de style, il est indispensable de pouvoir visualiser les options des noeuds.

Pour connaître la hiérarchie des options :

1. Cliquez, à l'aide du bouton droit, sur un noeud du projet et choisissez View options hierarchy.
La boîte de dialogue Options Hierarchy s'ouvre. Elle indique tous les noeuds contenus dans le projet, ainsi que leurs options. Les noeuds d'auto-dépendance ne sont pas listés. Comme dans la fenêtre d'édition, il est possible d'étendre et de condenser la liste des noeuds.
2. Cliquez sur le noeud à visualiser. Ses options s'affichent à sa droite.

3. Pour modifier une option, faites un double-clic dessus ou sélectionnez-la et cliquez sur Edit. Si l'option appartient à une feuille de style, vous modifierez toute la feuille de style. S'il s'agit d'une option "Local Override", vous modifierez les options Local Override du noeud sélectionné.
4. Pour refermer la boîte de dialogue, cliquez sur Close.

La liste des options indique le nom du noeud entre crochets, suivi du nom de la feuille de style du noeud. Elle indique également les options auxquelles vous avez dérogé. Cette hiérarchie vous permet de savoir quelles sont les options qui sont passées aux noeuds dépendants.

Traducteurs

Un traducteur est un programme qui modifie le type d'un fichier. Le compilateur, par exemple, traduit les fichiers .C et .CPP en fichiers .OBJ, et l'éditeur de liens traduit les fichiers .OBJ, .LIB, .DEF et .RES en un fichier .EXE.

Le gestionnaire de projet vous permet de définir vos propres traducteurs. Les traducteurs que vous ajoutez à un projet restent dans son fichier de projet ; ils ne sont pas intégrés de façon permanente à l'EDI. Cependant, les traducteurs, les feuilles de styles, les visualiseurs et tout autre outil peuvent être transférés à un autre projet : si un fichier de projet ouvert contient des outils ou des feuilles de style de votre cru, le prochain projet que vous créez (par Project | New project) héritera de ces outils. Voir la page plus haut la section "Partage de feuilles de style" pour de plus amples informations sur le partage d'outils entre projets.

Pour visualiser les traducteurs par défaut, choisissez Options | Tools. Ceci permet également de connaître les visualiseurs et autres outils installés.

Installation d'un traducteur

Pour installer un traducteur :

1. Choisissez Options | Tools. La boîte de dialogue qui s'affiche liste les outils, les visualiseurs et les traducteurs. Il est possible aussi d'installer des traducteurs en choisissant Build Attributes dans le TurboMenu du gestionnaire de projet.
2. Cliquez sur New.

3. Entrez le nom du traducteur, son chemin d'accès et ses éventuelles options de ligne de commande. Vous pouvez utiliser des macros de transfert dans la ligne de commande. Pour de plus amples informations sur ces macros, voir l'Aide en ligne.
4. Entrez le texte du menu. Ce texte s'affichera dans les TurboMenus et dans le menu principal des Outils ("Tools"). Pour assigner une touche de raccourci à ce menu, faites précéder l'une des lettres du caractère "&". Cette lettre sera soulignée dans le menu. La touche de raccourci du menu File, par exemple, est F. Dans le texte du menu, File serait entré sous la forme &File. Pour afficher le caractère "&" dans un menu, entrez deux fois ce caractère (par exemple, &&Test pour obtenir &Test).
5. Tapez un éventuel texte d'aide ; celui-ci s'affichera dans la ligne d'état lorsque le curseur sera sur l'option du menu.
6. Cliquez sur Advanced pour ouvrir la boîte de dialogue Tool Advanced Options.
7. Cochez Translator.
8. Cochez l'option Place on Tools menu si vous désirez que ce traducteur s'affiche dans le menu principal des outils. Cochez Place on TurboMenu si vous souhaitez qu'il s'affiche lorsque vous cliquez, à l'aide du bouton droit, sur un noeud associé à votre traducteur.
9. Si vous souhaitez que le traducteur puisse traiter les cibles, cochez Target translator. Lorsque vous utilisez ce traducteur, le noeud devient une cible et le fichier traduit est enregistré dans le répertoire Final. Si vous ne cochez pas cette option, le fichier traduit est enregistré dans le répertoire intermédiaire.
10. Entrez l'extension des fichiers que vous souhaitez associer avec ce traducteur. BCC, par exemple, est le traducteur des fichiers .C et .CPP, aussi l'option Translate From contiendrait-elle .c ; .cpp : . Le caractère ";" sépare les diverses extensions et le caractère ":" indique la fin de la liste.
11. Entrez l'extension du fichier traduit généré. BCC, par exemple, convertissant les fichiers .CPP en fichiers .OBJ, .obj est affiché dans la case Translate to.
12. Si vous voulez que ce nouveau traducteur soit utilisé par défaut pour un type de noeud, entrez l'extension du fichier et le caractère ":" dans la case Default for.
13. Cliquez sur OK.

**Utilisation de
Special dans le
TurboMenu**

Dans le TurboMenu, certains noeuds ont une commande Special qui liste les autres traducteurs pour le type de noeud sélectionné. Vous voyez, par exemple, les commandes Assemble Output et Preprocess si un noeud .CPP est sélectionné, mais vous voyez la commande Implib si c'est un noeud .DLL qui est sélectionné.

**Installation de
visualiseurs et
d'outils**

Un visualiseur vous permet de voir le contenu du noeud sélectionné. Ainsi, un éditeur est un visualiseur qui vous permet de parcourir le code d'un fichier .CPP. Dans le TurboMenu d'un noeud .CPP, la commande Text Edit est disponible. L'éditeur par défaut pour la vue Text Edit est l'éditeur de l'EDI.

Pour visualiser un noeud, vous pouvez :

- Faire un double-clic sur le noeud de votre choix dans la fenêtre d'édition, ou
- Cliquer dessus, à l'aide du bouton droit, et choisir View pour afficher la liste des visualiseurs utilisables.

Les visualiseurs disponibles sont fonction du type du noeud sélectionné. Ainsi, Resource Workshop peut visualiser les fichiers .RC. Il n'est pas possible d'afficher un noeud .EXE dans un éditeur de texte, mais vous pouvez l'ouvrir avec le débogueur intégré, avec Turbo Debugger pour Windows, avec ObjectBrowser ou sous la forme d'un exécutable.

Les outils sont des applications que vous souhaitez lancer depuis l'EDI (TurboDebugger ou GREP, par exemple). La procédure d'installation des visualiseurs et des outils est la même que pour les traducteurs. Pour de plus amples informations, reportez-vous à la section "Traducteurs" de ce chapitre.

Compilation

Les applications peuvent être compilées dans l'EDI (Environnement de Développement Intégré) ou au moyen du programme BCC.EXE (applications 16 bits) ou BCC32.EXE (applications 32 bits) lancé à partir de la ligne de commande. Les options du compilateur vous permettent de contrôler la génération du code en définissant le type d'application à créer (une version de débogage par exemple), l'emplacement des fichiers d'en-tête et des bibliothèques de liaison, la gestion du code C++ et bien d'autres aspects.

Ce chapitre se divise en trois sections :

- Compilation dans l'EDI ;
- Compilation au moyen du programme BCC ou BCC32 ;
- Options du compilateur.

La table des options indique les options du compilateur en ligne de commande et les options correspondantes dans l'EDI. Les options de compilation sont regroupées par sujet dans la boîte de dialogue Options | Projet où elles font l'objet d'une description plus détaillée.

Compilation dans l'EDI

Cette section vous explique comment compiler de simples programmes au moyen des options de compilation dans l'EDI. Pour apprendre à compiler des programmes plus complexes, reportez-vous au chapitre 2.

La barre d'icônes de l'EDI contient trois boutons de compilation qui correspondent à des commandes de menu :



Project | Compile (*Alt+F9*) compile le code dans la fenêtre d'édition sélectionnée au moyen des options de compilation définies dans la boîte de dialogue Project Options. Si vous choisissez une fenêtre de projet, tous les noeuds sélectionnés dans le projet sont traduits ; les noeuds fils ne sont traduits que s'ils ont été sélectionnés (pour plus de détails sur les traducteurs, reportez-vous au chapitre 2).



Project | Make All (*F9*) traduit tous les noeuds périmés d'un projet. Si aucun projet n'est ouvert, tous les fichiers contenus dans les fenêtres d'édition sont compilés au moyen des traducteurs de projet par défaut.



Project | Build All traduit tous les noeuds d'un projet - même s'ils sont à jour. La commande Project | Build All compile le projet en commençant par son premier noeud. Pour interrompre l'opération, cliquez sur Cancel.

Utilisation des options de compilation dans l'EDI

Il existe deux moyens de définir les options de compilation dans l'EDI :

- En sélectionnant Options | Project pour définir les options requises dans la boîte de dialogue. Ces options sont utilisées lorsque vous effectuez une compilation sans avoir chargé de fichier de projet. Si un projet est chargé, les options sont appliquées à l'ensemble du projet durant sa compilation.
- Les options de projet doivent être définies localement pour chaque fichier en utilisant le gestionnaire de projet. Pour plus de détails sur les options locales, reportez-vous au chapitre 2.

Exemple : pour compiler un code qui génère une application 32 bits destinée à un processeur 80386 dans une fenêtre d'édition (sans projet chargé), procédez de la manière suivante :

1. Sélectionnez la fenêtre d'édition qui contient le code à compiler.
2. Sélectionnez la commande Options | Project dans le menu.
3. Dans la boîte de dialogue Project Options, cliquez sur le sujet 32-bit Compiler pour afficher les sous-sujets, puis cliquez sur Processor.
4. Les options réservées aux processeurs 32 bits s'affichent sur la droite. Cochez l'option 80386. Elle demeure en vigueur jusqu'à ce que vous choisissiez un autre type de processeur ou que vous sortiez de l'EDI. Pour en faire l'option par défaut (qui sélectionnera automatiquement une application 32 bits pour processeur 80386 à chaque opération de compilation), choisissez Options | Save dans le menu principal.
5. Cliquez sur le bouton de compilation dans la barre d'icônes, ou sélectionnez Project | Compile (cette commande compile le code dans la fenêtre d'édition courante si aucun projet n'a été chargé).

Utilisation des compilateurs en ligne de commande

Cette section vous explique comment utiliser les compilateurs en ligne de commande (BCC.EXE et BCC32.EXE). BCC.EXE est une application 16 bits qui génère du code 16 bits, alors que BCC32.EXE, comme son nom l'indique, est une application 32 bits générant du code 32 bits. BCC et BCC32 fonctionnent de la même manière, mais possèdent des valeurs par défaut différentes (comme indiqué dans la table des option) et génèrent un code différent. Sauf indication contraire, les instructions et options spécifiées pour BCC s'appliquent également à BCC32.

Vous pouvez utiliser BCC pour envoyer des fichiers à TLINK ou TASM (fichiers .ASM si vous avez installé TASM sur votre machine). La syntaxe générale de BCC.EXE est la suivante :

```
:BCC [option [option...]] fichier [fichier...]
```

Pour obtenir la liste des options de compilation les plus couramment utilisées, tapez BCC (sans préciser d'option ni de nom de fichier) et appuyez sur Entrée. La commande BCC doit être séparée de chaque option ou nom de fichier par au moins un espace. Chaque option doit en outre être précédée d'un tiret (-) ou d'une barre oblique (/); à titre d'exemple, pour spécifier un type de chemin d'inclusion, vous devez spécifier :

```
BCC -Ic:\code\Fich_h
```

Les options et noms de fichier spécifiés en ligne de commande ont priorité sur les valeurs par défaut contenues dans les fichiers de configuration.

Par défaut, BCC compile les fichiers sans extension et les fichiers .CPP en tant que fichiers C++. Les fichiers ayant l'extension .C ou une extension autre que .CPP, .OBJ, .LIB ou .ASM sont compilés en tant que fichiers C.

BCC.EXE et BCC32.EXE possèdent des options actives par défaut (ces options sont signalées par un petit carré noir dans le tableau 3.1 dans la table des options). Pour désactiver une option par défaut, tapez BCC *-option-*.

Par défaut, BCC essaie de réaliser une édition de liens avec un fichier de définition de module du même nom que le fichier exécutable. Utilisez TLINK pour lier un fichier de définition de module portant un nom différent. Il n'est pas possible de lier plus d'un fichier de définition de module.

Fichiers de configuration

Si vous avez l'occasion d'utiliser régulièrement un ensemble d'options particulier à partir de la ligne de commande, vous pouvez les définir dans un *fichier de configuration* (fichier texte ASCII standard). Les options doivent être séparées les unes des autres par un espace et peuvent figurer sur une seule ligne ou plusieurs.

TURBOC.CFG
configure BCC.EXE
et BCC32.CFG
configure
BCC32.EXE ; les
fichiers de projet
(.IDE) configurent
l'EDI.

Par défaut, BCC.EXE utilise le fichier de configuration TURBOC.CFG, tandis que BCC32.EXE utilise le fichier BCC32.CFG (les options par défaut correspondants sont signalées par un petit carré noir dans le tableau 3.1). Les compilateurs commencent par rechercher les fichiers .CFG dans le répertoire à partir duquel vous avez lancé BCC, puis dans le répertoire où est stocké BCC.EXE ou BCC32.EXE.

Vous pouvez créer de multiples fichiers de configuration ou modifier TURBOC.CFG. Pour utiliser un fichier de configuration, tapez + [chemin] fichier sur la ligne de commande BCC. Pour utiliser le fichier de configuration MACONFIG.CFG, vous devez donc entrer la commande suivante sur la ligne de commande :

```
BCC +C:\MACONFIG.CFG moncode.cpp
```

Les options définies en ligne de commande BCC ont priorité sur celles qui sont fournies par les fichiers de configuration.

Fichiers de recherche

Pour spécifier plusieurs options ou fichiers sur la ligne de commande, entrez-les dans un *fichier de recherche* (fichier texte ASCII standard). Les fichiers de recherche vous permettent d'utiliser une ligne de commande plus longue que celle disponible sur la plupart des systèmes d'exploitation.

Pour utiliser un fichier de recherche :

1. Entrez les options en ligne de commande que vous souhaitez utiliser dans un fichier et sauvegardez-le. Ces options peuvent être définies sur une ou plusieurs ligne dans le fichier, mais doivent être séparées les unes des autres par des espaces. Les fichiers de recherche fournis avec Borland C++ ont l'extension .RSP.
2. Tapez BCC @@[chemin]fichrech.rsp.

Vous pouvez également spécifier plus d'un fichier de recherche au moyen de la commande suivante :

```
BCC @@[chemin]fichrech.rsp@[chemin]autresfic.rsp.
```

Les options définies en ligne de commande ont priorité sur les options correspondantes et les noms de fichier spécifiés dans un fichier de recherche.

Règles de priorité des options

BCC.EXE et BCC32.EXE évaluent les options de gauche et à droite selon les règles suivantes :

- Lorsqu'une option est dupliquée, à l'exception de **-I** et de **-L**, la dernière valeur attribuée est prise en compte.
- Si vous définissez plusieurs options **-L** ou **-I** sur la ligne de commande, le résultat est cumulé : le compilateur consulte tous les répertoires listés de gauche à droite.
- Les options en ligne de commande ont priorité sur celles qui sont définies dans les fichiers de configuration et de recherche.

Options de compilation

Le tableau 3.1 des options dresse la liste des options de compilation disponibles dans l'EDI et en ligne de commande. La plupart des options de l'EDI sont affichées dans la boîte de dialogue Project Options ; si elles n'y figurent pas, l'option équivalente de l'EDI est mentionnée entre signes < et >. Dans l'EDI, une option est répétée lorsqu'elle peut être activée ou désactivée par une case à cocher. Pour obtenir une description détaillée de chaque option, consultez la page correspondante spécifiée dans le tableau.

Les options par défaut des compilateurs en ligne de commande 16 et 32 bits sont signalées par un petit carré noir (■) ; les options applicables uniquement à l'un ou l'autre des compilateurs sont également signalées, respectivement, par ■₁₆ et ■₃₂ par une puce carrée pour b16 et b32. A noter que dans l'EDI, les options par défaut sont différentes. La principale différence dans les options par défaut est que l'EDI compile en utilisant les informations de débogage et de Browser et qu'il produit des fichiers compilés plus volumineux que les compilateurs en ligne de commande. (Les applications ont toutefois la même taille lorsque vous utilisez les mêmes options dans l'EDI et en ligne de commande.)

Tableau 3.1 : Récapitulatif des options

Option	Page	Equivalent dans l'EDI	Description
@ <i>fichier</i>	49	[Nom du fichier de projet]	Lit les options du compilateur dans le fichier de recherche spécifié.
+ <i>fichier</i>	48	<aucune>	Utilise le fichier de configuration secondaire précisé.
-1	90	<aucune>	Génère des instructions 80186.
■ ₁₆ -2	68	16-bit Compiler Processor 80286	Génère des instructions compatibles avec le mode protégé 80286 (16 bits uniquement).
-3	68	16-bit Compiler Processor 80386	Génère des instructions compatibles avec le mode protégé 80386 16 bits (option BCC).
■ ₃₂ -3	76	32-bit Compiler Processor 80386	Génère des instructions compatibles avec le mode protégé 80386 32 bits (option BCC32).
-4	68	16-bit Compiler Processor i486	Génère des instructions compatibles avec le mode protégé 80486 16 bits (option BCC).
-4	76	32-bit Compiler Processor i486	Génère des instructions compatibles avec le mode protégé 80486 32 bits (option BCC32).
-5	76	32-bit Compiler Processor Pentium	Génère des instructions compatibles avec le mode protégé Pentium 32 bits.
-A	66	Compiler Source ANSI	Conformité au langage ANSI.

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
-A-, -AT	66	Compiler Source Borland extensions	Conformité au langage BC.
-AK	66	Compiler Source Kernighan and Ritchie	Conformité au langage Kernighan et Ritchie.
-AU	66	Compiler Source UNIX V	Conformité au langage UNIX V.
-an	76	16- or 32-bit Compiler Processor Byte, Word, Double Word	Alignement sur n : 1 = octet, 2 = mot, 4 = double mot (32 bits uniquement).
■ -a-	68	16-bit Compiler Processor Byte	Alignement sur un octet.
-B	90	<aucune>	Compile et appelle l'assembleur pour traiter le code assembleur.
■ -b	63	Compiler Code Generation Allocate enums and ints	enums à la taille d'entier.
-b-	63	Compiler Code Generation Allocate enums and ints (non cochée)	enums à la taille d'octet si possible.
-C	65	Compiler Source Nested Comments	Active les commentaires imbriqués.
■ -C-	65	Compiler Source Nested Comments (non cochée)	Désactive les commentaires imbriqués.
-c	90	Project Compile	Compile en .OBJ, sans édition de liens.
-Dnom	62	Compiler Defines	Attribue une chaîne constituée du caractère nul comme <i>nom</i> .
-Dnom=chaîne	62	Compiler Defines	Attribue la <i>chaîne</i> au <i>nom</i> .
-d	63	Compiler Code Generation Duplicate strings merged	Fusionne les chaînes dupliquées.
■ -d-	63	Compiler Code Generation Duplicate strings merged	Ne fusionne pas les chaînes dupliquées.
-dc	71	16-bit Compiler Memory Model Put strings in code segments	Transfère les constantes du segment de données vers le segment de code (16 bits uniquement).
-Efichier	91	<aucune>	Utilise l'assembleur spécifié.
-efichier	91	<Editez les attributs de noeud dans le gestionnaire de projet>	Effectue l'édition de liens pour produire le <i>fichier</i> spécifié.
-Fc	65	Compiler Compiler Output Generate COMDEFs	Génère des COMDEF (C 16 bits uniquement).
-Ff	65	16-bit Compiler Memory Model Automatic far data	Crée automatiquement des variables far (16 bits uniquement).
-Ff=taille	65	16-bit Compiler Memory Model Far Data Threshold	Crée automatiquement des variables far, sélectionne "taille" comme limite (16 bits uniquement).

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
-Fm	65	<aucune>	Active les options -Fc , -Ff et -Fs (16 bits uniquement).
-Fs	65	<TargetExpert Alternate startup>	Suppose que DS = SS dans tous les modèles de mémoire (DOS 16 bits uniquement).
■ -f	64	Compiler Floating point No floating point (non cochée)	Permet l'utilisation de la virgule flottante.
-f-	64	Compiler Floating point No floating point	Ne permet pas l'utilisation de la virgule flottante.
■ -ff	64	Compiler Floating point Fast floating point	Calculs flottants rapides.
-ff-	64	Compiler Floating point Fast floating point (non cochée)	Calculs en virgule flottante stricts ANSI.
-f87	91	<aucune>	Utilise les instructions matérielles 8087.
-f287	91	Compiler Floating point Fast floating point	Utilise les instructions matérielles 80287 (pour applications DOS).
-G	81	Optimizations Specific Executable Speed	Code le plus rapide.
■ -G-	81	Optimizations Specific Executable Size	Code le plus compact.
■ -gn	87	Messages Stop after n warnings	Avertissements : interruption au bout de n messages (100 par défaut).
-H	68	Compiler Precompiled headers Generate and use	Génère et utilise des en-têtes précompilés.
■ -H-	68	Compiler Precompiled headers Do not generate or use	N'utilise pas d'en-têtes précompilés et ne les génère pas.
-Hc	91	<aucune>	En-têtes précompilés de cache. Cette option doit être utilisée avec -H or -Hxxx .
-Hu	68	Compiler Precompiled headers Use but don't generate	Utilise des en-têtes précompilés, mais ne les génère pas.
-H"xxx"	68	Compiler Precompiled headers Stop precompiling after header	Interrompt la compilation des en-têtes précompilés au fichier "xxx" (32 bits uniquement). Cette option doit être utilisée avec -H , -Hu ou -H=fichier .
-H=fichier	68	Compiler Precompiled headers precompiled header file name	Définit le nom du fichier contenant les en-têtes précompilés.
-h	72	16-bit Compiler Memory Model Fast huge pointers	Utilise l'arithmétique fast huge pointer (16 bits uniquement).
-lchemin	60	Directories Include	Définit le chemin d'accès aux répertoires pour les fichiers include.
■ -in	65	Compiler Source Identifier length	Sélectionne la longueur significative d'identificateur n (par défaut 32).

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
■ -Jg	80	C++ Options Templates Smart	Génère des définitions pour toutes les instances de modèles et fusionne les duplications.
-Jgd	80	C++ Options Templates Global	Génère des définitions publiques pour toutes les instances de modèles ; duplique les résultats dans les erreurs de redéfinition.
-Jgx	80	C++ Options Templates External	Génère des références externes pour toutes les instances de modèles.
■ -jn	87	Messages Stop after n errors	Erreurs : interruption après n messages (25 par défaut).
-K	63	Compiler Code Generation Unsigned characters	Type de caractère non signé par défaut.
■ -K-	63	Compiler Code Generation Unsigned characters (non cochée)	Type de caractère signé par défaut.
-K2	78	C++ Options C++ Compatibility Don't treat char as distinct	Autorise uniquement deux types de caractères (signés et non signés) ; char est traité comme caractère signé (16 bits uniquement). Compatibilité avec Borland C++ 3.1 et versions plus anciennes.
■ -k	66	Compiler Debugging Standard stack frame	Active le cadre de pile standard.
-L <i>chemin</i>	60	Directories Library	Définit le chemin d'accès des fichiers bibliothèques.
-lx	91	<options de l'éditeur de liens>	Passé l'option x à l'éditeur de liens (peut utiliser plus d'un x).
-l-x	91	<options de l'éditeur de liens>	Désactive l'option x pour l'éditeur de liens.
-M	91	Linker Map File Segment, Public, or Detailed	Demande à l'éditeur de liens de créer un fichier mappe.
-mc	70	16-bit Compiler Memory Model Compact	Compile au moyen du modèle de mémoire COMPACT (16 bits uniquement).
-mh	70	16-bit Compiler Memory Model Huge	Compile au moyen du modèle de mémoire HUGE (16 bits uniquement).
-ml	70	16-bit Compiler Memory Model Large	Compile au moyen du modèle de mémoire LARGE (16 bits uniquement).
-mm	70	16-bit Compiler Memory Model Medium	Compile au moyen du modèle de mémoire MEDIUM (16 bits uniquement).
-mm!	70	16-bit Compiler Memory Model Medium and Never	Compile au moyen du modèle de mémoire MEDIUM ; suppose que DS != SS (16 bits uniquement).

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Équivalent dans l'EDI	Description
■ -ms	70	16-bit Compiler Memory Model Small	Compile au moyen du modèle de mémoire SMALL (16 bits uniquement).
-ms!	70	16-bit Compiler Memory Model Small and Never	Compile au moyen du modèle de mémoire SMALL ; suppose que DS != SS (16 bits uniquement).
-mt	70	16-bit Compiler Memory Model Tiny	Compile au moyen du modèle de mémoire TINY (16 bits uniquement).
-mt!	70	16-bit Compiler Memory Model Tiny and Never	Compile au moyen du modèle de mémoire TINY ; suppose que DS != SS (16 bits uniquement).
-N	66	Compiler Debugging Test stack overflow	Vérifie la présence éventuelle d'un débordement de pile.
-nchemin	60	Directories Final	Définit le répertoire de sortie.
-O	83	Optimizations Size Jump optimizations	Optimise les branchements.
-O1	81	Optimizations Specific Executable size	Génère le code le plus compact.
-O2	81	Optimizations Specific Executable speed	Génère le code le plus rapide (identique à -Ox).
-Oa	82	Optimizations Specific Assume no pointer aliasing	Optimise en supposant qu'il n'existe pas d'alias d'expressions de pointeurs à l'évaluation des sous-expressions communes.
-Ob	83	Optimizations Size Dead code elimination	Élimine le code superflu.
-Oc	82	Optimizations Specific Optimize locally	Élimine les expressions dupliquées dans les blocs de base.
-Od	81	Optimizations Disable all optimizations	Désactive toutes les optimisations.
-Oe	84	Optimizations Size Global register allocation	Attribue des registres globaux et analyse les zones de variables actives.
-Og	82	Optimizations Specific Optimize globally	Élimine les expressions dupliquées dans les fonctions.
-Oi	84	Optimizations Speed Inline intrinsic functions	Étend les fonctions intrinsèques communes en ligne.
-Ol	83	Optimizations Size Loop optimization	Compacte les boucles.
-Om	85	Optimizations Speed Invariant code motion	Extrait le code invariable des boucles.
-Op	85	Optimizations Speed Copy propagation	Propage les copies.
-Os	81	Optimizations Specific Executable size	Génère le code le plus compact.
-Ot	81	Optimizations Specific Executable speed	Génère le code le plus rapide (identique à -Oxt).
-Ov	86	Optimizations Speed Induction variables	Permet l'induction de variables de boucle et la réduction de puissance.
-OW	83	Optimizations Size Windows prolog/epilog	Supprime inc bp/dec bp dans les fonctions far de Windows (16 bits uniquement).

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
-Ox	81	Optimizations Specific Speed	Génère le code le plus rapide ; compatible Microsoft.
-ofichier	92	<aucune>	Compile le fichier source pour produire un fichier.OBJ.
-P	92	<utilisez Tools>	Effectue une compilation C++, quelle que soit l'extension du fichier source.
-Pext	92	<utilisez Tools>	Effectue une compilation C++ et définit l'extension par défaut ext.
■ -P-	92	<utilisez Tools>	Effectue une compilation C++ ou C selon l'extension du fichier source.
-P-ext	92	<utilisez Tools>	Effectue une compilation C++ ou C selon l'extension ; définit l'extension par défaut ext.
-p	69	16-bit Compiler Calling Convention Pascal	Utilise la convention d'appel Pascal dans les applications 16 bits (option BCC).
-p	69	32-bit Compiler Calling Convention Pascal	Utilise la convention d'appel Pascal dans les applications 32 bits (option BCC32).
■ ₁₆ -p- -pc	69	16-bit Compiler Calling Convention C	Utilise la convention d'appel C (option BCC).
■ ₃₂ -p- -pc	69	32-bit Compiler Calling Convention C	Utilise la convention d'appel C (option BCC32).
-po	63	Compiler Code Generation FastThis	Utilise la convention d'appel fastthis pour transmettre le paramètre this aux registres (16 bits uniquement).
-pr	69	16-bit Compiler Calling Convention Register	Utilise la convention d'appel fastcall pour transmettre les paramètres aux registres (option BCC).
-pr	69	32-bit Compiler Calling Convention Register	Utilise la convention d'appel fastcall pour transmettre les paramètres aux registres (option BCC32).
-ps	69	32-bit Compiler Calling Convention Standard call	Utilise la convention d'appel stdcall (32 bits uniquement).
■ -r	64	Compiler Code Generation Automatic	Utilise les variables de registre.
-r-	64	Compiler Code Generation None	Désactive l'utilisation des variables de registre.
-rd	64	Compiler Code Generation Register keyword	Seules les variables de registre déclarées peuvent être conservées dans les registres.
-R	67	Compiler Debugging Browser reference information in OBJs	Inclut les informations de browser dans les fichiers .OBJ générés.
■ -RT	81	C++ options Exception handling/RTTI Enable RTTI	Active les informations de type au moment de l'exécution.

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
-S	92	<TurboMenu du gestionnaire de projet>	Produit un fichier de sortie .ASM.
-T <i>chaîne</i>	90	<utilisez Tools>	Transmet la chaîne sous forme d'option à TASM, à TASM32 ou à l'assembleur spécifié par -E .
-T-	90	<utilisez Tools>	Supprime toutes les options précédentes de l'assembleur.
-tDc	92	<TargetExpert>	La cible est un fichier DOS .COM.
■ 16 -tDe	92	<TargetExpert>	La cible est un fichier DOS .EXE.
-tD	92	<TargetExpert>	La cible est un fichier DOS .EXE.
-tW	74	16-bit Compiler Entry/Exit Windows all functions exportable	La cible est un fichier Windows .EXE, toutes fonctions exportables.
-tWC	92	<TargetExpert>	La cible est un fichier console .EXE, toutes fonctions exportables.
-tWCD	93	<TargetExpert>	La cible est un fichier console .DLL, toutes fonctions exportables.
-tWD	74	16-bit Compiler Entry/Exit Windows DLL, all funcs exported	La cible est un fichier Windows .DLL, toutes fonctions exportables.
-tWDE	74	16-bit Compiler Entry/Exit Windows DLL, explicit exported	La cible est un fichier Windows .DLL, fonctions explicites exportables.
-tWCDE	74	<TargetExpert>	La cible est un fichier console .DLL, fonctions explicites exportables.
-tWE	74	16-bit Compiler Entry/Exit Windows explicit funcs exported	La cible est un fichier Windows .EXE, fonctions explicites exportables.
-tWM	93	<TargetExpert>	La cible est multithread.
-tWS	74	16-bit Compiler Entry/Exit Windows smart callbacks, all funcs	La cible est un fichier Windows .EXE utilisant des rappels smart (16 bits uniquement).
-tWSE	74	16-bit Compiler Entry/Exit Windows smart callbacks, explicit	La cible est un fichier Windows .EXE utilisant des rappels smart, fonctions explicites exportables (16 bits uniquement).
-U <i>nom</i>	93	<utilisez LocalOverride dans le projet>	Annule les définitions précédentes du nom.
■ -u	65	Compiler Compiler Output Generate underscores	Génère des caractères de soulignement.
-v, -v-	67	Compiler Debugging Debug information in OBJs	Active le débogage de source.

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
-vi,-vi-	67	Compiler Debugging Out-of-line inline functions	Contrôle l'expansion des fonctions en ligne.
-V0	79	C++ Options Virtual Tables External	Tables virtuelles externes C++.
-V1	79	C++ Options Virtual Tables Public	Tables virtuelles publiques C++.
■ -V	79	C++ Options Virtual Tables Smart	Utilise les tables virtuelles smart C++.
-Va	78	C++ Options C++ Comp Pass class values via reference to temp	Transmet les arguments de classe par référence à une variable temporaire.
-Vb	78	C++ Options C++ Compatibility Same size as 'this' pointer	Donne au pointeur de classe virtuelle de base la même taille que le pointeur de classe (16 bits uniquement).
■ 16 -Vb-	78	C++ Options C++ Compatibility Always near	Pointeur de classe virtuelle de base de type near (16 bits uniquement).
-Vc	78	C++ Options C++ Compatibility Disable constructor displacement	Ne modifie pas l'agencement des classes pour limiter les restrictions sur les pointeurs membres (16 bits uniquement).
-Vf	71	16-bit Compiler Memory Model Far virtual tables	Tables virtuelles far C++ (16 bits uniquement).
-Vmd	78	C++ Options Member Pointer Smallest for class	Utilise la plus petite représentation pour les pointeurs membres.
-Vmm	78	C++ Options Member Pointer Support multiple inheritance	Les pointeurs membres supportent l'héritage multiple.
-Vmp	77	C++ Options Member Pointer Honor precision of member pointers	Respecte la précision déclarée pour tous les types de pointeurs membres.
-Vms	78	C++ Options Member Pointer Support single inheritance	Les pointeurs membres supportent l'héritage individuel.
■ -Vmv	77	C++ Options Member Pointer Support all cases	Pas de restriction sur les pointeurs membres (représentation la plus générale).
-Vo	93	<aucune>	Active toutes les options de compatibilité ascendante (-Va, -Vb, -Vc, -Vp, -Vt, -Vv).
-Vp	78	C++ Options C++ Compatibility Push "this" first for pascal	Transmet le paramètre "this" comme premier paramètre de la pile aux fonctions membres "pascal".
-Vs	79	C++ Options Virtual Tables Local	Tables virtuelles locales C++.
-Vt	79	C++ Options C++ Comp Vtable Pointer follows data members	Place le pointeur de table virtuelle après les membres de données non statiques.

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
-Vv	79	C++ Options C++ Compatibility 'deep' virtual bases	N'ajoute pas de membres cachés et de code aux classes comportant des pointeurs vers les membres de classe virtuelle de base.
-W	74	16-bit Comp Entry/Exit Win all functions exportable	La cible est un fichier Windows .EXE , toutes fonctions exportables.
-W	92	<TargetExpert>	Crée une application GUI 32 bits lorsqu'elle est utilisée avec BCC32.
-WD	74	16-bit Comp Entry/Exit Win DLL, all functions exportable	La cible est un fichier Windows .DLL, toutes fonctions exportables.
-WDE	74	16-bit Comp Entry/Exit Win DLL, explicit functions exported	La cible est un fichier Windows .DLL, fonctions explicites exportables.
-WE	74	16-bit Comp Entry/Exit Windows explicit functions exported	La cible est un fichier Windows .EXE, fonctions explicites exportables.
-WM	93	<TargetExpert ; cochez multithread>	La cible est un fichier multithread Windows .EXE , toutes fonctions exportables.
-WS	74	16-bit Entry/Exit Win smart callbacks, all functions exported	La cible est un fichier Windows .EXE qui utilise des rappels smart, toutes fonctions exportables (16 bits uniquement).
-WSE	74	16-bit Entry/Exit Win smart callbacks, explicit functions exported	La cible est un fichier Windows .EXE qui utilise des rappels smart, fonctions explicites exportables (16 bits uniquement).
-w!	90	Make Break Make on warnings	Produit un code de retour différent de zéro à partir du compilateur en ligne lorsqu'il existe des avertissements et ne compile pas en fichier .OBJ.
■ -w	87	Messages All	Active l'affichage d'avertissements.
-w-	87	Messages None	N'affiche pas les avertissements.
-wxxx	87	Messages Selected (voir avertissement spécifique)	Active le message d'avertissement xxx.
-w-xxx	87	Messages Selected (voir avertissement spécifique)	Désactive le message d'avertissement xxx.
-X	65	Compiler Compiler Output Autodependency information (non cochée)	N'utilise pas la sortie d'auto-dépendance du compilateur.
■ -X-	65	Compiler Compiler Output Autodependency information	Utilise la sortie d'auto-dépendance du compilateur.
■ -x	80	C++ Options Exception handling Enable exceptions	Active la gestion d'exceptions.

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Équivalent dans l'EDI	Description
■ -xd	80	C++ Options Exception handling Enable destructor cleanup	Active la mise au net des destructeurs.
-xp	80	C++ Options Exception handling Enable exception location info	Active les informations de localisation d'exception.
-Y	93	<TargetExpert DOS Overlay>	Active la génération de code de recouvrement.
-Yo	67	<éditez les attributs de noeud et cochez Overlay this module>	Recouvre les fichiers compilés
-y	67	Compiler Debugging Line numbers	Numéros de ligne activés.
-Z	83	Optimizations Size Suppress redundant loads	Active le chargement de registre.
-zAnom	74	16-bit Compiler Segment Names Code Code Class	Classe de code.
-zBnom	73	16-bit Compiler Segment Names Data Uninitialized Data Class	Classe BSS.
-zCnom	74	16-bit Compiler Segment Names Code Code Segment	Segment de code.
-zDnom	73	16-bit Compiler Segment Names Data Uninitialized Data Segment	Segment BSS.
-zEnom	73	16-bit Compiler Segment Names Far Data Segment	Segment far (16 bits uniquement).
-zFnom	73	16-bit Compiler Segment Names Far Data Class	Classe far (16 bits uniquement).
-zGnom	73	16-bit Compiler Segment Names Data Uninitialized Data Group	Groupe BSS.
-zHnom	73	16-bit Compiler Segment Names Far Data Group	Groupe far (16 bits uniquement).
-zPnom	74	16-bit Compiler Segment Names Code Code Group	Groupe de code.
-zRnom	72	16-bit Compiler Segment Names Data Initialized Segment	Segment de données.
-zSnom	73	16-bit Compiler Segment Names Data Initialized Group	Groupe de données.
-zTnom	73	16-bit Compiler Segment Names Data Initialized Class	Classe de données.
-zVnom	73	16-bit Compiler Segment Names Far Data VT Segment	Segment virtuel far (16 bits uniquement).

Tableau 3.1 : Récapitulatif des options (suite)

Option	Page	Equivalent dans l'EDI	Description
-zWnom	73	16-bit Compiler Segment Names Far Data VT Class	Classe virtuelle far (16 bits uniquement).
■ -zX*	74	<aucune>	Utilise le nom par défaut pour X ; X a pour valeur A-H, P, R, S, T, V ou W.

■ Par défaut en 16 bits et 32 bits ■ 16 Par défaut en 16 bits ■ 32 Par défaut en 32 bits

Répertoires

-lchemin L'option **Include** recherche les fichiers include dans les répertoires du *chemin d'accès* (lettre d'unité ou chemin d'accès d'un sous-répertoire), ainsi que dans les répertoires standard. La lettre d'unité est une lettre majuscule ou minuscule suivie du signe deux-points (:). Le répertoire est un répertoire ou un chemin d'accès valide. Vous pouvez utiliser plus d'une option de répertoire **-I** (I majuscule).

-Lchemin L'option **Library** oblige l'éditeur de liens à utiliser le fichier objet de démarrage C0x.OBJ et les fichiers de bibliothèque Borland C++ situés dans le répertoire spécifié. Par défaut, l'éditeur de liens recherche ces fichiers dans le répertoire courant.

Source désigne le répertoire dans lequel le compilateur recherche le code source.

Intermédiaire est le répertoire dans lequel le compilateur place les fichiers temporaires éventuellement créés.

-nchemin L'option **Final** place les fichiers de sortie finalisés (.OBJ, .I ou .ASM) que le compilateur a créé dans le répertoire ou l'unité désignée par le *chemin d'accès*.

Vous pouvez spécifier plusieurs répertoires en ligne de commande selon une des méthodes suivantes :

- Vous pouvez spécifier plusieurs entrées à la suite d'une option -L ou -I individuelle sur la ligne de commande en les séparant par un signe deux-points :

```
BCC.EXE -Lnomrép1;nomrép2;nomrép3 -Iincl;inc2;inc3 monfich.c
```

- Vous pouvez entrer plus d'une option sur la ligne de commande, comme suit :

```
BCC.EXE -Lnomrép1 -Lnomrép2 -Lnomrép3 -Iinc1 -Iinc2 -Iinc3
monfich.c
```

Vous pouvez combiner les types de listes :

```
BCC.EXE -Lnomrép1;nomrép2 -Lnomrép3 -Iinc1;inc2 -Iinc3
monfich.library
```

Si vous définissez plusieurs options **-L** ou **-I** sur la ligne de commande, le résultat est cumulé : le compilateur consulte tous les répertoires spécifiés, de gauche à droite. L'EDI supporte également plusieurs répertoires de bibliothèque.

Algorithmes de recherche de fichiers

Les algorithmes de recherche de fichiers include Borland C++ recherchent les fichiers d'en-tête spécifiés dans le code source en procédant de la manière suivante :

- Si vous entrez une instruction `#include <fichier.h>` dans le code source, Borland C++ ne recherche fichier.h que dans les répertoires include spécifiés.
- Par contre, si vous entrez une instruction `#include <fichier.h>` dans le code, Borland C++ recherche tout d'abord fichier.h dans le répertoire courant ; s'il n'y figure pas, Borland C++ le recherche ensuite dans les répertoires include spécifiés en ligne de commande.

Les algorithmes de recherche de fichiers de bibliothèque sont similaires à ceux des fichiers include :

- Bibliothèques implicites : Borland C++ recherche uniquement les bibliothèques implicites dans les répertoires de bibliothèque spécifiés, comme l'algorithme de recherche **#include "fichier.h"**. Les fichiers de bibliothèque implicites sont ceux qui sont automatiquement liés par Borland C++, ainsi que le fichier objet de démarrage (C0x.OBJ).
- bibliothèques explicites : les répertoires dans lesquels Borland C++ recherche les bibliothèques explicites (utilisateurs) dépendent de la façon dont vous avez spécifié le nom de fichier de bibliothèque. Les fichiers de bibliothèque explicites sont ceux que vous spécifiez en ligne de commande ou dans un fichier projet ; leur nom est suivi de l'extension .LIB.

- Si vous spécifiez un nom de fichier de bibliothèque explicite, sans préciser l'unité ni le répertoire dans lequel il se trouve (ex. `mylib.lib`), Borland C++ recherche tout d'abord la bibliothèque en question dans le répertoire courant. Si elle ne s'y trouve pas, Borland C++ la recherche dans les répertoires de bibliothèque spécifiés. Cet algorithme est donc similaire à l'algorithme de recherche utilisé pour **#include** "*fichier.h*".
- Si vous spécifiez une bibliothèque utilisateur en précisant l'unité et/ou le répertoire dans lequel elle figure (ex. `:c:perso\mabiblio1.lib`), Borland C++ recherche uniquement la bibliothèque dans le répertoire désigné par le chemin d'accès, et non pas dans les répertoires de bibliothèque spécifiés.

Compiler | Defines

Les définitions de macros vous permettent de définir et d'annuler des macros en ligne de commande ou dans l'EDI. Les macros définies en ligne de commande ou dans la boîte de dialogue Options Setting ont priorité sur celles qui sont définies dans le fichier source. Entrez les définitions de macros dans la boîte Defines au-dessous du sujet Code Generation | Settings.

`-Dnom` Affecte une chaîne vide au *nom* d'identificateur spécifié. Séparez les macros par un point-virgule.

`-Dnom=chaîne` Affecte la *chaîne* qui suit le signe égal au *nom* d'identificateur spécifié. Les chaînes ne doivent pas contenir d'espaces, ni de tabulation. Séparez les macros par un point-virgule.

Borland C++ vous permet de spécifier plusieurs entrées **#define** en ligne de commande en procédant selon une des méthodes suivantes :

- Vous pouvez préciser plusieurs entrées à la suite d'une option **-D** individuelle en les séparant les unes des autres par un point-virgule :

```
BCC.EXE -Dxxx;yyy=1;zzz=NO MONFICH.C
```

- Vous pouvez entrer plusieurs options **-D** en les séparant par des espaces :

```
BCC.EXE -Dxxx -Dyyy=1 -Dzzz=NO MONFICH.C
```

- Vous pouvez combiner plusieurs listes **-D** contenant des entrées séparées par des points-virgules :

```
BCC.EXE -Dxxx -Dyyy=1;zzz=NO MONFICH.C
```


- b -b L'option **Allocate enums as ints** attribue un **int** de deux octets (16 bits) ou de quatre octets (32 bits) aux types d'énumération. Elle est activée par défaut. Si elle n'est pas cochée (-b-), elle attribue la plus petite taille de variable en mesure de contenir les valeurs d'énumération : le compilateur attribue un char **non signé** ou **signé** si les valeurs d'énumération sont comprises dans l'éventail de 0 à 255 (minimum) ou de -128 à 127 (maximum), ou un entier court **non signé** ou **signé** si les valeurs d'énumération sont comprises dans l'éventail 0 à 65 535 (minimum) ou -32 768 à 32 767 (maximum). Le compilateur attribue un **int** de deux octets (16 bits) ou de quatre octets (32 bits) pour représenter les valeurs d'énumération si l'une d'entre elles se situe en dehors de l'éventail de valeurs.
- K -K L'option **Unsigned characters** traite toutes les déclarations **char** comme s'il s'agissait de caractères **non signés** afin d'assurer la compatibilité avec les autres compilateurs. Par défaut, BBC utilise des déclarations **char** signées (option non cochée ou **-K-**).
- d L'option **Duplicate strings merged** fusionne les chaînes littérales lorsqu'une chaîne est identique à une autre pour réduire la taille des programmes. Elle augmente toutefois la durée de compilation. Cette option risque de produire des erreurs lorsqu'une chaîne est modifiée. Par défaut elle n'est pas activée (**-d-**).
- po **FastThis** utilise la convention d'appel `_ _fastthis` pour passer le pointeur **this** dans un registre aux fonctions membres.

Lorsque **fastthis** est actif (applications 16 bits uniquement dans la mesure où **fastthis** est toujours activé dans les applications 32 bits), le compilateur compile les fonctions membres de façon à ce que leur pointeur **this** soit transmis dans un registre (ou dans une paire de registres pour les modèles de données 16 bits plus importants). De même, les appels de fonctions membres chargent la valeur de **this** (dans le registre) ou la paire de registres.

Vous pouvez activer **fastthis** au moyen de l'option de ligne de commande **-po** ou de l'option d'appel Compiler | Code Generation | FastThis. Vous pouvez également utiliser le mot-clé de spécification de langage `_ _fastthis`.

Dans les modèles de données SMALL ou FLAT, **this** est fourni dans le registre SI ; les modèles de données 16 bits plus importants utilisent DS:SI. Si nécessaire, le compilateur sauvegarde et restaure DS. Toutes les références aux données membre dans la fonction membre sont effectuées par l'intermédiaire du registre SI.

Le nom des fonctions membres compilées avec **fastthis** est transformé (mangled) différemment des fonctions membres **non-fastthis** pour éviter de mélanger les deux. Il est plus facile de compiler toutes les classes avec **fastthis**, mais vous pouvez très bien compiler certaines classes avec **fastthis** et d'autres sans.

- r L'option **None** n'utilise pas de variables de registre.
- rd L'option **Register keyword** spécifie que les variables de **registre** ne sont utilisées que si vous utilisez le mot-clé de registre et qu'un registre est disponible. Vous pouvez utiliser **-rd** dans les **options #pragma**. Utilisez cette option ou l'option -r pour optimiser l'utilisation des registres.
- r L'option Automatic utilise les variables de registre. Le compilateur affecte automatiquement des variables aux registres, même lorsque vous ne spécifiez pas de variable de registre au moyen d'un spécificateur de type de registre. L'option **-r** est activée par défaut.

Compiler|Floating Point

- f-f L'option **No floating point (-f-)** spécifie que le programme ne contient aucun calcul en virgule flottante ; aucune bibliothèque en virgule flottante n'est donc liée. Si cette option n'est pas cochée (**-f**), elle émule les appels 80x87 au moment de l'exécution.
- ff -ff L'option **Fast floating point (-ff)** optimise les opérations en virgule flottante sans tenir compte des conversions de type implicite ou explicite. Le temps de réponse est parfois plus rapide qu'en mode d'exploitation ANSI. Non cochée (**-ff-**), cette option désactive les calculs flottants rapides. Le compilateur suit les règles ANSI strictes pour la conversion des calculs en virgule flottante.

Compiler | Compiler Output

- X -X- **Autodependency information (-X)** génère des informations d'auto-dépendance. Les modules compilés lorsque cette option est active peuvent utiliser la fonction d'auto-dépendance de MAKE. Par défaut, l'auto-dépendance est activée (-X-).
- u L'option **Generate underscores** insère automatiquement un caractère de soulignement devant les identificateurs avant de les sauvegarder dans un module objet. Les caractères de soulignement des identificateurs C et C++ sont optionnels, mais sont activés par défaut. Pour les désactiver, utilisez l'option -u-. A noter cependant que lorsque les caractères de soulignement sont désactivés, des erreurs de liaison se produisent au cours de la liaison des bibliothèques Borland C++ standard. Pour plus de détails sur les caractères de soulignement, consultez le chapitre 10 du *Guide du programmeur*.
- Fc **Generate COMDEFs** (16 bits uniquement) génère des variables communes (COMDEF) pour les variables C globales qui ne sont pas initialisées, ni déclarées **statiques** ou **externes**. L'avantage de cette option est que les fichiers d'en-tête qui sont inclus dans plusieurs fichiers sources peuvent contenir des déclarations de variables globales. Tant qu'une variable donnée ne doit pas être initialisée pour prendre une valeur différente de zéro, vous n'avez pas à inclure de définition correspondante dans les fichiers sources. Vous pouvez utiliser cette option pour porter du code utilisant une fonction similaire dans une autre implémentation.

Compiler | Source

- C **Nested comments** vous permet d'imbriquer des commentaires. Les commentaires ne peuvent normalement pas être imbriqués.
- in **Identifiant length** oblige le compilateur à reconnaître uniquement les n premiers caractères des noms d'identificateur. Tous les identificateurs, qu'il s'agisse de variables, de macros préprocesseurs ou de membres de structure, ne sont traités en tant qu'identificateurs distincts que si les n premiers caractères sont uniques. Si vous donnez à n la valeur 0 ou une valeur supérieure à 249, ou si vous ne spécifiez pas l'option **-in**, les identificateurs ont une longueur illimitée.

Par défaut, Borland C++ utilise 32 caractères par identificateur. Les autres systèmes, y compris certains compilateurs UNIX, ignorent les caractères au-delà des huit premiers. Si le code doit être porté dans d'autres environnements, il est préférable de le compiler en limitant le nombre de caractères significatifs pour vous aider à localiser les conflits de noms dans les identificateurs relativement longs qui ont été tronqués.

- A -AT **Borland extensions** utilise les mots-clés Borland C++. Pour obtenir la liste complète des mots clés Borland C++, reportez-vous au chapitre 1 du Guide du programmeur.
- A **ANSI** compile du code compatible ANSI. Les mots-clés Borland C++ sont ignorés et peuvent être utilisés comme identificateurs normaux.
- AU **UNIX V** utilise uniquement la conformité d'extension de langage UNIX.
- AK **Kernighan and Ritchie** utilise uniquement la conformité de langage Kernighan et Ritchie.

Compiler | Debugging

- k **Standard stack frame** génère un cadre de pile standard qui est utile lorsqu'un débogueur est utilisé pour remonter la pile de procédures invoquées. Cette option est active par défaut. Lorsqu'elle est désactivée, toute fonction qui n'utilise pas de variable locale et ne possède pas de paramètre est compilée avec une entrée et un code de retour abrégés, rendant ainsi le code moins volumineux et plus rapide.
- N **Test stack overflow** génère une logique de débordement de pile à l'entrée de chaque fonction. Cette option affiche un message de débordement de pile en présence d'un débordement de pile à l'exécution. Elle affecte à la fois la taille et la vitesse du programme, mais a été prévue parce qu'il est parfois très difficile de détecter un débordement de pile. Lorsqu'un débordement de pile est détecté, le message Stack overflow! apparaît et le programme se termine en produisant le code de sortie 1.

-vi **Out-of-line inline functions** étend les fonctions **en ligne** C++. Pour contrôler l'expansion des fonctions en ligne, l'option **-v** agit de manière légèrement différente en C++ : lorsque l'expansion de fonction en ligne n'est pas activée, la fonction est générée et appelée comme toute autre fonction. Le débogage est parfois difficile avec l'expansion en ligne, c'est pourquoi Borland C++ fournit les options suivantes :

- **-v** active le débogage et désactive l'expansion en ligne. Lorsque l'option est désactivée, vous pouvez lier des fichiers .OBJ plus volumineux. Cette option affecte la durée de compilation, mais pas la vitesse d'exécution.
- **-v-** désactive le débogage et active l'expansion en ligne.
- **-vi** active l'expansion en ligne.
- **-vi-** désactive l'expansion en ligne.

Par exemple, si vous voulez activer le débogage et l'expansion en ligne, vous devez utiliser **-v -vi**.

-y **Line numbers** ajoute des numéros de ligne dans le fichier .OBJ pour le débogueur intégré de l'EDI. Les numéros augmentent la taille du fichier .OBJ, mais n'affectent pas la taille, ni la vitesse du programme exécutable. Cette option est utile avec les débogueurs symboliques. En général, **-v** est plus utile que **-y** avec le débogueur intégré.

Les options de débogage comprennent les informations de débogage dans le code généré. Pour plus de détails sur le débogage d'application, consultez le chapitre 6 ; pour plus de détails sur les informations de browser, reportez-vous au chapitre 1

-v **Debug information in OBJs** ajoute des informations de débogage aux fichiers .OBJ pour permettre de les déboguer au moyen du débogueur intégré ou d'un débogueur autonome. Le compilateur transmet cette option à l'éditeur de liens pour lui permettre d'inclure les informations de débogage dans le fichier .EXE. Dans le cadre du débogage, cette option traite les fonctions en ligne C++ comme des fonctions normales.

-R **Browser reference information in OBJs** ajoute des informations de browser lorsque le compilateur génère les fichiers .OBJ ; ceci vous permet d'examiner une application au moyen du Browser intégré de l'EDI. Le Browser est décrit au chapitre usingbc. Lorsque cette option est désactivée, vous pouvez lier des fichiers .OBJ plus volumineux. Cette option affecte la durée de compilation, mais pas la vitesse d'exécution.

Compiler | Precompiled headers

- H **Generate and use** génère et utilise des en-têtes précompilés au moyen du fichier par défaut BCDEF.CSM (16 bits) ou BC32DEF.CSM (32 bits) pour les compilateurs en ligne de commande et <nom_projet>.CSM pour les projets dans l'EDI. Les en-têtes précompilés peuvent accélérer sensiblement la vitesse de compilation, mais occupent un espace considérable sur le disque. Pour plus de détails sur les en-têtes précompilés, voir l'annexe C.
- Hu **Use but do not generate** utilise les en-têtes précompilés mais ne les génère pas.
- H **Do not generate or use** ne génère pas d'en-têtes précompilés et ne les utilise pas.
- H=fichier **Precompiled header name** génère et utilise des en-têtes précompilés et définit le nom du fichier contenant les en-têtes précompilés (différent de BCDEF.CSM ou BC32DEF.CSM).
- H"xxx" **Stop precompiling after header file** interrompt la compilation d'en-têtes précompilés lorsqu'elle compile le fichier spécifié par xxx.

16-bit Compiler | Processor

- 2 **80286** génère des instructions compatibles avec le mode protégé 80286 16 bits.
- 3 **80386** génère des instructions compatibles avec le mode protégé 80386 16 bits.
- 4 **i486** génère des instructions compatibles avec le module protégé 80486 16 bits.
- an **Data alignment Byte/Word** aligne sur un octet (**Byte**, $n = 1$) ou sur un mot (**Word**, $n = 2$). Voir plus loin la version 32 bits de cette option.

Word (-a) oblige à aligner les éléments à la taille d'entier et plus importants sur une limite de mot machine. Les octets supplémentaires sont insérés dans une structure pour garantir l'alignement correct des membres. Les variables automatiques et globales sont alignées correctement. Les variables char et non signées et les zones peuvent être placées à toute adresse ; toutes les autres variables sont placées à une adresse portant un numéro pair. **Byte (-a-)** permet l'alignement d'octet. L'alignement de mot augmente la vitesse d'extraction et de stockage des données des processeurs 80x86.

16-bit Compiler | Calling Convention

Les conventions d'appel font l'objet d'une description plus détaillée dans le chapitre 2 du *Guide du programmeur*.

- pc -p **C** génère tous les appels de procédures et toutes les fonctions au moyen de la convention d'appel d'appel C, ce qui revient à déclarer toutes les procédures et fonctions au moyen du mot-clé `__cdecl`. Les appels de fonctions résultants sont plus courts et plus rapides. Les fonctions doivent transmettre le nombre et le type d'arguments corrects, à la différence de l'usage normal du C qui permet un nombre variable d'arguments de fonction. Vous pouvez utiliser le mot-clé `__cdecl`, `__pascal` ou `__fastcall` pour déclarer spécifiquement une fonction ou une procédure d'appel au moyen d'une convention d'appel différente.
- p **Pascal** génère tous les appels de procédures et toutes les fonctions au moyen de la convention d'appel Pascal, ce qui revient à déclarer toutes les procédures et fonctions au moyen du mot-clé `__pascal`. Les appels de fonctions résultants sont plus courts et plus rapides. Les fonctions doivent transmettre le nombre et le types d'arguments corrects. Vous pouvez utiliser le mot-clé `__cdecl`, `__stdcall` ou `__fastcall` pour déclarer spécifiquement une fonction ou procédure au moyen d'une autre convention d'appel.
- pr **Register** génère tous les appels de procédures et toute les fonctions au moyen de la convention d'appel Register, ce qui revient à déclarer toutes les procédures et fonctions au moyen du mot-clé `__fastcall`. Vous pouvez utiliser le mot-clé `__stdcall`, `__pascal` ou `__cdecl` pour déclarer spécifiquement une fonction ou procédure au moyen d'une convention d'appel différente.

16-bit Compiler | Memory Model

Les options de modèle de mémoire vous permettent d'indiquer au compilateur le modèle de mémoire à utiliser pour compiler les applications 16 bits (les applications 32 bits utilisent automatiquement le modèle FLAT). Les modèles de mémoire disponibles sont les suivants : SMALL, MEDIUM, COMPACT et LARGE. Pour plus de détails sur les modèles de mémoire, consultez le chapitre 8 du *Guide du programmeur*.

-ms -ms! **Small** effectue la compilation au moyen du modèle de mémoire SMALL (modèle par défaut). L'option en ligne de commande **-ms!** effectue la compilation au moyen du modèle de mémoire SMALL et suppose que DS != SS. Pour obtenir le même résultat dans l'EDI, vous devez cocher deux options (Small et Never).

-mm -mm! **Medium** effectue la compilation au moyen du modèle de mémoire MEDIUM. L'option en ligne de commande **-mm!** effectue la compilation au moyen du modèle de mémoire MEDIUM et suppose que DS != SS. Pour obtenir le même résultat dans l'EDI, vous devez cocher deux options (Medium et Always).

L'effet réel des options **-mt!**, **-ms!** et **-mm!** est en fait très limité. Si vous utilisez l'adresse d'une variable de pile (paramètre ou automatique), le pointeur résultant est, par défaut (DS= SS), un pointeur near (relatif à DS). Ceci vous permet d'affecter l'adresse à un pointeur à la taille par défaut dans ces modèles sans aucun problème. Lorsque DS != SS, le type de pointeur créé lorsque vous utilisez l'adresse d'une variable de pile est un pointeur **_ss**. Vous pouvez donc affecter ou transmettre librement le pointeur à un pointeur far ou **_ss**. Cependant, l'attribution de l'adresse à un pointeur near ou à la taille par défaut produit un avertissement "Suspicious pointer conversion" dans les modèles de mémoire affectés. De tels avertissements représentent généralement une erreur et sont donc activés par défaut.

-mc **Compact** effectue la compilation au moyen du modèle de mémoire COMPACT.

-ml **Large** effectue la compilation au moyen du modèle de mémoire **LARGE**.

-mh **Huge** effectue la compilation au moyen du modèle de mémoire **HUGE**.

- mt -mt! **Tiny** effectue la compilation au moyen du modèle de mémoire **TINY**. L'option en ligne de commande **-mt** compile selon le modèle **SMALL** et suppose que $DS \neq SS$. Pour réaliser ceci dans l'EDI, vous devez cocher deux options (**Tiny** et **Never**).
Default for model détermine, en fonction du modèle, si le segment de pile est égal au segment de données.
 Par défaut, le modèle est utilisé pour déterminer si le segment de pile est égal au segment de données.
- Fs **Never** suppose que le segment de données n'est jamais égal au segment de pile, quel que soit le modèle de mémoire utilisé.
- Fs **Always** suppose que le segment de données (DS) est égal au segment de pile (SS) dans tous les modèles de mémoire ; vous pouvez l'utiliser pour porter du code développé à l'origine pour une implémentation qui intègre la pile au segment de données.
- dc **Put strings in code segments** transfère toutes les constantes chaînes du segment de données dans le segment de code du fichier objet généré pour donner un type de données **const** (16 bits uniquement). Ces options permettent de limiter l'espace occupé par le segment de données. Dans les programmes volumineux, notamment ceux qui contiennent de nombreuses constantes chaînes, cette option transfère le poids des constantes chaînes du segment de données au segment de code.
- Ff **Automatic far data** change le type des variables globales supérieures ou égales à la taille de seuil pour en faire des variables **far**. La taille de seuil par défaut est de 32 767. Cette option est utile lorsque le code n'utilise pas le modèle de mémoire **HUGE** mais déclare des variables globales dont la taille totale dépasse de 64 ko (ou s'en rapproche). Cette option n'a aucun effet pour les modules de mémoire **TINY**, **SMALL** et **MEDIUM**. Si vous utilisez cette option avec **-Fc**, les **COMDEF** générés prennent le type **far** dans les modèles **COMPACT**, **LARGE** et **HUGE**.
- Vf **Far virtual tables** crée les tables virtuelles dans le segment de code au lieu du segment de données (sauf en cas de changement au moyen des options **-zV** et **-zW**) et convertit les pointeurs de table virtuelle en pointeurs 32 bits complets (cette opération est effectuée automatiquement si vous utilisez le modèle de mémoire **HUGE**).

Cette option peut être utilisée pour deux raisons : pour supprimer les tables virtuelles du segment de données, parce qu'il arrive à saturation par exemple, et pour pouvoir partager les objets (de classes avec fonctions virtuelles) entre les modules qui utilisent différents segments de données (par exemple, une DLL et un exécutable utilisant le DD en question). Pour tous les modules en mesure de partager des objets, vous devez effectuer la compilation totalement avec ou totalement sans cette option. Vous obtenez le même effet en utilisant les modificateurs **huge** ou **_export** sur chaque classe.

- h **Fast huge pointers** offre une méthode alternative pour le calcul d'expressions pointeurs HUGE ; cette méthode est beaucoup plus rapide que la méthode standard, mais doit être utilisée avec précaution. Lorsque vous utilisez cette option, les huge pointers ne sont normalisés que lorsqu'un bouclage de segment se produit dans la partie déplacement, ce qui crée des problèmes avec les grands tableaux lorsqu'un élément du tableau franchit une limite de tableau. Cette option est désactivée par défaut.

En principe, Borland C++ normalise un huge pointer lorsqu'il ajoute ou soustrait un élément de celui-ci. Ceci permet de s'assurer que l'indexation dans un tableau et la sélection de zones struct fonctionneront avec des **structs** de toute taille si vous utilisez un grand tableau de struct supérieur à 64 ko par exemple. Borland C++ y parvient en normalisant toujours le résultat des opérations pointeurs HUGE, de sorte que la partie déplacement contienne une valeur ne dépassant pas 15 pour éviter les bouclages avec les huge pointers. L'inconvénient de cette méthode est qu'elle tend à ralentir sensiblement la vitesse d'exécution. Cette option est automatiquement sélectionnée pour la compilation Windows.

- Ff=size **Far Data Threshold** modifie le point auquel les données prennent le type far (utilisé par l'option **-Ff**).

16-bit Compiler | Segment Names Data

Ces options demandent une excellente connaissance de la segmentation sur les processeurs 80x86. En temps normal, vous n'avez pas à spécifier les noms de segment.

- zRnom **Initialized Data Segment** donne le *nom* spécifié au segment de données initialisé. Par défaut, le segment de données initialisé est nommé **_DATA**.

- zSnom **Initialized Data Group** change le nom du groupe de segments de données initialisé pour adopter le *nom* spécifié. Par défaut, le groupe de données initialisé est nommé DGROUP.
- zTnom **Initialized Data Class** donne le *nom* spécifié au segment de données initialisé. Par défaut, le nom du segment initialisé est DATA.
- zDnom **Uninitialized Data Segment** change le *nom* du segment de données non initialisé pour adopter le nom spécifié. Par défaut, le *nom* du segment de données non initialisé est _BSS.
- zGnom **Uninitialized Data Group** change le *nom* du groupe de segments de données non initialisé pour adopter le nom spécifié. Par défaut, le *nom* du groupe de données est DGROUP.
- zBnom **Uninitialized Data Class** change le *nom* de la classe de segments de données non initialisée pour adopter le *nom* spécifié. Par défaut, les segments de données non initialisés sont attribués à la classe BSS.

16-bit Compiler | Segment Names Far Data

- zEnom **Far Data Segment** change le nom du segment où sont placés les objets `__far` pour adopter le nom spécifié. Par défaut, le nom du segment est le nom du fichier source suivi de `_DATA`. Un nom commençant par un astérisque (*) indique que la chaîne par défaut doit être utilisée (16 bits uniquement).
- zHnom **Far Data Group** oblige le programme à placer les objets `__far` dans le groupe portant le nom spécifié. Par défaut, les objets `__far` ne sont pas placés dans un groupe. Un nom commençant par un astérisque (*) indique que la chaîne par défaut doit être utilisée (16 bits uniquement).
- zFnom **Far Data Class** change le nom de la classe d'objets `__far` pour adopter le nom spécifié. Par défaut, elle est nommée `FAR_DATA`. Un nom commençant par un astérisque (*) indique que la chaîne par défaut doit être utilisée (16 bits uniquement).
- zVnom **Far Virtual Tables Segment** donne le nom spécifié au segment de table virtuelle far. Par défaut, les tables virtuelles sont générées dans le segment de code (16 bits uniquement).
- zWnom **Far Virtual Tables Class** donne le nom spécifié au segment de classe de tables virtuelles far. Par défaut, les classes de tables virtuelles far sont générées dans le segment `CODE` (16 bits uniquement).

16-bit Compiler | Segment Names Code

- zCnom **Code Segment** change le nom du segment de code pour adopter le nom spécifié. Par défaut, le segment de code est nommé `_TEXT`.
- zPnom **Code Group** oblige le programme à générer des fichiers de sortie éventuels ayant un code de groupe pour le segment de code spécifié.
- zAnom **Code Class** change le nom de la classe de segment de code pour adopter le nom par défaut. Par défaut, le segment de code est affecté à la classe `CODE`.

16-bit Compiler | Entry/Exit Code

Les options Entry/Exit code spécifient le type d'application que le compilateur doit créer. Pour plus de détails sur la création d'une application Windows, consultez le chapitre 2.

- tW -W -WC- **Windows all functions** exportable crée un module objet Windows dont toutes les fonctions peuvent être exportées. Cette option permet de créer le type d'exécutable Windows le plus général, mais n'est pas nécessairement la plus efficace. Il s'agit de l'option par défaut (**-W-**). Elle génère les informations système nécessaires pour chaque fonction **far**, que la fonction en ait besoin ou pas. Elle suppose que toutes les fonctions peuvent être invoquées par le noyau Windows ou par d'autres modules.

Lorsque cette option est utilisée avec une application 16 bits, elle crée un prologue/épilogue de fonction .EXE Windows pour toutes les fonctions **far**, puis configure celles qui seront invoquées à partir d'un autre module. Pour exporter véritablement l'adresse de fonction du fichier .EXE dans une .DLL, le code comprend un appel à *MakeProcInstance()* qui transmet le pointeur résultant à la .DLL demandant l'adresse de la fonction. Pour pouvoir exporter l'adresse de fonction de la .DLL, les noms de fonction doivent être inclus dans le fichier .DEF de l'exécutable.

- tWE -WE\$I-W
(option
BCC.EXE)[W];compila
tion avec toutes les
fonctions far
exportables **Windows explicit functions exported** crée un module objet Windows dans lequel seules les fonctions déclarées comme `_export` sont exportables. Utilisez cette option si certaines des fonctions ne sont pas appelées par le noyau Windows. Windows Explicit Functions Exported fonctionne de la même manière que Windows All Functions Exportable, si ce n'est que le prologue/épilogue supplémentaire est attribué uniquement aux fonctions signalées par le mot clé `_export` (et aux méthodes des classes marquées **`_export`**).

Cette option est bien plus efficace que Windows All Functions Exportable pour les applications 16 bits dans la mesure où le temps système associé au prologue est attribué uniquement aux fonctions appelées à l'extérieur du module. Cette option exige toutefois que les fonctions et classes à exporter soient déterminées à l'avance. `MakeProcInstance()` est quand même utilisé, mais aucune manipulation de fichier `.DEF` n'est nécessaire.

`-tWS -WS` **Windows smart callbacks, all functions exportable** (16 bits uniquement) crée un module objet dont les fonctions utilisent les rappels smart et sont toutes exportées. Cette option ne doit être utilisée que si le compilateur peut supposer que `DS == SS` pour toutes les fonctions du module (ce qui est le cas dans la vaste majorité des programmes Windows et qui constitue la valeur par défaut pour les outils Borland).

Cette option crée un prologue/épilogue de fonction EXE Windows pour toutes les fonctions "far", puis configure les fonctions appelées par un autre module. Il n'est pas nécessaire d'appeler `MakeProcInstance()`, ni d'éditer de fichier `.DEF`.

`-tWSE -WSE` **Windows smart callbacks, explicit functions exportable** crée une application Windows 16 bits avec rappels smart et fonctions explicites exportées. Cette option est identique à Windows Smart Callbacks, si ce n'est que le prologue/épilogue supplémentaire n'est attribué qu'aux fonctions signalées par le mot clé `_export` (et aux méthodes de classes `_export`). Elle est donc bien plus efficace dans la mesure où le temps système associé au prologue n'est affecté qu'aux fonctions appelées à l'extérieur du module.

`-tWD -WD` **Windows DLL, all functions exportable** crée un module objet DLL dont toutes les fonctions sont exportables. Cette option crée un prologue/épilogue de fonction DLL Windows pour toutes les fonctions 'far', puis configure les fonctions appelées par un autre module. Pour exporter véritablement l'adresse de fonction du `.DLL`, les noms de fonction doivent être inclus dans le fichier `.DEF` de l'exécutable.

-tWDE -WDE **Windows DLL, explicit functions exported** crée un module objet DLL dans lequel seules les fonctions marquées `_export` sont exportables. L'option `The Windows DLL, Explicit Functions Exported` est identique à `Windows DLL, All Functions Exportable`, mais le prologue/épilogue supplémentaire est attribué uniquement aux fonctions signalées par le mot clé `_export` (et aux méthodes de classe marquées `_export`). Cette option est bien plus efficace que `Windows DLL, All Functions Exportable` dans la mesure où le temps système associé au prologue est attribué uniquement aux fonctions appelées à l'extérieur du module. Elle exige toutefois que les fonctions et classes à exporter soient déterminées à l'avance. Aucune manipulation de fichier `.DEF` n'est nécessaire.

32-bit Compiler | Processor

- 3 **80386** génère des instructions compatibles avec le mode protégé 80386 32 bits.
- 4 **i486** génère des instructions compatibles avec le mode protégé 80486 32 bits.
- 5 **Pentium** génère des instructions compatibles avec le mode protégé Pentium.
- an **Data alignment Byte/Word/Double word** définit l'alignement selon la valeur n indiquée en ligne de commande. La correspondance est la suivante **Byte** ($n=1$) aligne sur un octet ; **Word** ($n=2$), sur un mot de deux octets ; **Double word** ($n=4$), sur un double mot de quatre octets. L'alignement de mot accélère la vitesse d'extraction et de stockage des données des processeurs 80x86.

32-bit Compiler | Calling Convention

Les conventions d'appel font l'objet d'une description plus détaillée dans le chapitre Phrase du guide de programmation.

- pc -p **C** génère tous les appels de procédures et toutes les fonctions au moyen de la convention d'appel C, ce qui équivaldrait à déclarer toutes les procédures et fonction au moyen du mot clé `__cdecl`. Les appels de fonction qui en résultent sont moins volumineux et plus rapides. Les fonctions doivent transmettre le nombre et le type d'arguments corrects, à la différence de l'usage normal du C qui permet un nombre variable d'arguments de fonction. Vous pouvez utiliser le mot clé `__cdecl`, `__pascal`, `__stdcall` ou `__fastcall` pour déclarer spécifiquement une fonction ou procédure au moyen d'une autre convention d'appel.
- p **Pascal** génère tous les appels de procédures et toutes les fonctions au moyen de convention d'appel Pascal, ce qui équivaldrait à déclarer toutes les procédures et fonctions au moyen du mot clé `__pascal`. Les appels de fonction résultant sont moins volumineux et plus rapides. Les fonctions doivent transmettre le nombre et le type d'arguments corrects. Vous pouvez utiliser le mot clé `__cdecl`, `__stdcall` ou `__fastcall` pour déclarer spécifiquement une fonction ou procédure au moyen d'une autre convention d'appel.
- pr **Register** génère tous les appels de procédures et toutes les fonctions au moyen de la convention d'appel Register, ce qui équivaldrait à déclarer toutes les procédures et fonctions au moyen du mot clé `__fastcall`. Vous pouvez utiliser le mot clé `__stdcall`, `__pascal`, `__fastcall` ou `__cdecl` pour déclarer spécifiquement une fonction ou procédure au moyen d'une autre convention d'appel.
- ps **Standard call** utilise les conventions d'appel `stdcall`. Cette option indique au compilateur d'utiliser l'ordre Pascal pour empiler les paramètres. Les paramètres sont empilés de gauche à droite.

C++ Options | Member Pointer

- Vmp **Honor precision of member pointers** utilise la précision déclarée pour les types de pointeurs membres. Cette option doit être utilisée lorsqu'un pointeur de classe dérivée est explicitement classé comme pointeur membre d'une classe de base plus simple (lorsque le pointeur désigne en fait un membre de classe dérivée).
- Vmv **Support all cases** permet aux pointeurs membres de désigner tout membre. Les pointeurs membres utilisent la représentation la plus générale (qui n'est pas toujours la plus efficace).

- Vmm **Support multiple inheritance** permet aux pointeurs de membre de désigner les membres de classes d'héritage multiple, sauf les membres des classes virtuelles de base.
- Vms **Support single inheritance** permet aux pointeurs de membre de désigner des membres de classes d'héritage individuel uniquement.
- Vmd **Smallest for class** utilise la plus petite représentation qui permet aux pointeurs de membres de désigner tous les membres de leur classe. Si la classe n'est pas totalement définie au point où le type de pointeur membre est déclaré, le compilateur choisit la représentation la plus générale (en générant un avertissement).

C++ Options | C++ Compatibility

- K2 **Allow only two character types** traite **char** tel qu'il est signé. Compatibilité avec bc 3.1 et les versions précédentes (16 bits uniquement).
- Vb-\$|classe de base virtuelle;pointeur caché **Always near** enregistre un pointeur caché near. Lorsqu'une classe hérite virtuellement d'une classe de base, le compilateur enregistre un pointeur caché dans l'objet de classe pour accéder au sous-objet de classe virtuelle de base. Cette option génère un code moins volumineux et plus rapide.
- Vb **Same size as 'this' pointer** donne au pointeur caché la taille du pointeur utilisé par la classe proprement dite.
- Va **Pass class values via reference to temporary.** Lorsqu'une classe de types d'arguments avec constructeurs est transmise à une fonction par une valeur, cette option demande au compilateur de créer une variable temporaire au site appelant, d'initialiser cette variable temporaire selon la valeur de l'argument et de transmettre à la fonction une référence à cette variable temporaire.
- Vc **Disable constructor displacements.** Lorsque cette option est active, le compilateur n'ajoute pas de membre caché et de code à une classe dérivée (valeur par défaut). Cette option permet d'assurer la compatibilité avec les versions précédentes du compilateur.
- Vp **Push 'this' first for Pascal member functions** demande au compilateur de transmettre le pointeur 'this' aux fonctions membres 'pascal' en tant que premier paramètre de la pile. Par défaut, le compilateur transmet le paramètre 'this' comme dernier paramètre de la pile.

- Vv **"deep" virtual bases.** Lorsqu'une classe dérivée a priorité sur une fonction virtuelle dont elle hérite de la classe virtuelle de base, et qu'un constructeur ou destructeur de la classe dérivée appelle cette fonction virtuelle au moyen d'un pointeur vers la classe virtuelle de base, le compilateur peut parfois ajouter des membres cachés à la classe dérivée et ajouter du code à ses constructeurs et destructeurs. Cette option demande au compilateur de ne pas ajouter de membre caché, ni de code, pour que l'agencement d'instance de classe soit le même que celui des versions précédentes de Borland C++.
- Vt **Virtual table pointer follows data members** place le pointeur de table virtuelle après les membres de données non statiques de la classe particulière, pour assurer la compatibilité lorsque les instances de classe sont partagées avec un code autre que C++ et lors du partage de classes avec du code compilé par une version précédente de Borland C++.

C++ Options | Virtual Tables

L'option **-Vn** contrôle les tables virtuelles C++. Elle peut s'utiliser de cinq façons différentes :

- V **Smart** génère les tables virtuelles les plus communes en C++ et les fonctions inline définies hors ligne dans les modules de l'application. En conséquence, seule une instance de table virtuelle donnée ou d'une fonction **inline** définie hors ligne est incluse dans le programme pour produire des fichiers exécutables moins volumineux et plus rapides. Ils n'utilisent cependant que les extensions .OBJ et .ASM disponibles dans TLINK et TASM.
- Vs **Local** génère des tables virtuelles locales et des fonctions **inline** définies hors ligne. En résultat, chaque module possède sa propre copie des tables virtuelles et fonctions **inline** définies hors ligne qu'il utilise. Cette option produit donc des exécutables plus volumineux que l'option Smart.
- V0 **External** crée des références externes aux tables virtuelles. Si vous ne voulez pas utiliser l'option Smart ou Local, vous pouvez choisir External et Public pour produire des tables virtuelles globales et les référencer.
- V1 **Public** produit des définitions publiques de tables virtuelles. Lorsque vous utilisez l'option External ou Public, au moins un des modules du programme doit être compilé avec l'option Public pour fournir les définitions des tables virtuelles. Tous les autres modules doivent être compilés au moyen de l'option **-V0** pour faire référence à la copie publique des tables virtuelles.

C++ Options | Templates

Pour plus de détails sur les modèles, consultez le chapitre plus dans le *Guide de programmation*.

- Jg **Smart** génère des définitions publiques de toutes les instances de modèle. Si plus d'un module génère la même instance de modèle, l'éditeur de liens les fusionne pour produire une seule copie de l'instance. Pour générer des instances, le compilateur doit cependant disposer du corps de la fonction (dans le cas d'une fonction de modèle) ou du corps des fonctions membres et des définitions pour les membres de données statiques (dans le cas d'une classe de modèles).
- Jgd **Global** génère des définitions publiques pour toutes les instances de modèle rencontrées. Les instances dupliquées ne sont pas fusionnées, obligeant ainsi l'éditeur de liens à signaler des erreurs de redéfinition de symbole public si plus d'un module définit la même instance de modèle.
- Jgx **External** génère les références externes aux instances de modèle. Vérifiez que les instances sont définies publiquement dans un autre module (au moyen de l'option **-Jgd**), de sorte que les références externes soient correctement résolues.

C++ Options | Exception handling/RTTI

- x **Enable exceptions** permet la gestion des exceptions C++. Si vous utilisez les constructions de gestion d'exceptions C++ dans le code et que vous compilez avec cette option désactivée (en annulant la coche de l'option dans l'EDI ou en utilisant l'option en ligne de commande **-x-**), une erreur se produit.
- xp **Enable exception location** information fournit une identification des exceptions au moment de l'exécution en fournissant les numéros de ligne où se produisent les exceptions dans le code source. Ceci permet au programme d'interroger le fichier et le numéro de ligne auquel l'exception C++ s'est produite.

- xd **Enable destructor cleanup.** Des destructeurs sont appelés pour tous les objets déclarés automatiquement dans la portée des instructions `catch` et `throw` lorsqu'une exception est rejetée. Notez que les destructeurs ne sont pas appelés automatiquement pour les objets dynamiques et que les objets dynamiques ne sont pas libérés automatiquement.
- RT **Enable runtime type information** génère le code qui permet l'identification de type au moment de l'exécution.

Optimisations

Le compilateur Borland contient un optimiseur permettant d'accélérer une application ou de réduire sa taille. La compilation ne prend que 50 % plus longtemps et 20 % de plus pour l'optimisation de taille. Vous effectuez une compilation avec optimisation à tout moment pendant le cycle du projet. Au débogage, la compilation avec optimisation aide parfois à révéler des bogues dans le code (le débogueur intégré fonctionne avec le code optimisé).

- Od **Disable all optimizations** désactive toutes les optimisations. Vous pouvez donner aux options du gestionnaire de projet la priorité sur cette option.

Optimizations | Specific

- O2 -Ot -Ox -G **Executable speed** crée un code plus rapide. Le compilateur détermine s'il peut générer le code en toute sécurité pour exécuter une instruction `rep movsw`, au lieu d'appeler une fonction d'aide pour effectuer la copie. Il produit ainsi des copies de structures plus rapides pour les structures et unions de plus de huit octets qu'avec l'appel de fonction d'aide. L'option en ligne de commande **-Ox** a été ajoutée pour la compatibilité avec le compilateur Microsoft.
- O1 -Os -G- **Executable size** crée le code le plus compact en examinant le contenu du code généré pour identifier les séquences dupliquées. Lorsque de telles séquences le permettent, l'optimiseur remplace une séquence de code par un branchement à la séquence identique et élimine la première partie du code. Cette situation se produit le plus souvent avec les instructions **switch**.

No optimization inhibe l'optimisation des sous-expressions communes. Cette option est activée par défaut. Le fonctionnement par défaut des compilateurs en ligne de commande ne comprend pas ce type d'optimisation. Il n'y a donc pas d'option en ligne de commande équivalente : il vous suffit de ne rien indiquer.

- Oc **Optimize locally** élimine les sous-expressions communes dans un groupe d'instructions non interrompues par des branchements (blocs de base).
- Og **Optimize globally** élimine les expressions dupliquées dans la portée cible et enregistre immédiatement la valeur calculée de ces expressions (au lieu de recalculer les expressions). Bien qu'en théorie, cette optimisation soit de nature à réduire la longueur du code, elle effectue en fait une optimisation de la vitesse et réduit rarement la taille du code. Utilisez cette option si vous préférez réemployer les expressions au lieu de les recalculer pour chaque instance.
- Oa **Assume no pointer aliasing** affecte la façon dont l'optimiseur effectue l'élimination des sous-expressions communes et la propagation des copies en permettant à l'utilisateur de conserver des informations de propagation des copies à travers les appels de fonction et de conserver des informations de sous-expressions communes à travers certains registres. Sans cette option, l'optimiseur doit supprimer les informations sur les copies et les sous-expressions. Le crénelage de pointeur risque d'introduire des bugs difficiles à repérer et n'est donc appliqué que lorsque vous utilisez **-Oa**.

-Oa détermine comment l'optimiseur doit traiter les expressions désignées par les pointeurs. En compilant des sous-expressions communes lorsque **-Oa** est activé, l'optimiseur reconnaît `*p * x` comme sous-expression commune dans la fonction **func1**.

```
int g, y;
int func1(int *p)
{
    int x=5;
    y = *p * x;
    g = 3;
    return (*p * x);
}
void func2(void)
{
    g=2;
    func1(&g); // Correct-l'affectation g = 3
              // invalide l'expression *p * x
}
```

Optimizations | Size

- O **Jump optimizations** optimise les branchements. Lorsque l'option Jump Optimization est active, le compilateur réduit la taille du code en éliminant les branchements redondants et réorganise les instructions de boucles et de switch. Lorsque cette option est active, les séquences de tracé normal et approfondi dans le débogueur peuvent prêter à confusion dans la mesure où plusieurs lignes du code source peuvent être associées à une séquence de code générée. En désactivant cette option, vous obtiendrez de meilleurs résultats de tracé au débogage.
- Ol **Loop optimization** exploite les instructions de déplacement de chaînes des processeurs 80x86 en remplaçant le code d'une boucle par une instruction de déplacement de chaîne pour accélérer le code.

```
int v[100];
void t(void)
{
    int i;
    for (i = 0; i < 100; i++)
        v[i] = 0;
}
```

Selon la complexité des opérandes, le code de boucle compacté est parfois plus court que la boucle non compactée correspondante.

- Z **Suppress redundant loads** supprime les chargements redondants. Cette option, qui optimise à la fois la vitesse d'exécution et la taille de l'application en prenant note des valeurs chargées dans les registres, doit être utilisée pour compiler avec optimisation. Les valeurs figurant déjà dans un registre ne sont pas rechargées.
- Ob **Dead-code elimination** révèle des variables qui ne sont pas toujours nécessaires. Comme l'optimiseur doit déterminer où les variables ne sont plus utilisées et où leur valeur est nécessaire (analyse de zone active), vous devez utiliser l'allocation de registre global (-Oe) avec -Ob.
- OW **Windows prolog/epilog** supprime le inc bp / dec bp d'un code de prologue et d'épilogue de fonction far Windows. Si les informations de débogage sont activées dans l'option OBJs (-v), cette option est désactivée parce que certains outils de débogage (comme WinSpector et Turbo Debugger pour Windows) ont besoin de inc bp/dec bp pour afficher les informations de cadre de pile.

- Oe **Global register allocation**, que vous devez toujours utiliser en optimisant le code, accélère la vitesse d'exécution et réduit la taille de l'application. Lorsque l'option d'allocation de registre est active, l'allocation de registre global et l'analyse de zone active variable sont également actives.

Optimizations | Speed

- Oi **Inline intrinsic functions** génère le code pour les fonctions de mémoire (comme **strcpy** ou **memcpy**) dans la portée de votre fonction, éliminant ainsi la nécessité d'un appel de fonction. Le code résultant est s'exécute plus rapidement, mais est aussi plus volumineux. Les fonctions suivantes sont affectées par un traitement inline avec cette option :

■ <code>alloca</code>	■ <code>memset</code>	■ <code>strchr</code>	■ <code>strncmp</code>
■ <code>fabs</code>	■ <code>rotl</code>	■ <code>strcmp</code>	■ <code>strncpy</code>
■ <code>memchr</code>	■ <code>rotr</code>	■ <code>strcpy</code>	■ <code>strnset</code>
■ <code>memcpy</code>	■ <code>stpcpy</code>	■ <code>strlen</code>	■ <code>strrchr</code>
■ <code>memcpy</code>	■ <code>strcat</code>	■ <code>strncat</code>	

Vous pouvez déterminer si les fonctions sont inline ou non au moyen du `pragma intrinsic`. A titre d'exemple, `#pragma intrinsic strcpy` génère le code inline de tous les appels **strcpy** suivants dans la fonction, tandis que `#pragma intrinsic -strcpy` empêche le compilateur d'effectuer un inline pour **strcpy**. L'utilisation de ces pragmas dans un fichier a priorité sur les options du compilateur.

Lors d'un traitement inline des fonctions intrinsèques, vous devez inclure un prototype pour la fonction avant de l'utiliser. En effet, le compilateur crée une macro qui renomme la fonction inline en une fonction que le compilateur reconnaît en interne. Dans l'exemple précédent, le compilateur crée la macro `#define strcpy _strcpy_ _`.

Le compilateur reconnaît les appels de fonction avec deux caractères de soulignement à gauche et deux caractères de soulignement à droite et essaie de faire correspondre le prototype de la fonction à son propre prototype interne. Si vous ne fournissez pas de prototype, ou si le prototype fourni ne correspond pas au prototype du compilateur, le compilateur rejette la tentative de traitement inline de la fonction et génère une erreur.

- Om **Invariant code motion** extrait le code invariable des boucles pour optimiser la vitesse d'exécution. L'optimiseur utilise les informations au sujet de toutes les expressions de la fonction qui ont été rassemblées pendant l'élimination des sous-expressions communes pour déterminer les expressions dont la valeur ne change pas dans une boucle. Pour éviter de répéter un calcul à plusieurs reprises dans la boucle, l'optimiseur extrait le code de la boucle afin qu'il ne soit calculé qu'une seule fois. L'optimiseur réutilise ensuite la valeur calculée à l'intérieur de la boucle. Vous devez utiliser l'extraction de code invariable de boucle à chaque fois que vous compilez pour optimiser la vitesse d'exécution et que vous avez utilisé des sous-expressions communes globales. En effet, en extrayant le code des boucles, il est possible d'obtenir une amélioration considérable de la vitesse d'exécution. Dans le code suivant, par exemple, l'expression $x * y * z$ est évaluée à chaque tour de boucle :

```
int v[10];
void f(void)
{
    int i, x, y, z;
    for (i = 0; i < 10; i++)
        v[i] = x * y * z;
}
```

L'optimiseur réécrit le code de la manière suivante :

```
int v[10];
void f(void)
{
    int i, x, y, z, t1;
    t1 = x * y * z;
    for (i = 0; i < 10; i++)
        v[i] = t1;
}
```

- Op **Copy propagation** sert principalement à optimiser la vitesse d'exécution, mais n'augmente jamais la taille du code. Comme l'extraction de code invariable de boucle, la propagation de copie se fie à l'analyse exécutée pendant l'élimination des sous-expressions communes. Grâce à la propagation de copie, l'optimiseur mémorise la valeur des expressions attribuées. Les copies des constantes, expressions et variables peuvent être propagées.

- Ov **Induction variables** crée les variables d'induction et exécute la réduction de puissance pour optimiser la vitesse d'exécution des boucles. Elle doit être utilisée lorsqu'une application est compilée pour accélérer sa vitesse d'application et que le code associé contient des boucles. L'optimiseur fait appel à l'induction pour créer de nouvelles variables (*variables d'induction*) à partir des expressions utilisées dans les boucles. L'optimiseur s'assure que les opérations effectuées sur ces nouvelles variables utilisent moins de temps système (puissance réduite) que sur les variables d'origine.

Les optimisations sont communes lorsque vous utilisez une indexation à l'intérieur des boucles, dans la mesure où une opération de multiplication est nécessaire pour calculer la position indiquée par l'index dans un tableau. Dans le code suivant par exemple, l'optimiseur crée une variable d'induction à partir de l'opération parce que celle-ci demande une multiplication, ce qui évite aussi de conserver la valeur de *i* :

```
int v[10];
void f(void)
{
    int i,x,y,z;
    for (i = 0; i < 10; i++)
        v[i] = x * y * z;
}
```

Lorsque les variables d'induction sont actives, le code change comme suit :

```
int v[10];
void f(void)
{
    int i,x,y,z, *p;
    for (p = v; p < &v[10]; p++)
        *p = x * y * z;
}
```


Messages

Les messages activés par défaut sont signalés par un astérisque à côté de la ligne de commande ; ces options sont cochées dans l'EDI.

- w **All** affiche tous les messages d'avertissement.
- WXXX -w-XXX **Selected** active le message d'avertissement spécifique entré en ligne de commande ou coché dans l'EDI. Si le pragma **warn** est utilisé dans le code source, il a priorité sur les options de message définies en ligne de commande ou dans l'EDI. Pour plus de détails sur les pragmas, voir le chapitre 5 du *Guide du programmeur*.
- w- **None** n'affiche pas les messages d'avertissement, mais les erreurs sont quand même signalées.
- gn **Stop after n warnings** interrompt la compilation au bout de n avertissements dans le projet.
- jn **Stop after n errors** interrompt la compilation au bout de n erreurs dans le projet.

Messages | Portability

- wrpt* Nonportable pointer conversion.
- wcpt* Nonportable pointer comparison.
- wrng* Constant out of range in comparison.
- wcln Constant is long.
- wsig Conversion may lose significant digits.
- wucp Mixing pointers to different 'char' types.

Messages | ANSI Violations

- wvoi* Void functions may not return a value.
- wret* Both return and return with a value used.
- wsus* Suspicious pointer conversion.
- wstu* Undefined structure *structure*.

- wdup* Redefinition of *macro* is not identical.
- wbig* Hexadecimal value contains more than 3 digits.
- wbbf Bit fields must be **signed** or **unsigned int**.
- wext* *Identificateur* is declared as both external and static.
- wdpu* Declare *type* prior to use in prototype.
- wzdi* Division by zero.
- wbei* Initializing *identificateur* with *identificateur*.
- wpin Initialization is only partially bracketed.
- wnak Non-ANSI keyword used: *mot*.

Messages | Obsolete C++

- wobi* Base initialization without a class name is now obsolete.
- wofp* Style of function definition is now obsolete.
- wpre* Overloaded prefix operator used as a postfix operator.

Messages | Potential C++ Errors

- wnci* Constant member *identificateur* is not initialized.
- weas* Assigning *type* to enumeration.
- whid* *Fonction1* hides virtual function *fonction2*.
- wncf* Non-const function *fonction* called for const object.
- wibc* Base class *base1* is also a base class of *base2*.
- wdsz* Array size for 'delete' ignored.
- wnst* Use qualified name to access nested type *type*.
- whch* Handler for *xxx* is hidden by previous handler for *yyy*.
- wmpc* Conversion to type will fail for members of virtual base *base*.
- wmpd* Maximum precision used for member pointer type *type*.

- wntd* Use '> >' for nested templates instead of '>>'.
- wncf* Non-volatile function *fonction* called for volatile object.

Messages | Inefficient C++ Coding

- winl* Functions containing *identificateur* are not expanded inline.
- wlin* Temporary used to initialize *identificateur*.
- wlvc* Temporary used for parameter in call to *identificateur*.

Messages | Potential errors

- wpia* Possibly incorrect assignment.
- wdef* Possible use of *identificateur* before definition.
- wnod* No declaration for function *fonction*.
- wpro* Call to function with no prototype.
- wrvl* Function should return a value.
- wamb* Ambiguous operators need parentheses.
- wccc* Condition is always true/false.

Messages | Inefficient Coding

- waus* *Identificateur* is assigned a value that is never used.
- wpar* Parameter *identificateur* is never used.
- wuse* *Identificateur* is declared but never used.
- wstv* Structure passed by value.
- wrch* Unreachable code.
- weff* Code has no effect.

Messages | General

- wasm Unknown assembler instruction.
- will* Ill-formed pragma.
- wias* Array variable *variable* is near.
- wamp Superfluous & with function.
- wobs* *Identificateur* is obsolete.
- wpch* Cannot create precompiled header: *en-tête*.

Make

- w! **Break Make on warnings** renvoie un code de retour différent de zéro à partir du compilateur en ligne de commande lorsqu'il y a des avertissements et ne crée pas de fichiers .OBJ compilés.

Options en ligne de commande

Les options énumérées ci-dessous ne peuvent être utilisées qu'avec les compilateurs en ligne de commande (BCC.EXE et BCC32.EXE). Il n'existe pas d'options directement équivalentes dans l'EDI ; cependant, comme la plupart de ces tâches peuvent être effectuées dans l'EDI, chaque option contient des instructions pour l'EDI.

- 1 Génère des instructions 80186 étendues, et des programmes 80286 exécutables en mode réel.
- B Compile en assembleur et appelle TASM pour assembler le code. Si vous ne possédez pas TASM, cette option génère une erreur. En outre, les anciennes versions de TASM risquent de poser des problèmes avec le code 32 bits.
- c Compile et assemble les fichiers .C, .CPP et .ASM spécifiés, mais n'exécute pas de commande d'édition de liens. Choisissez l'option Project | Compile dans l'EDI.

- e*fichier* Dérive le nom du programme exécutable du *nom de fichier* spécifié et lui ajoute l'extension de *fichier*.EXE (le nom de programme est alors *fichier*.EXE). Le nom de fichier doit suivre immédiatement l'option **-e**, sans espace intermédiaire. Sans cette option, l'éditeur de liens dérive le nom du fichier .EXE du nom du premier fichier source ou objet dans la liste de noms de fichier.
- E*fichier* Fichier est le nom de l'assembleur à utiliser, par défaut TASM. Dans l'EDI, vous pouvez ajouter un outil pour le programme assembleur à utiliser. Pour plus de détails sur l'addition d'outils dans l'EDI, reportez-vous au chapitre 2.
- f87 Utilise des instructions pour matériel 8087 (DOS 16 bits uniquement)
- f287 Utilise des instructions pour matériel 80287 (DOS 16 bits uniquement). Dans l'EDI, cochez l'option Fast floating point de TargetExpert lorsque vous créez une cible DOS
- Fm Cette option active toutes les autres options **-F (-Fc, -Ff et -Fs)**. Elle constitue un raccourci pratique pour porter du code à partir d'autres compilateurs. Pour obtenir le même résultat dans l'EDI, cochez les options de **-Fc, -Ff et -Fs**.
- Hc En-têtes précompilés de cache. Cette option doit être utilisée avec **-H** ou **-Hxxx**. Elle est utile lorsque la compilation se fait sur plus d'un en-tête précompilé.
- lx Transmet l'option x à l'éditeur de liens (TLINK pour BCC et TLINK32 pour BCC32). Plus d'une option peut être précisée après **-l** (L minuscule). Pour sélectionner les options d'éditeur de liens dans l'EDI, choisissez Options | Project | Linker. Pour obtenir la liste des options d'éditeur de liens, reportez-vous au chapitre 9.
- lx Supprime l'option x de l'éditeur de lien. Plus d'une option peut être précisée après le **-l** (L minuscule suivi d'un tiret). Vous pouvez cocher et annuler les options de l'éditeur de liens dans l'EDI en sélectionnant Options | Project | Linker.
- M Oblige l'éditeur de liens à produire une table de liaison complète. Par défaut, il ne produit pas de table de liaison. Dans l'EDI, cochez Segment, Public ou Detailed dans la partie Linker | MapFile de la boîte de dialogue Project Options.
- o*fichier* Compile le fichier spécifié pour produire *fichier*.obj.

- P- Compile les fichiers ayant l'extension .CPP en tant que fichiers C++ ; les autres fichiers sont compilés en tant que fichiers C. Dans l'EDI, utilisez des outils différents pour compiler un noeud de projet. Pour plus de détails, reportez-vous au chapitre 2.
- P Compile tous les fichiers en tant que fichiers C++, quelle que soit leur extension. Dans l'EDI, utilisez des outils différents pour compiler un noeud de projet. Pour plus de détails, reportez-vous au chapitre 2.
- Pext Compile tous les fichiers en tant que fichiers C++ ; cette option modifie l'extension par défaut pour utiliser l'extension spécifiée Cette option est disponible parce que certains programmeurs utilisent .C ou toute autre extension comme extension par défaut dans le code C++. Dans l'EDI, utilisez des outils différents pour compiler un noeud de projet. Pour plus de détails, reportez-vous au chapitre 2.
- P-ext Effectue la compilation en fonction de l'extension (.CPP pour le code C++, tout autre extension de nom de fichier pour code code C) et définit l'extension par défaut (autre que .CPP). Dans l'EDI, utilisez des outils différents pour compiler un noeud de projet. Pour plus de détails, reportez-vous au chapitre 2.
- S Cette option compile les fichiers source spécifiés et produit des fichiers de sortie en langage assembleur (.ASM), mais n'effectue pas l'assemblage proprement dit. Lorsque vous utilisez cette option, Borland C++ inclut les lignes source C ou C++ comme commentaires dans le fichier .ASM produit. Dans l'EDI, utilisez différents outils pour compiler un noeud de projet. Sélectionnez Special | C++ to Assemble dans le TurboMenu de la fenêtre projet. Pour plus de détails, reportez-vous au chapitre 2.
- T*chaîne* Passe une *chaîne* sous forme d'option à TASM (ou une option à l'assembleur défini par l'option **-E**).
- T Supprime toutes les options d'assembleur définies précédemment.
- tD -De Crée un fichier .EXE 16 bits pour DOS. Dans l'EDI, choisissez ce type de cible lorsque vous créez la cible de votre projet.
- tDc Crée un fichier .COM 16 bits pour DOS. L'EDI ne permet pas de créer ce type de fichiers
- tWC -WC Crée une application en mode console 32 bits. Dans l'EDI, sélectionnez ce type de cible pour créer une cible de projet.
- tW -W Crée une application GUI 32 bits. Dans l'EDI, sélectionnez ce type de cible pour créer une cible de projet.

- tWCD -WCD Crée une DLL de mode console 32 bits avec toutes les fonctions exportées. Dans l'EDI, sélectionnez ce type de cible pour créer une cible de projet.
- tWCDE -WCDE Crée une DLL en mode console 32 bits avec fonctions explicites exportées. Dans l'EDI, sélectionnez ce type de cible en créant une cible de projet.
- tWD -WD Crée une DLL GUI 32 bits avec toutes les fonctions exportables. Dans l'EDI, sélectionnez ce type de cible pour créer une cible de projet.
- tWDE -WDE Crée une DLL GUI 32 bits avec fonctions explicites exportables. Dans l'EDI, sélectionnez ce type de cible pour créer une cible de projet.
- tWM -WM Crée une application ou une DLL multithread. Utilisez cette option avec -Wm et -WCD. Dans l'EDI, sélectionnez ce type de cible pour créer une cible de projet.
- Unom Annule toute définition de l'identificateur spécifié.
- Vo Cette option active toutes les options de compatibilité ascendante répertoriées dans cette section. Elle constitue un raccourci pratique pour lier les bibliothèques créées au moyen d'une version plus ancienne de Borland C++.
- Y Permet la génération du code de recouvrement. Dans l'EDI, choisissez DOS Overlay comme type de cible lorsque vous créez une cible pour votre projet.
- Yo Demande le recouvrement des fichiers compilés. Dans l'EDI, cochez Overlay this module dans la boîte de dialogue Node Attribute pour tous les noeuds rattachés à une cible DOS Overlay (-Y).
- zX* Utilise le nom par défaut de X. Par exemple, -zA* attribue le nom de classe par défaut CODE à la classe de segments de code.

Création d'applications avec AppExpert

Reportez-vous au chapitre 5 ou à l'aide en ligne pour plus de renseignements sur AppExpert.

Ce chapitre vous explique comment utiliser AppExpert pour créer des applications Windows à partir d'ObjectWindows 2.0. Exploitant les classes de Resource Workshop et d'ObjectWindows 2.0 ainsi que le gestionnaire de projet de l'EDI, AppExpert permet une approche visuelle de la génération d'applications. Familiarisez-vous avec ces éléments pour une utilisation optimale d'AppExpert.

AppExpert vous permet de créer un fichier Windows exécutable avec des fonctions du type barre d'icônes, barre d'état, menus, aide en ligne et fenêtres MDI. Vous pouvez aussi sélectionner des options pour implémenter les fonctions d'impression, d'aperçu avant impression et de Document/View.

Notions de base sur AppExpert

La création d'applications avec AppExpert s'effectue en quatre temps :

1. Utilisez ClassExpert pour définir l'interface utilisateur et les fonctions de l'application, ainsi que pour générer le code.
2. Utilisez ClassExpert pour ajouter des classes et des gestionnaires d'événements, pour implémenter des fonctions virtuelles et pour explorer le code source des classes existantes. ClassExpert peut également associer des objets créés avec Resource Workshop à des classes ou des gestionnaires. Utilisez ClassExpert si vous avez des problèmes de gestion d'événements, d'implémentation de fonctions virtuelles et de variables modèles.
3. Utilisez Resource Workshop pour modifier ou ajouter des ressources.
4. Utilisez le gestionnaire de projet pour créer le fichier exécutable.

AppExpert crée obligatoirement les fichiers suivants pour chaque application :

AppExpert crée un fichier .APX contenant des informations importantes qui seront utilisées par ClassExpert.

- Un fichier de projet (.IDE)
- Un fichier source principal (.CPP)
- Un fichier d'en-tête principal (.H)
- Un fichier script de ressource (.RC)
- Un fichier d'en-tête de ressource (.RH)
- Un fichier de base de données pour le programme AppExpert source (.APX)

Selon les options choisies, AppExpert peut créer les fichiers suivants :

- des fichiers d'aide source (.RTF)
- un fichier de projet d'aide (.HPJ).
- des fichiers d'icônes et de bitmap (.ICO et .BMP)

Création d'une application avec AppExpert

Cette section vous explique comment créer une application AppExpert.

1. Lancez l'EDI et sélectionnez Project | AppExpert. Une boîte de dialogue apparaît.
2. Tapez le nom de votre fichier de projet. La plupart des fichiers générés (notamment les fichiers .EXE) sont par défaut dérivés du nom du projet (par exemple, <nom_prj>.CPP).
3. Sélectionnez le chemin d'accès où sera enregistré le fichier de projet AppExpert (AppExpert crée le répertoire s'il n'existe pas). Ce répertoire devient le répertoire par défaut pour tous les fichiers source créés (vous pouvez changer le répertoire par défaut dans la boîte de dialogue Application Generation Options avant de générer l'application). Vous pouvez, si vous le souhaitez, mettre chaque projet AppExpert dans son propre répertoire pour faciliter la modification des fichiers. Cliquez sur OK. La boîte de dialogue Application Generation Options d'AppExpert apparaît.

- ▣ Application
 - Basic Options
 - Advanced Options
 - Code Gen Control
 - Admin Options

4. Vous pouvez cliquer sur le bouton Generate dans le bas de la boîte de dialogue pour générer l'application Windows par défaut ; vous pouvez aussi modifier les options affichées dans la boîte de dialogue avant de générer l'application. La boîte de dialogue présente une liste de sujets sur la gauche et à droite un bref descriptif du sujet sélectionné (appuyez sur le bouton Help pour plus de renseignements sur les options associées au sujet concerné). Pour changer d'options d'application,
 - Visualisez les options en cliquant sur un + pour afficher un des sous-sujets (le + signifie que le sujet comporte des sous-sujets, un - que tous les sous-sujets sont affichés), puis sélectionnez un sous-sujet. Vous pouvez par exemple cliquer sur le + qui suit Application (ou double-cliquer sur le mot "Application"), puis sélectionner le sous-sujet Basic Options. Le tableau à droite affiche les options de base de toute application AppExpert.
 - Choisissez les options que vous voulez inclure dans votre application. Vous pouvez par exemple modifier le répertoire Base où sont sauvegardés les fichiers AppExpert (et que vous avez indiqué à l'étape 3).
5. Cliquez sur le bouton Generate dans le bas de la boîte de dialogue Options.
6. Une boîte de dialogue confirmant la génération du code apparaît. Cliquez sur Yes pour générer le code (cliquez sur No pour revenir aux options de configuration). Pendant qu'AppExpert génère votre application, une boîte de message apparaît.

AppExpert crée tous les fichiers destinés à votre application et les place dans le répertoire Base (vous pouvez modifier les répertoires avant de générer l'application ; voir l'option Base Directory à la section "Application | Basic_Options" ci après).

Avec AppExpert, vous choisissez une fois pour toutes vos options d'application avant de générer le code. Lorsque le code et les ressources sont générés, vous pouvez les modifier et en ajouter d'autres, mais vous ne pouvez pas revenir à AppExpert pour changer d'options. Par exemple, si vous générez une application qui ne contient pas de ligne d'état, vous ne pouvez pas utiliser AppExpert pour ajouter cette fonction : il vous faudra l'ajouter manuellement.

7. La fenêtre du projet apparaît avec la liste de quelques-uns des fichiers indispensables à votre application (les fichiers bitmap, les fichiers d'icônes et de textes d'aide ne sont pas affichés). Vous pouvez utiliser ClassExpert pour modifier votre application, ou bien vous pouvez d'abord créer votre application (reportez-vous au chapitre 5 pour en savoir plus sur ClassExpert). Pour créer votre application, sélectionnez Project | Make all (vous pouvez sélectionner Build all, mais Make all est plus rapide). Le fichier exécutable (.EXE) est sauvegardé par défaut dans le répertoire Base.

**Applications
AppExpert par
défaut**

Si vous ne modifiez aucune des options d'AppExpert au moment où vous générez votre application, vous obtenez une application par défaut. Vous pouvez explorer les options pour savoir quelles sont les options par défaut.

Le fait de désactiver les options dont vous n'avez pas besoin accélère la génération et la création de l'application.

Options du menu Application

Les options du menu Application déterminent le style de votre application.

Multiple Document Interface définit le style de votre application conformément au modèle Multiple Document Interface (MDI).

Single Document Interface définit le style de votre application conformément au modèle Single Document Interface (SDI).

Document/View détermine si votre application gèrera les objets de l'application selon le modèle Document/View. "Document" correspond aux données et "view" à l'interface entre l'utilisateur et les données. Dans un modèle Document/View, ces deux éléments sont séparés (reportez-vous au *Guide du programmeur Object Windows* pour plus d'informations sur Document/View). Vous pouvez utiliser cette option pour des applications SDI ou MDI.

SpeedBar place une barre d'icônes dans le haut de la fenêtre principale de votre application.

Status Line place une ligne d'état dans le bas de la fenêtre principale de votre application et génère un code pour afficher des messages d'aide sur la ligne d'état quand des options de menu sont mises en surbrillance.

Drag/Drop traite les opérations Windows standard de glisser-déplacer.

Printing traite les activités d'impression et crée les menus File | Print Setup, Print Preview et Print.

Application | Basic Options

Les options Basic définissent l'endroit où est stocké le code généré et commandent les fichiers d'aide.

Target Name prend le nom du projet que vous voulez créer comme base du nom par défaut des autres éléments du projet (par exemple, les fichiers d'en-tête, de base de données des classes, des classes de l'application et les fichiers source).

Base Directory définit le chemin d'accès du répertoire de base qui sera utilisé pour enregistrer tous les répertoires du projet. Tous les chemins du projet sont liés à ce répertoire. Vous pouvez soit taper un nom de votre choix, soit en sélectionner un dans la boîte de dialogue Browse (cliquez sur le bouton Browse). Le nom du répertoire est transmis au gestionnaire de projet pour la nouvelle application AppExpert à créer. La valeur par défaut du répertoire de base est le répertoire du noeud père du projet défini dans le gestionnaire de projet. Si vous donnez un nouveau nom, AppExpert crée le répertoire correspondant.

Help File Support génère des fichiers sources d'aide (.RTF) et un fichier d'aide (.HPJ). Le fichier projet d'aide est ajouté au gestionnaire de projet ; il est automatiquement créé avec l'application résultante. Le fichier source d'aide comprend un texte de marque de réservation destiné aux options des menus de l'application.

Help File Name indique le nom des fichiers d'aide (.HLP et .HPJ) associés à votre application.

Application | Advanced Options

Les options Advanced déterminent le comportement de votre application au démarrage, ainsi que son aspect.

Start Up définit l'état initial de la fenêtre principale de l'application.

- Normal (par défaut) : démarrage avec un format par défaut (défini par WS_NORMAL).
- Minimized : démarrage avec une icône dans l'écran Windows.
- Maximized : remplit la totalité de l'écran Windows au démarrage.

Control Style détermine le type des commandes utilisées par l'application.

- Windows (par défaut) : commandes Windows standard.
- BWCC : style de commandes spécifique à Borland.
- 3D : nouvelles commandes Windows en trois dimensions.

Les options de Code Gen Control concernent divers aspects du processus de génération du code et déterminent l'endroit où est enregistré le code généré.

Target Name affiche le nom du projet tel qu'il a été défini dans BasicOptions | Target.

Base Directory affiche le nom du répertoire de base du projet, tel que défini dans Basic Options | Base Directory.

Source Directory indique le répertoire où sont enregistrés les fichiers source de l'application. Ce chemin d'accès utilise comme point de départ le répertoire indiqué comme Base Directory. Si un chemin absolu est indiqué, il est converti en chemin lié au Base Directory (vous ne pouvez spécifier un autre lecteur). Vous pouvez taper directement un nom de répertoire de votre choix ou en sélectionner un dans une liste (cliquez sur le bouton Browse). La valeur par défaut du chemin source est ".\".

Header Directory indique le répertoire où sont enregistrés les fichiers d'en-têtes de l'application. Ce chemin d'accès utilise comme point de départ le répertoire indiqué comme Base Directory. Si un chemin absolu est indiqué, il est converti en chemin lié au Base Directory (vous ne pouvez spécifier un autre lecteur). Vous pouvez taper directement un nom de répertoire de votre choix ou en choisir un dans une liste (cliquez sur le bouton Browse). La valeur par défaut du chemin des en-têtes est ".\".

Main Source File désigne le fichier source principal de l'application.

Main Header File désigne le fichier d'en-têtes principal de l'application.

Application Class désigne la classe qui est dérivée par AppExpert de *TApplication*. Le nom de la classe par défaut est formé à partir du nom du projet.

About Dialog Class désigne la classe qui est dérivée par AppExpert de *TDialog*. Le nom de la classe par défaut est formé à partir du nom du projet.

Comments donne des renseignements partiels (terse) ou complets (verbose) sur le code généré.

**Application|Admin
Options**

Les options Admin permettent d'identifier les informations placées dans un bloc de commentaires placé au début de tous les fichiers générés pour le projet. Une partie de ces informations est affichée dans la boîte de dialogue Help | About de l'application.

Version Number correspond au numéro de version du projet qui s'affiche dans la boîte de dialogue Help | About (le numéro par défaut est "1.0"). Cette information est stockée dans le fichier .RC de votre projet.

Copyright définit les informations de copyright qui s'affichent dans la boîte de dialogue Help | About de l'application.

Description décrit l'application et affiche le texte correspondant dans la boîte de dialogue Help | About de l'application. La valeur par défaut est le nom du projet.

Author désigne les programmeurs qui génèrent le code source ; il accompagne tout commentaire sur le code généré.

Company désigne la société des programmeurs ; il accompagne tout commentaires sur le code généré.

Options du menu Main Window

Les options du menu Main Window commandent les fonctions de la fenêtre principale de votre application, son aspect et son type.

Window title désigne le texte destiné à la barre de titre de la fenêtre principale de l'application.

Background color définit la couleur du fond de la fenêtre principale de l'application ; cliquez sur le bouton Background color pour sélectionner une couleur.

**Main Window |
Basic Options**

Les options Basic commandent l'aspect général de la fenêtre principale de l'application.

Window Styles commande l'aspect de la fenêtre principale de l'application en indiquant le style de ses zones non-clientes.

- **Caption** crée un cadre à simple bordure et une barre de titre aux endroits où une légende peut être affichée.
- **Border** entoure la fenêtre principale d'une simple bordure, sans barre de titre.

- **Max box** ajoute un bouton d'agrandissement à droite de la barre de titre de la fenêtre principale. Cette option n'est disponible que si l'option Caption est active.
- **Min box** ajoute un bouton de réduction à droite de la barre de titre de la fenêtre principale (disponible uniquement si l'option Caption est active).
- **Vertical scroll** ajoute une barre de défilement vertical à droite de la fenêtre principale. Cette option n'est disponible que si vous cochez Caption ou Border.
- **Horizontal scroll** ajoute une barre de défilement horizontal dans le bas de la fenêtre principale. Cette option n'est disponible que si vous cochez Caption ou Border.
- **System menu** ajoute une case de menu système Windows à gauche de la barre de titre de la fenêtre principale (disponible uniquement si l'option Caption est active).
- **Visible** rend la fenêtre principale visible. Quand Visible est désactivé, le style `WS_VISIBLE` est remplacé par `NOT WS_VISIBLE`.
- **Disabled** désactive la fenêtre principale par défaut (par exemple, si vous voulez afficher une image quand l'application est lancée).
- **Thick frame** entoure la fenêtre principale d'une double bordure et rend la taille de la fenêtre réglable.
- **Clip siblings** protège les noeuds des fenêtres filles. Le rafraîchissement d'écran est limité à cette fenêtre (voir `WS_CLIPSIBLINGS` dans l'aide en ligne de l'API).
- **Clip children** empêche les fenêtres filles d'être recouvertes par le rafraîchissement de la fenêtre principale de l'application (voir `WS_CLIPCHILDREN` dans l'aide en ligne de l'API).

Main Window | SDI Client

SDI Client définit la classe correspondant à la zone cliente de la fenêtre principale de l'interface SDI.

Client/view class indique le nom de la classe de fenêtre ou d'affichage des zones clientes de la SDI. L'interprétation de cette valeur varie selon que vous avez ou non sélectionné l'option Document/view au niveau de Application Model. Si Document/view est actif, Client/view class sélectionne la classe d'affichage attribuée par défaut à Document/view. Si Document/view n'est pas actif, Client/view class sélectionne la classe de la fenêtre cliente.

Tableau 4.1
Client/view class
avec Document/view

Document/view actif	Document/view inactif
<i>TEditView</i> (par défaut)	<i>TEditFile</i> (par défaut)
<i>TListView</i>	<i>TListBoxT</i>
<i>WindowView</i>	<i>TWindow</i>

Cette valeur est automatiquement mise en correspondance avec la valeur attribuée à Document/view. Par exemple, si vous désactivez l'option Document/view, *TListView* bascule sur *TListBox*. Inversement, si vous activez l'option Document/view, *TListBox* bascule sur *TListView*.

Document class (*TFileDocument* par défaut) désigne la classe du document par défaut (disponible si l'option Document/view est active).

Description décrit la classe des fichiers associés à Document/view. La valeur par défaut est "All Files (*.*)".

Filters (*.* par défaut) énumère, séparés par des points-virgules ou des virgules, les masques de fichier avec caractères génériques précisant le nom des fichiers que l'application doit reconnaître. Cette valeur est transmise aux boîtes de dialogue des fichiers courants de Windows pour filtrer les fichiers affichés dans ces boîtes.

Default extension précise l'extension par défaut du nom des fichiers. Cette valeur est transmise aux boîtes de dialogue des fichiers courants pour être ajoutée au nom des fichiers quand aucune extension n'est donnée. L'extension par défaut est utilisée dans les boîtes de dialogue de File | Open et File | New.

**Main Window|MDI
Client**

MDI Client décrit la classe qui définit la fenêtre cliente de la fenêtre principale en mode Multiple Document Interface (disponible si MDI est sélectionné au niveau d'Application Model).

Client class précise le nom utilisé par AppExpert pour la classe dérivée de TMDIClient correspondant à la zone cliente de la fenêtre MDI.

Source file désigne le fichier source dans lequel est enregistrée l'implémentation de la classe indiquée dans Client Class.

Header file désigne le fichier d'en-têtes dans lequel est enregistrée la définition de la classe indiquée dans Client Class.

Options du menu MDI Child/View

Les options de MDI Child/View définissent la classe de la fenêtre fille ou de Document/view (disponible si MDI et Document/view ont été sélectionnés dans Application Model).

MDI child désigne la classe dérivée de TMDIChild correspondant au format des fenêtres filles de MDI par défaut.

Source file désigne le fichier source dans lequel est enregistrée l'implémentation de la classe indiquée dans MDI child.

Header file désigne le fichier d'en-têtes dans lequel est enregistrée la définition de la classe indiquée dans MDI child.

**MDI
Child/View|Basic
Options**

Les options Basic définissent la fenêtre fille MDI par défaut.

MDI client/view class désigne la classe d'affichage MDI par défaut. Le décodage de cette valeur diffère selon que vous avez ou non sélectionné l'option Document/View lors de la sélection des paramètres du menu Application :

Tableau 4.2
MDI client/view class
avec Document/view

Document/view actif	Document/view inactif
<i>TEditView</i> (par défaut)	<i>TEditFile</i> (par défaut)
<i>TListView</i>	<i>TListBox</i>
<i>TWindowView</i>	<i>TWindow</i>

Cette valeur est automatiquement mise en correspondance avec les valeurs sélectionnées pour Document/view. Par exemple, si vous désactivez l'option Document/view, *TListView* bascule sur *TListBox*. Inversement, si vous activez l'option Document/view, *TListBox* bascule sur *TListView*.

Document class désigne la classe du document dans l'option Document/view par défaut (*TFileDocument* par défaut).

Description décrit la classe de fichier associée à l'option Document/view. La valeur par défaut est "All Files (*.*)".

Filters (*.* par défaut) énumère, séparés par des points-virgules ou des virgules, les masques de fichier avec caractères génériques précisant le noms de fichiers qui doivent être reconnus par l'application. Cette valeur est transmise aux boîtes de dialogue des fichiers courants de Windows pour filtrer les fichiers affichés à l'intérieur de ces boîtes.

Default extension précise l'extension par défaut des noms de fichiers ; elle est transmise aux boîtes de dialogue des fichiers courants de Windows et sera ajoutée au nom des fichiers dont l'extension n'a pas été précisée.

Utilisation de ClassExpert

ClassExpert affiche les fonctions et événements virtuels pour les classes existantes et vérifie celles qui sont implémentées dans votre application.

ClassExpert permet de créer de nouvelles classes, de modifier et redéfinir l'implémentation des classes, et d'explorer le code source pour savoir quelles sont les classes de vos applications AppExpert. En adjoignant Resource Workshop à ClassExpert, vous pouvez associer des classes à des ressources (par exemple associer une classe *TDialog* à une ressource de dialogue).

Lancement de ClassExpert

Pour lancer ClassExpert,

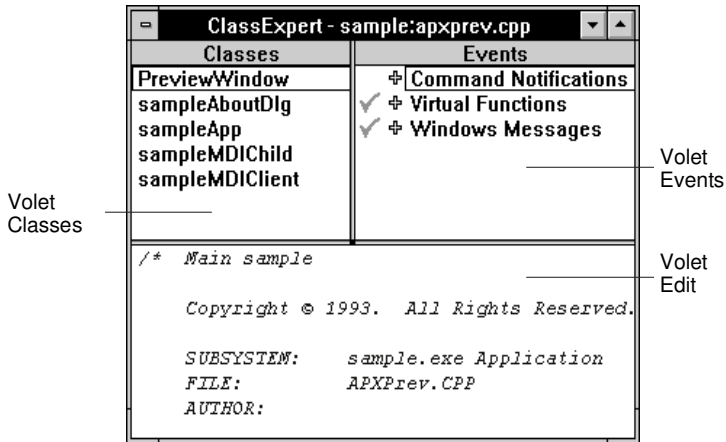
1. Ouvrez un fichier de projet AppExpert en sélectionnant Project | Open project.
2. Double-cliquez sur le noeud de l'application AppExpert (ClassExpert est le gestionnaire de visualisation par défaut des applications AppExpert), ou sélectionnez View | ClassExpert, ou encore cliquez sur le bouton de la barre d'icônes affichée à gauche. ClassExpert apparaît avec la liste des classes et leur implémentation dans votre application.



ClassExpert : notions de base

Cette section décrit les trois volets de ClassExpert et leur fonctions. Vous pouvez régler la taille des volets en tirant sur leur cadre. Quand vous modifiez la taille de ClassExpert, les volets gardent leurs proportions relatives.

Figure 5.1
Fenêtre ClassExpert



Volet Classes

Le volet Classes énumère les classes gérées par ClassExpert pour l'application en cours. Les informations affichées dans les volets Events et Edit dépendent de la classe sélectionnée dans ce volet. Vous pouvez double-cliquer sur une classe pour accéder au code source du constructeur de la classe qui s'affichera dans le volet Edit. A l'aide du TurboMenu (cliquez avec le bouton droit dans le volet Classes), vous pouvez ajouter des classes, associer des classes de documents à des classes de visualisation, obtenir des renseignements sur une classe, accéder au code source ou au fichier d'en-têtes de la classe, modifier la classe et lancer Resource Workshop (en sélectionnant Edit dialog ou Edit menu).

Volet Events

Le volet Events énumère les événements et fonctions virtuelles de la classe de base de la classe sélectionnée dans le volet Classes. Les informations affichées dans le volet Events dépendent du type de classe de base choisi.

Grâce au TurboMenu d'Events (cliquez avec le bouton droit dans le volet Events), vous pouvez ajouter ou supprimer des gestionnaires de messages et des variables d'instance.

Volet Edit

Le volet Edit est un éditeur qui affiche le code source des articles sélectionnés dans le volet Classes et dans le volet Events. Le volet Edit a la même fonction qu'une fenêtre d'éditeur EDI : si vous modifiez ou mettez à jour les options de l'éditeur EDI, celles-ci sont immédiatement disponibles dans le volet Edit de ClassExpert.

Le volet Edit utilise le menu principal de l'EDI et possède un TurboMenu accessible en cliquant avec le bouton droit dans le volet Edit. Le volet Edit fonctionne exactement comme une fenêtre d'éditeur EDI, à ceci près que vous ne pouvez pas scinder les volets ou ouvrir d'autres fichiers à l'intérieur du volet Edit.

Ajouter une classe

ClassExpert vous permet d'ajouter des classes du type ObjectWindows et peut prendre en charge un niveau d'héritage (vous pouvez ajouter manuellement d'autres dérivations).

Pour ajouter une classe,

1. Cliquez avec le bouton droit dans le volet Classes. Le TurboMenu apparaît.
2. Sélectionnez Create new class ou cliquez sur le bouton de la barre d'icônes affiché à gauche. La boîte de dialogue Add New Class apparaît.
3. Sélectionnez la classe de base ObjectWindows à partir de laquelle vous voulez dériver la classe. Appuyez sur *Tab*.
4. Tapez le nom que vous voulez donner à la nouvelle classe. Appuyez sur *Tab*.
5. Tapez le nom du fichier source dans lequel vous voulez voir figurer le code source. Le fichier est sauvegardé dans le chemin source du projet. Appuyez sur *Tab*.
6. Tapez le nom du fichier d'en-têtes qui définit la classe. Par défaut ce fichier prend le nom du fichier source, mais avec l'extension *.H*. Appuyez sur *Tab*.
7. Les sélections qui suivent dépendent de la classe de base :
 - Si la classe de base est *TDialog*, vous devez indiquer ou sélectionner un ID de modèle de dialogue. La boîte Dialog ID list affiche les ID de toutes les ressources de dialogue de votre application AppExpert. Si vous indiquez un ID qui n'existe pas, AppExpert crée une boîte de dialogue vide contenant l'ID indiqué (par souci de cohérence vous pouvez utiliser le préfixe *IDD_*), puis Resource Workshop se charge pour vous permettre de définir la boîte de dialogue.
 - Si la classe de base est *TFrameWindow* ou une classe dérivée de *TFrameWindow*, vous pouvez choisir une classe existante et affichée dans la boîte Client class list pour représenter la zone cliente de la nouvelle fenêtre principale de l'application.



- Si la classe de base est *TWindow* ou une classe dérivée de *TWindow*, cliquez sur le bouton Set Window Properties. Une boîte de dialogue apparaît dans laquelle vous pouvez définir les propriétés de la fenêtre, notamment sa couleur, son cadre et sa légende. Reportez-vous à l'aide en ligne (cliquez sur le bouton Help) pour de plus amples renseignements sur chaque propriété.

8. Cliquez sur OK pour ajouter la nouvelle classe.

Créer des types de documents

Quand vous créez une application AppExpert prenant en charge document/view, vous pouvez utiliser ClassExpert pour créer des classes de visualisation et des types de documents.

Pour créer un type de document,

1. Créez une classe de visualisation, à moins que vous ne vouliez utiliser l'une des trois classes de visualisation prédéfinies (*TEditView*, *TListView*, ou *TWindowView*).
2. Lancez ClassExpert à partir de votre projet. Cliquez avec le bouton droit dans le volet Classes, puis sélectionnez Create doc types dans le TurboMenu.
3. Sélectionnez une classe de visualisation (si vous avez créé votre propre classe, elle figure dans la liste). Les classes de visualisation par défaut sont les suivantes :

- *TEditView* fournit un visualisateur de textes formatés pour la classe d'édition de texte d'ObjectWindows.

- *TListView* permet la visualisation de boîtes d'énumération.

- *TWindowView* permet la visualisation dans des fenêtres.

4. Tapez un descriptif des types de fichiers que votre type de document acceptera. Ce texte apparaît dans la boîte de dialogue File | Open.
5. Tapez le nom des filtres que vous voulez utiliser en les séparant par des virgules ; ces filtres sont affichés dans la boîte de dialogue File | Open et permettent de filtrer tous les fichiers qui peuvent être ouverts par un utilisateur et utilisés dans votre application. Par exemple, si vous créez un type de document pour des images en mode points, vous pourriez avoir un filtre * .BMP.
6. Tapez une extension par défaut pour votre application ; elle sera utile quand vous sauvegarderez des fichiers.

Voir la documentation d'ObjectWindows pour toute information complémentaire sur ces classes.

7. Cliquez sur le bouton Style pour définir les styles de document/view. Les choix possibles sont les suivants (pour plus de renseignements, reportez-vous à la documentation d'ObjectWindows) :
 - *dtAutoDelete* supprime l'objet document quand le dernier écran est fermé.
 - *dtNoAutoView* ne crée pas automatiquement un type d'écran par défaut.
 - *dtSingleView* ne donne qu'un écran pour chaque document.
 - *dtAutoOpen* ouvre un document quand il est créé.
 - *dtUpdateDir* met à jour le répertoire avec le répertoire des boîtes de dialogue.
 - *dtHidden* n'affiche pas le modèle dans la liste des sélections utilisateur.
 - *dtSelected* indique le dernier modèle sélectionné.
 - *dtReadOnly* coche la case à cocher "lecture seule" quand la boîte de dialogue est créée.
 - *dtOverWritePrompt* demande à l'utilisateur s'il est d'accord pour écraser un fichier existant quand il utilise la boîte de dialogue Save As.
 - *dtHideReadOnly* cache la case "lecture seule".
 - *dtPathMustExist* n'autorise l'utilisateur à taper que des chemins existants.
 - *dtFileMustExist* n'autorise l'utilisateur à taper que des noms de fichiers existants.
 - *dtCreatePrompt* demande confirmation à l'utilisateur avant de créer un nouveau document.
 - *dtNoReadOnly* restitue le fichier indiqué sans protection contre l'écriture.
8. Cliquez sur Add pour inclure le type de document dans votre application (cette opération met à jour une structure des données du fichier source principal qui décrit tous les types de documents disponibles). Document/view apparaît dans la liste des types existants.
9. Reprenez les étapes 1 à 8 pour chaque type de document que vous voulez créer. Quand vous avez fini, cliquez sur Close pour revenir à ClassExpert.



Pour ajouter un gestionnaire destiné à un événement donné,

1. Sélectionnez la classe du gestionnaire de messages. Les événements s'affichent dans le volet Events.
2. Sélectionnez l'événement à traiter (il se peut que vous soyez obligé de développer la liste des événements), puis cliquez avec le bouton droit sur l'événement pour afficher le TurboMenu.
3. Choisissez Add handler dans le TurboMenu. Si vous choisissez d'ajouter un gestionnaire pour un message Windows, ClassExpert ajoute une entrée à la table de recherche dont le nom est défini par défaut, puis la fonction associée au gestionnaire apparaît dans la fenêtre d'édition. D'autres gestionnaires, par exemple des commandes, vous demandent d'indiquer le nom de la fonction avant d'ajouter l'entrée dans la table de recherche.
4. ClassExpert affiche une coche à côté de l'événement dans le volet Events pour vous montrer que l'événement est pris en charge. Une marque d'un gris plus clair indique que certains événements à l'intérieur de la catégorie concernée sont pris en charge (développez la liste pour visualiser ces événements).

Pour supprimer le gestionnaire d'un événement,



1. Sélectionnez la classe du gestionnaire de messages. Les événements s'affichent dans le volet Events.
2. Sélectionnez l'événement coché avec le gestionnaire que vous voulez supprimer (il vous faudra peut-être développer la liste des événements), puis cliquez avec le bouton droit sur l'événement pour afficher le TurboMenu.
3. Sélectionnez Delete handler. ClassExpert supprime uniquement l'entrée de la table de recherche, pas le code du fichier source. Le code du gestionnaire apparaît dans le volet Edit, ce qui vous permet de le supprimer. Si vous supprimez la fonction, supprimez sa définition dans le fichier des en-têtes (vous pouvez sélectionner Edit header dans le TurboMenu du volet Classes pour visualiser le fichier).

Ajout et suppression de variables d'instance

Les variables d'instance facilitent la gestion de nombreux contrôles. Quand vous créez des variables d'instance, ClassExpert ajoute un tampon de transfert dans votre code. Ce tampon collecte les informations au moment de l'exécution, informations que vous pouvez utiliser au lieu de créer un code pour vérifier si toutes les cases sont cochées. Par exemple, si vous avez une boîte de dialogue avec six cases à cocher et si vous voulez que votre application fasse quelque chose en fonction des cases cochées, vous pouvez utiliser des variables d'instance pour chaque contrôle puis les données du tampon de transfert de votre code. Reportez-vous au *Guide du programmeur ObjectWindows* pour tout renseignement complémentaire sur les tampons de transfert.

Pour ajouter (associer) une variable d'instance à un contrôle,



1. Sélectionnez le contrôle dans le volet Events (vous serez peut-être obligé de développer la liste des événements pour visualiser les contrôles).
2. Cliquez avec le bouton droit sur le contrôle et sélectionnez Add Instance variable.
3. Dans la boîte de dialogue Add Instance variable tapez le nom de la variable. Cliquez sur OK. ClassExpert ajoute ce qui suit au code de votre application :
 - Dans le fichier d'en-têtes, il ajoute une déclaration de structure avec une entrée pour la variable d'instance.
 - Dans le constructeur de classes du fichier source .CPP, la variable est affectée (pour associer la classe ObjectWindows à l'objet ressource).
 - Dans le fichier .CCP, un exemple statique de la structure de transfert est déclaré.
4. L'étiquette de contrôle du volet Events indique la classe et le nom de la variable d'instance que vous venez de créer.

Pour supprimer une variable d'instance,



1. Sélectionnez le contrôle accompagné de la variable d'instance que vous voulez supprimer.
2. Cliquez avec le bouton droit sur le contrôle et sélectionnez Delete Instance variable.

3. ClassExpert supprime de votre code ce qui suit :
 - L'entrée de la structure
 - La variable de pointeur dans la déclaration de classe
 - L'affectation de la variable de classe associée au contrôle ressource dans le constructeur

Si vous supprimez toutes les variables d'exemple de votre code, il restera une structure vide et la convention d'appel du tampon de transfert définie. Ces informations n'ayant aucune incidence sur le reste du code, vous n'avez pas besoin de les supprimer manuellement.

**Accès au code
source des classes**

Pour visualiser le code source d'une classe, sélectionnez cette classe dans le volet Classes (cliquez une fois sur son nom). Le code apparaît dans le volet Edit. Si vous déplacez le curseur dans le volet Edit, ClassExpert mémorise sa position et y revient dès que vous sélectionnez à nouveau cette classe.

Pour accéder au code du constructeur de la classe, double-cliquez sur le nom de la classe dans le volet Classes. Pour accéder à un événement géré, double-cliquez sur l'événement concerné dans le volet Events. Vous pouvez aussi visualiser le fichier source ou son fichier d'en-têtes dans un éditeur EDI :

1. Sélectionnez la classe dans le volet Classes.
2. Cliquez avec le bouton droit sur la classe. Un TurboMenu apparaît.
3. Choisissez Edit source pour visualiser le fichier source du constructeur de classes (fichier .CPP), ou choisissez Edit header pour visualiser le fichier d'en-têtes dans lequel la classe est définie.

Utilisation de Resource Workshop avec ClassExpert

Resource Workshop est le gestionnaire de visualisation des scripts ressources (fichiers .RC) de ClassExpert . Quand vous lancez Resource Workshop à partir de ClassExpert (en cliquant avec le bouton droit sur une classe et en sélectionnant Edit dialog ou Edit menu), Resource Workshop charge automatiquement le script RC de cette application.

Quand vous utilisez Resource Workshop avec un code généré par AppExpert, faites-le tourner à partir de ClassExpert car Resource Workshop et ClassExpert se mettent mutuellement à jour chaque fois que vous apportez des modifications au projet. Dès qu'il est lancé, Resource Workshop vérifie le code ressource pour détecter toute modification éventuelle, puis envoie immédiatement des mises à jour à ClassExpert. Par exemple, si vous ajoutez un bouton dans une boîte de dialogue, Resource Workshop en informe ClassExpert, qui ajoute le contrôle correspondant dans le volet Events. Pour visualiser ce contrôle dans ClassExpert, sélectionnez-le dans Resource Workshop, cliquez avec le bouton droit puis sélectionnez ClassExpert dans le TurboMenu. Resource Workshop vous renvoie ClassExpert avec le contrôle en surbrillance dans le volet Events.

**Exécution de
Resource
Workshop à partir
de l'EDI**

Quand vous lancez Resource Workshop comme gestionnaire de visualisation d'une application AppExpert (par l'intermédiaire soit du gestionnaire de projet de l'EDI, soit de ClassExpert), il n'a pas le même comportement que quand il tourne séparément :

- Quand vous effectuez dans Resource Workshop des modifications qui ont une incidence sur la structure des classes ou leurs fonctions (par exemple les menus d'édition ou les boîtes de dialogue), ces modifications sont instantanément mises à jour dans la fenêtre ClassExpert.
- Vous ne pouvez pas ouvrir un autre script (ni File | Open ni File | New).
- Si vous fermez l'EDI, Resource Workshop est lui aussi fermé et toutes les modifications que vous avez apportées sont automatiquement sauvegardées.
- Si vous fermez le fichier de projet AppExpert à partir duquel vous avez lancé Resource Workshop, vous fermez du même coup Resource Workshop.
- Si vous créez un projet alors que Resource Workshop est ouvert, ce dernier crée un fichier .RES basé sur les ressources chargées. Par exemple, si vous modifiez une boîte de dialogue et ne la sauvegardez pas, le fichier .RES conservera les modifications *non sauvegardées*.
- Vous pouvez accéder à l'EDI à partir de Resource Workshop grâce au TurboMenu (cliquez avec le bouton droit), en sélectionnant ClassExpert.

Utilisation de Rescan

Rescan est un outil spécial qui inspecte tous les codes source dont la liste figure dans votre projet AppExpert (fichier .IDE), pour mettre à jour et reconstituer la base de données du projet (fichier .APX) en fonction de ce qu'il trouve dans le code source. Rescan recherche les marques spéciales dans le code source pour reconstituer le fichier de base de données d'AppExpert, puis lance Resource Workshop pour reconstituer les informations relatives aux ressources du projet. Si la relecture se déroule normalement, le fichier de base de données initial du projet est renommé *.~AP et un nouveau fichier de base de données créé ; sinon, la base de données d'origine reste *.APX.

Vous pouvez utiliser Rescan pour :

- Supprimer une classe
- Déplacer une classe d'un fichier source à un autre
- Renommer une classe, un gestionnaire, une variable d'instance ou un ID de dialogue
- Importer une classe d'un autre projet AppExpert
- Reconstituer un fichier de base de données de projet (*.APX) perdu ou endommagé

Supprimer une classe

Pour supprimer une classe,

1. Retirez le fichier source de la classe du projet EDI en sélectionnant le noeud source, en cliquant avec le bouton droit et en sélectionnant Delete node. Si la classe partage un fichier source avec d'autres classes, supprimez dans le fichier source le code de la classe concernée, ainsi que tout référence à cette classe dans les autres fichiers source.
2. Sélectionnez l'application AppExpert dans le projet, cliquez avec le bouton droit, puis sélectionnez Special | Rescan. Rescan explore les fichiers source mentionnés comme dépendances de l'application AppExpert. Resource Workshop explore et met à jour les fichiers de ressources. Quand Rescan a terminé, vous revenez au fichier de projet – mis à jour – à partir duquel vous pourrez soit créer votre application, soit utiliser ClassExpert. Vous pouvez ajouter au projet la classe supprimée en ajoutant le fichier source de la classe et en le liant à l'application AppExpert, puis en exécutant une nouvelle relecture.

Déplacer une classe

Pour déplacer une classe d'un fichier source à un autre,

1. Déplacez (par couper-coller) le code source de la classe pour l'intégrer dans le nouveau fichier. Si le nouveau fichier ne figure pas dans le projet en tant que noeud de l'application AppExpert, ajoutez-le (reportez-vous au chapitre 2). Si la classe déplacée était son propre fichier source, vous pouvez supprimer du projet ce fichier source, désormais vide.
2. Sélectionnez l'application AppExpert dans le projet, cliquez avec le bouton droit pour afficher le TurboMenu, puis sélectionnez Special | Rescan. Lorsque Rescan a terminé, la fenêtre de projet EDI s'affiche à nouveau.

Renommer un élément d'AppExpert

Pour renommer une classe, une fonction de gestion d'événement, une variable d'instance ou un ID de dialogue,

1. Utilisez l'éditeur EDI pour rechercher et remplacer partout le nom d'origine par le nouveau nom. Veillez à vérifier tous les fichiers source liés au projet (fichiers .CPP et .H).
2. Dans la fenêtre du projet, sélectionnez l'application AppExpert, cliquez avec le bouton droit, puis sélectionnez Special | Rescan. Quand Rescan a terminé, la fenêtre du projet EDI s'affiche à nouveau.

Importer une classe

Pour importer une classe d'un projet AppExpert dans un autre,

1. Déplacez ou copiez le fichier source et le fichier d'en-têtes qui définissent la classe dans le répertoire source et d'en-têtes de l'autre projet. Tous les fichiers source d'un projet doivent se trouver dans le répertoire source (fichiers .CPP) ou dans le répertoire d'en-têtes (fichiers .H) de ce projet. Ces répertoires ont été créés quand vous avez généré pour la première fois le projet AppExpert.
2. Ajoutez le fichier source de classe sous forme de noeud lié à l'application AppExpert dans le projet EDI (à l'aide de l'option Add node du TurboMenu).
3. Dans la fenêtre du projet, sélectionnez l'application AppExpert, cliquez dessus avec le bouton droit, puis sélectionnez Special | Rescan.

**Reconstitution du
fichier de base de
données .APX**

Pour reconstituer un fichier de base de données perdu ou endommagé (le fichier .APX),

1. Ouvrez le fichier de projet qui contient l'application AppExpert et les noeuds dépendants (dans le fichier .IDE).
2. Sélectionnez l'application AppExpert, cliquez avec le bouton droit, puis sélectionnez Special | Rescan dans le TurboMenu. Rescan crée automatiquement un nouveau fichier de base de données en utilisant les marques du code source de l'application AppExpert .

Débogage dans l'EDI

Même si vous prenez toutes vos précautions lorsque vous entrez votre code, votre programme risque de contenir des erreurs (des bugs). Le débogage est l'opération qui consiste à localiser et à corriger ces bugs. Ce chapitre explique comment procéder.

Types de bugs

Il existe trois types de bugs : les erreurs à la compilation, les erreurs à l'exécution et les erreurs de logique.

Erreurs à la compilation

Ces erreurs, aussi appelées erreurs de syntaxe, se produisent lorsque votre code enfreint la syntaxe C ou C++, auquel cas l'EDI ne peut compiler votre programme et ouvre une fenêtre de message contenant toutes les erreurs et les avertissements.

Vous devrez corriger les erreurs l'une après l'autre. Pour ce faire, double-cliquez dessus ; le curseur se place automatiquement sur la ligne erronée. Répétez cette opération autant de fois que nécessaire.

Les avertissements affichés dans la fenêtre de message n'empêchent pas la compilation de votre programme, mais ils attirent votre attention sur du code qui peut poser problème (code non portable, non optimisé ou non conforme au standard ANSI). Vous pouvez choisir le type d'avertissements que vous désirez voir :

1. Choisissez Options | Project et double-cliquez sur Messages.
L'EDI affiche les diverses catégories de messages et, à droite, les options correspondantes.

2. Sélectionnez une option déterminant quels avertissements sont affichés par l'EDI.
 - Cochez All pour afficher tous les messages.
 - Cochez Sélectionnez pour n'afficher que les avertissements sélectionnés (voir étape 4)
 - Cochez None pour n'afficher aucun message.
3. Limitez le nombre de messages d'erreurs et avertissements affichés dans la fenêtre Message. Dans les cases Stop After, indiquez le nombre de messages maximum d'avertissements et d'erreurs (de 0 à 255) affichés lors de la compilation.

Si vous entrez 0 dans les deux cases, aucune limite n'est imposée.
4. Dans la zone Topics, choisissez une catégorie de messages.
5. A droite, sélectionnez les messages à afficher dans l'EDI.

Par exemple, si vous désirez que le compilateur vous avertisse des conversions de pointeurs risquant de n'être pas conforme au standard ANSI, choisissez ANSI Violations et cochez Suspicious Pointer Conversion.

Les erreurs à la compilation les plus fréquentes sont les fautes de frappe, les oublis de points-virgules, les références à des variables non déclarées, les passages d'arguments erronés aux fonctions et les mauvaises assignations de valeurs aux variables.

Relancez la compilation après avoir corrigé vos erreurs. Si tout se passe bien, vous pouvez rechercher les éventuelles erreurs à l'exécution et de logique.

Erreurs à l'exécution

Si votre programme se compile bien mais ne s'exécute pas, il s'agit d'une erreur à l'exécution : les instructions sont correctes, mais elles ne peuvent pas s'exécuter (demande d'ouverture d'un fichier introuvable ou division par zéro, par exemple). Le système d'exploitation détecte ces erreurs et stoppe l'exécution du programme.

Erreurs de logique

Les erreurs de logique sont des erreurs de conception ou d'implémentation. Vos instructions sont correctes et déclenchent bien une action, mais pas celle que vous aviez prévue. Ces erreurs sont les plus délicates à corriger, car l'EDI ne sait pas les détecter automatiquement. Heureusement, l'EDI intègre des outils de débogage qui vous aideront dans cette tâche.

Les erreurs de logique se produisent lorsque des variables ont une valeur incorrecte ou inattendue, lorsque des images n'ont pas l'aspect voulu ou lorsque du code ne s'exécute pas quand il le devrait. Les pages qui suivent expliquent comment corriger ces erreurs.

Création d'informations de débogage

Vous devez compiler et éditer les liens de votre programme pour que des informations de débogage soient générées dans vos EXE et vos OBJ.

- Pour ajouter des informations de débogage à vos OBJ, choisissez Options | Project et sélectionnez Compiler | Debugging | Debug Information in OBJs dans la boîte de dialogue qui s'ouvre alors. Cette option est activée par défaut.
- Pour ajouter des informations de débogage à vos EXE, choisissez Options | Project et sélectionnez Linker | General | Include Debug Information dans la boîte de dialogue qui s'ouvre alors. Cette option est activée par défaut.

Dès que vous compilez un programme, le compilateur génère une table des identificateurs utilisés et la stocke dans le fichier exécutable. Cette liste, appelée la table des symboles, est utilisée par le débogueur pour tracer les variables, constantes, types, fonctions et instructions de votre programme.

Indication des arguments du programme

Si le programme à déboguer requiert le passage d'arguments, vous devez les spécifier :

1. Choisissez Options | Environment et sélectionnez un sujet pour le débogueur.
2. Dans la case Run Arguments, entrez les arguments que vous désirez passer au programme.

Contrôle de l'exécution du programme

Le contrôle de l'exécution du programme est la partie la plus importante du débogage. Etant donné qu'il est possible de contrôler l'exécution de chaque instruction, il est aisé de savoir quelle partie du programme pose un problème.

Le mode pas à pas ("stepping") et le tracé ("tracing") permettent l'exécution d'un programme instruction par instruction. Vous pouvez alors étudier l'état du programme et des données, vérifier la sortie générée par le programme et la valeur des variables, ou modifier ou évaluer des expressions avant de passer à l'instruction suivante.

Vérification de la sortie générée par le programme

Lorsque vous exécutez votre programme en mode pas à pas, vous pouvez étudier la sortie de votre application dans sa propre fenêtre. Configurez votre écran pour qu'il affiche à la fois votre code source et la fenêtre de votre application. Si la fenêtre du bureau de l'EDI et celle de votre application se chevauchent lorsque vous déboguez, vous obtiendrez une image instable.

Mode pas à pas

Toute exécution dans le débogueur, y compris le mode pas à pas, le tracé et le respect des points d'arrêt, est liée aux lignes du code source. Même si une instruction tient sur plusieurs lignes à l'écran, elle est vue comme une seule ligne de code.

Vous pouvez contrôler la quantité de débogage au niveau de l'unité de ligne de code. Si vous placez plusieurs instructions sur une seule ligne, vous ne pourrez pas déboguer une seule de ces instructions. Par contre, vous pouvez répartir une instruction sur plusieurs lignes, et voir cette instruction s'exécuter en une seule fois.

Lorsque vous demandez au débogueur de passer en pas à pas ou en tracé, le point d'exécution (la ligne en surbrillance dans votre code) passe à la ligne suivante. Le point d'exécution marque toujours l'instruction qui sera exécutée après la présente.



Le mode pas à pas est la façon la plus simple de vous déplacer à votre rythme dans le code. Pour ce faire, choisissez Debug | Step Over (ou F8, ou le bouton Pas à pas dans la barre d'icônes). Le code en surbrillance s'exécute, y compris toute fonction qu'il doit appeler avant de vous rendre la main. Le point d'exécution passe ensuite sur la ligne suivante.

L'exemple ci-dessous vous aidera à comprendre comment fonctionne le mode pas à pas. Voici les premières lignes d'un programme chargé dans la fenêtre d'édition :

```
...
BOOL InitApplication ( HINSTANCE hInstance )
{
    WNDCLASS wc;

    wc.style          = CS_HREDRAW I CS_VREDRAW;
    wc.lpfnWndProc    = (long (FAR PASCAL*) ()) MainWndProc;
    wc.cbClsExtra     = 0;
    wc.cbWndExtra     = 0;
    ...
    return ( RegisterClass ( &wc) );
}
...

int PASCAL WinMain ( HINSTANCE hInstance, HINSTANCE
hPrevInstance,
LPSTR lpCmdLine, int nCmdShow)
{
    MSG msg;

    if ( !hPrevInstance )
    if ( !InitApplciation ( hInstance ) )
        return ( FALSE );

    if ( !InitInstance ( hInstance, nCmdShow ) )
        return ( FALSE );
    ...
}
```

Dans cet exemple, **InitApplication** est une fonction définie dans un module que vous avez compilé avec les informations de débogage. Si vous déboguez réellement ce programme, à chaque fois que vous choisiriez Debug | Step Over (ou appuieriez sur *F8*), le débogueur exécuterait la ligne en surbrillance, puis le point d'exécution passerait sur la ligne suivante. Si vous appuyiez sur *F8* lorsque le point d'exécution est sur l'instruction suivante,

```
if ( !InitApplication ( hInstance ) )
```

le débogueur exécuterait la fonction **InitApplication** et retournerait une valeur booléenne, mais vous ne verriez pas le point d'exécution passer sur l'instruction **InitApplication** elle-même : le débogueur passerait au-dessus de la fonction. La main vous serait rendue après l'exécution de cette fonction.

Tracé du code en mode approfondi



Dans ce mode, à la différence du mode pas à pas, lorsque vous arrivez sur une ligne qui appelle une fonction, le point d'exécution se place dans le code de la fonction. Dans l'exemple ci-dessus, si vous choisissez Debug | Trace Into (ou appuyez sur *F7* ou cliquez sur le bouton Tracé en mode approfondi de la barre d'outils) pour exécuter chaque instruction, vous voyez le point d'exécution se place sur le code implémentant la fonction **InitApplication** lorsque le débogueur atteint instructions évaluant la valeur de retour de **InitApplication**. Lorsque vous déboguez, vous pouvez choisir librement l'un ou l'autre de ces deux modes d'analyse de votre code source.

Mode pas à pas et mode approfondi appliqués aux fonctions membres d'une classe

Même si vous utilisez les classes dans votre programme, vous pouvez utiliser le débogueur intégré dans ces deux modes. Il gère les fonctions membres de la même façon qu'il gère les fonctions dans les programmes non orientés objets.

Mode pas à pas et mode approfondi appliqués au code externe

Si vous liez du code externe à votre programme, vous pouvez le déboguer dans ces deux modes si le code du fichier OBJ lié contient des informations de débogage.

Il est possible de déboguer du code externe écrit en tout langage, y compris C, C++, Pascal et assembleur. Si le code est conforme aux normes de linkage externe et contient des informations symboliques de débogage complètes aux normes Borland, le débogueur intégré peut le traiter dans les deux modes.

Contourner une partie du code

Parfois, pour atteindre directement le source qui vous semble erroné, vous souhaitez contourner une partie de votre code. Le débogueur intégré vous permet ceci, et vous redonne la main à l'emplacement de votre choix.

Recherche d'une zone précise

Vous pouvez atteindre directement une partie précise de votre code. Procédez comme suit :

1. Placez le curseur sur la ligne où vous désirez reprendre le débogage.
2. Choisissez Run to Cursor dans le TurboMenu, ou appuyez sur F4.

Vous pouvez faire ceci au début du débogage ou en cours de session.

Recherche d'une fonction

La commande Locate Function du menu Search vous permet d'atteindre rapidement une fonction précise. Pour l'utiliser, vous devez être en mode débogage.

Retour au point d'exécution

Lorsque vous déboguez, vous pouvez vous déplacer librement dans les fichiers ouverts, ouvrir ou fermer des fichiers, puis revenir rapidement au point d'exécution.

Pour ce faire, choisissez Debug | Find Point d'exécution.

Si la fenêtre contenant le point d'exécution a été fermée, le fichier est automatiquement réouvert.

Remonter dans le code

Il est parfois utile, lorsque vous déboguez, de savoir comment vous êtes parvenu à la section de votre code source dans laquelle vous vous trouvez. La fenêtre Call Stack vous indique la suite de fonctions qui vous a conduit là où vous êtes. Choisissez View | Call Stack pour afficher cette fenêtre.

La fenêtre Call Stack est très utile si vous arrivez par erreur dans une zone de code que vous désirez contourner. Vous pourrez ainsi revenir en arrière puis reprendre le tracé à l'endroit de votre choix :

1. Dans la fenêtre Call Stack, double-cliquez l'appel ayant lancé la fonction sur laquelle vous êtes arrivé par erreur ; il s'agit du second appel en partant du sommet de la fenêtre.
La fenêtre d'édition devient la fenêtre active ; le curseur est placé à l'endroit où l'appel a eu lieu.
2. Dans la fenêtre d'édition, placez le curseur au-dessous de l'appel.
3. Choisissez Run to Cursor dans le TurboMenu de la fenêtre d'édition.

La fenêtre Call Stack est aussi très utile pour visualiser les arguments passés à vos fonctions.

Pour voir ou éditer le code source contenant un appel précis, sélectionnez l'appel en question dans la fenêtre Call Stack et cliquez (avec le bouton droit de la souris) pour afficher le TurboMenu avec ses commandes de visualisation et d'édition.

Arrêt du programme

Pour arrêter votre programme, vous pouvez utiliser une autre méthode que les deux que nous venons d'étudier : il suffit de choisir Debug | Pause Program. Vous pouvez alors examiner la valeur des variables et visualiser les données de votre programme. Pour le relancer, choisissez Debug | Run.

Si votre programme refuse de vous rendre la main (s'il est pris dans une boucle sans fin, par exemple), appuyez sur *Ctrl+Alt+Syst* pour l'arrêter.

Recommencer depuis le début

Terminate Program (commandeSi, en cours de débogage, vous décidez de recommencer depuis le début de votre programme, choisissez Debug | Terminate Program ou appuyez sur *Ctrl+F2*.

Visualisation des valeurs

Le tracé du code et le mode pas à pas sont parfaits pour trouver les problèmes dans le déroulement d'un programme, mais il est aussi important de savoir quelle valeur prennent vos variables. Par exemple si vous exécutez pas à pas une boucle **for**, il est utile de vérifier l'évolution de la variable d'index. L'EDI offre plusieurs outils qui vous le permettent :

- La fenêtre Watch affiche la valeur d'une variable ou d'une expression.
- La boîte de dialogue Evaluate Expression vous permet d'évaluer toute expression en rapport avec votre programme, et de modifier la valeur d'une variable en cours de débogage.
- La fenêtre Data Inspector vous permet d'évaluer et de modifier la valeur d'une donnée.

Qu'est-ce qu'une expression ?

La visualisation, l'évaluation et l'inspection opèrent au niveau des *expressions*. Une expression consiste en une constante, une variable ou une structure de données associées à des opérateurs. Pratiquement tous les éléments utilisables dans la partie de droite d'une instruction d'assignation peuvent servir expression de débogage.

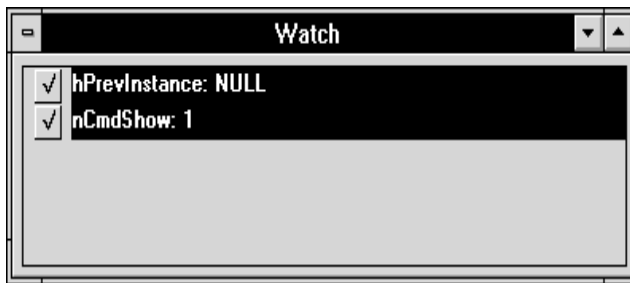
Visualisation des expressions

Si vous désirez suivre la valeur d'une expression ou d'une variable tout en exécutant votre code pas à pas, utilisez un suivi ("watch"). Un suivi est une expression, entrée dans la fenêtre Watch, qui indique la valeur courante de l'expression en reflétant toute modification que lui apporte votre programme.

Si le point d'exécution dépasse la portée d'une expression de suivi, l'expression est indéfinie. Lorsque le point d'exécution retourne dans la portée, la fenêtre Watch réaffiche la valeur de l'expression.

Pour ouvrir la fenêtre de suivi, choisissez View | Watch. Si aucun suivi n'a été défini, la fenêtre est vide.

Figure 6.1
Fenêtre de suivi



Ajout d'un suivi

Pour ajouter une variable dans la fenêtre de suivi, choisissez Debug | Add Watch. L'EDI ouvre la boîte de dialogue Watch Properties, dans laquelle vous entrez l'expression de suivi.

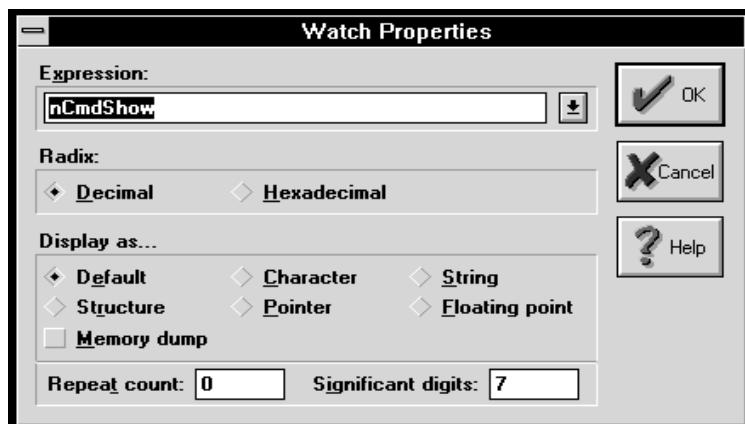
Cette fenêtre, comme toute fenêtre de l'EDI, peut être librement modifiée. Si vous la fermez, vous pourrez la rouvrir en choisissant View | Watch. Si la fenêtre de suivi est la fenêtre active, vous pouvez ajouter un suivi en choisissant Add Watch dans le TurboMenu.

Si la fenêtre d'édition est active, vous pouvez rapidement placer l'expression sous le curseur dans la fenêtre de suivi en choisissant Set Watch dans le TurboMenu.

Formatage des expressions de suivi

Si vous double-cliquez sur un suivi dans la fenêtre de suivi (ou si vous sélectionnez un suivi et choisissez Set Properties dans le TurboMenu de la fenêtre de suivi), la boîte de dialogue Watch Properties s'ouvre.

Figure 6.2
Boîte de dialogue
Watch Properties



L'expression par défaut dans cette boîte de dialogue est le mot sous le curseur dans la fenêtre d'édition courante. Les expressions déjà entrées sont conservées dans un historique.

Vous pouvez formater les résultats en sélectionnant des options dans cette boîte de dialogue. Par exemple, bien que les valeurs entières s'affichent normalement au format décimal, vous pouvez l'afficher en hexadécimal en cliquant sur le bouton correspondant. Vous pouvez aussi modifier le format d'une expression à l'aide des options Display as.

Si vous suivez un tableau ("array"), vous pouvez afficher la valeur des éléments de données consécutifs. Par exemple, si votre tableau compte cinq entiers, tapez 5 dans la case Repeat count de la boîte de dialogue pour voir la valeur des cinq éléments du tableau. Une expression utilisée avec un compteur de répétitions ("Repeat count") doit représenter un seul élément de donnée. Le débogueur considère l'élément de donnée comme le premier élément d'un tableau si l'élément n'est pas un pointeur, et comme un pointeur sur un tableau s'il s'agit effectivement d'un pointeur.

Si vous sélectionnez l'option Floating point, vous pouvez aussi indiquer le nombre de chiffres significatifs à afficher, en utilisant la case Significant digits.

Désactivation d'un suivi

Si vous désirez arrêter momentanément le suivi d'une expression, vous pouvez désactiver un suivi.

Si l'EDI doit évaluer de nombreuses expressions, il se peut que le tracé du code soit ralenti. Pour désactiver momentanément un suivi, cliquez sur la case à cocher à côté du suivi ; pour le réactiver, cliquez à nouveau sur la case à cocher.

Il est également possible d'obtenir le même résultat par le TurboMenu : sélectionnez un ou plusieurs suivis et choisissez la commande appropriée dans le TurboMenu.

Suppression d'un suivi

Pour supprimer une expression suivie, sélectionnez-la et choisissez Delete Watch dans le TurboMenu. Pour supprimer tous les suivis, choisissez Delete All Watches dans le TurboMenu.

Modification d'un suivi

Pour modifier les propriétés d'un suivi, ouvrez la boîte de dialogue Watch Properties de ce suivi et entrez vos modifications.

Evaluation et modification des expressions

Vous pouvez évaluer des expressions à tout moment et modifier la valeur des variables au moment de l'exécution.

Evaluation des expressions

Pour évaluer une expression, choisissez Debug | Evaluate/Modify. La boîte de dialogue Expression Evaluator s'ouvre. Le mot sous le curseur est placé dans la case Expressions. Vous pouvez éditer expression, en entrer une nouvelle ou en choisir une dans l'historique.

La valeur courante de expression sélectionnée est affichée dans la case Result lorsque vous cliquez sur Evaluate. Vous pouvez évaluer toute expression C ou C++ valide, à l'exception de celles qui contiennent les éléments suivants :

- Des symboles ou des macros définis avec **#define**.
- Des variables locales ou statiques en dehors de la portée de la fonction exécutée.
- Des appels de fonctions.

Vous pouvez formater des expressions en ajoutant une virgule et un, ou plusieurs, indicateurs de format. Ainsi, pour afficher un résultat en hexadécimal, tapez ,H après expression. Le tableau 6.1 indique quels sont les indicateurs de format utilisables, ainsi que leur effet.

Tableau 6.1 : Spécificateurs de format des expressions du débogueur

Caractère	Types affectés	Fonction
H ou X	Entiers	Hexadécimal. Les entiers sont précédés du préfixe 0x, y compris ceux qui appartiennent à des structures de données.
C	Caractères, chaînes	Caractère. Affiche de façon spéciale les caractères compris entre 0 et 31. Par défaut, ces caractères sont affichés avec les séquences d'échappement C standards (<code>\n</code> , <code>\t</code> , etc.).
D	Entiers	Décimal. Les entiers sont affichés au format décimal, y compris ceux qui appartiennent à des structures de données.
Fn	Virgule flottante	Virgule flottante. Affiche n chiffres significatifs (où n est compris entre 2 et 18. Valeur par défaut : 7).
nM	Tous	Dump mémoire. Affiche n octets en commençant à l'adresse de expression indiquée. Si n n'est pas spécifié, il prend par défaut la valeur de la taille (en octets) du type de la variable. Par défaut, chaque octet est affiché sous la forme de deux chiffres hexadécimal. Les spécificateurs C, D, H et S peuvent être utilisés avec M pour modifier le formatage des octets.
P	Pointeurs	Pointeur. Affiche les pointeurs sous la forme <i>seg:offset</i> , suivis d'informations sur l'adresse pointée. Indique dans quelle zone de la mémoire le segment est situé et, si utile, le nom de la variable à l'adresse de l'offset.
R	Structures, unions	Structure/Union. Affiche les noms de champs et les valeurs sous la forme X:1;Y:10;Z:5...
S	Char, chaînes	Chaîne. Affiche les caractères ASCII de 0 à 31 sous forme de séquences d'échappement C. A n'utiliser que pour modifier les dumps mémoire (voir nM ci-dessus).

Modification des variables

Vous pouvez modifier la valeur d'une variable à l'aide de la boîte de dialogue Expression Evaluator. Entrez la variable dans la case Expression, puis tapez la nouvelle valeur dans New Value. Si vous désirez que cette nouvelle valeur soit passée à votre programme, cliquez sur Modify. Dans le cas contraire, la valeur est ignorée lorsque vous fermez la boîte de dialogue.

Voici quelques points à toujours avoir à l'esprit lorsque vous modifiez la valeur d'une variable :

- Vous pouvez modifier des variables ou des éléments de tableaux ou de structures, mais non les tableaux, ni les structures elles-mêmes.
- L'expression dans la case New Value doit s'interpréter comme un résultat compatible avec sa variable d'affectation : si l'affectation cause une erreur de compilation ou d'exécution, la modification n'est pas valide.

- Vous ne pouvez pas modifier directement les arguments non typés passés à une fonction, mais vous pouvez les transtyper et leur assigner une nouvelle valeur.
 - Vous pouvez utiliser la boîte de dialogue Expression Evaluator pour examiner et modifier des valeurs dans les registres, y compris les registres de drapeaux. Par exemple, vous pouvez entrer des expressions telles que `:_CS,_BX,_FLAGS`. Ces valeurs sont des masques de bits.
- Attention !
- Soyez très prudent lorsque vous modifiez des valeurs, en particulier des pointeurs et des indices de tableaux : vous risquez d'écraser des variables et des structures de données.

Visualisation des éléments de données

Vous pouvez visualiser et modifier les valeurs d'un élément de donnée dans la fenêtre d'inspecteur. Pour inspecter un élément de données :

1. Choisissez Debug | Inspect pour afficher la fenêtre d'inspecteur de données.
2. Entrez l'expression à inspecter.
3. Choisissez Inspect pour afficher une fenêtre d'inspecteur.

Si le point d'exécution est dans la portée de cette expression, la valeur s'affiche dans la fenêtre. Dans le cas contraire, la valeur est indéfinie.

Vous pouvez aussi afficher une fenêtre d'inspecteur directement depuis la fenêtre d'édition :

1. Placez le curseur sur l'élément à inspecter.
2. Choisissez Inspect dans le TurboMenu ou appuyez sur *Entrée*.

Si vous choisissez cette méthode, l'élément de donnée est toujours évalué dans la portée de la ligne sur laquelle il apparaît.

Dans un cas comme dans l'autre, l'aspect des données dans une fenêtre d'inspecteur dépend du type des données. Si, par exemple, vous inspectez un tableau, vous voyez une ligne pour chacun de ses membres, avec l'indice de tableau du membre. La valeur du membre apparaît ensuite, puis la valeur en hexadécimal.

Lorsque vous êtes dans une fenêtre d'inspecteur, vous pouvez inspecter certains éléments pour restreindre la vue. Pour ce faire :

1. Sélectionnez l'élément à analyser de près.
2. Choisissez Inspect dans le TurboMenu, ou appuyez sur *Entrée*.

Vous pouvez changer la valeur des éléments de l'inspecteur. Pour ce faire :

1. Sélectionnez un élément.
2. Choisissez Change dans le TurboMenu.
3. Entrez sa nouvelle valeur et cliquez sur OK.

Si vous inspectez une structure de données, il se peut que le nombre d'éléments affichés soit si grand que vous soyez obligé de faire défiler la fenêtre d'inspecteur pour voir certaines données. Pour faciliter la lecture, vous pouvez réduire l'affichage à certains éléments.

Pour n'afficher que certains éléments :

1. Cliquez dans la fenêtre d'inspecteur.
2. Choisissez Range dans le TurboMenu.
3. Dans la case Start Index, entrez l'indice de l'élément que vous voulez voir en premier dans la fenêtre.
4. Dans la case Count, entrez le nombre d'éléments à afficher dans la fenêtre d'inspecteur.

Examen des valeurs de registres

Lorsque vous déboguez, vous pouvez afficher la valeur des registres de données, de pointeurs, d'indices, de segments et de pointeurs d'instructions, ainsi que les paramètres du mot ou des drapeaux d'état. Choisissez View | Register pour ouvrir la fenêtre Registers.

Vous pouvez aussi voir les valeurs standards de registres à l'aide de la boîte de dialogue Expression Evaluator (voir page précédente).

Tableau 6.2
Drapeaux CPU de la
fenêtre Registers

Lettre	Drapeau
c	Carry
z	Zéro
s	Signe
o	Dépassement
p	Parité
a	Carry auxiliaire
i	Activation d'interrupt
d	Direction

Le TurboMenu des registres permet de sélectionner un format d'affichage des valeurs des registres ; choisissez Hexadécimal ou Décimal. Vous pouvez aussi choisir les registres 16 bits (mot) ou 32 bits (mot double).

Points d'arrêt

Le point d'arrêt est l'endroit dans votre code où vous désirez que le programme s'arrête et rende la main au débogueur. Il a le même effet que la commande Run to Cursor, mais vous pouvez placer plusieurs points d'arrêt et vous pouvez ne vous arrêter à un point d'arrêt que sous certaines conditions.

Pose de points d'arrêt

Pour définir un point d'arrêt dans le code, placez le curseur sur la ligne choisie - qui doit contenir du code exécutable - et choisissez Toggle Breakpoint (F5) dans le menu Debug ou dans le TurboMenu de la fenêtre d'édition. La ligne passe en surbrillance.

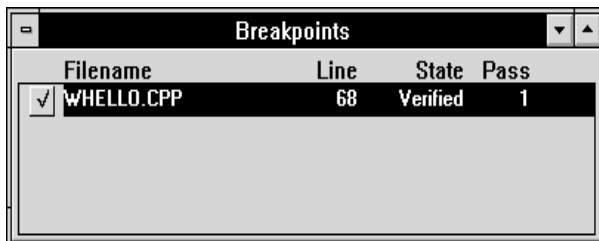
Dorénavant, lorsque vous exécutez votre programme dans l'EDI, il s'arrête lorsqu'il arrive sur cette ligne (sans l'exécuter). La ligne s'affiche dans la fenêtre d'édition, et le point d'exécution est placé dessus. Vous pouvez alors lancer toute opération de débogage : pas à pas, tracé, suivi, inspection ou évaluation.

Pour effacer un point d'arrêt, placez le curseur sur la ligne le contenant et appuyez sur F5.

Utilisation des points d'arrêt

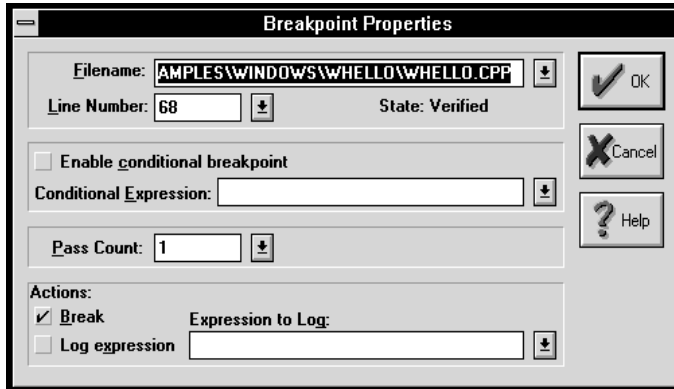
L'EDI mémorise tous vos points d'arrêt lors d'une session de débogage et les associe à votre projet courant. Vous pouvez garder la trace de tous vos points d'arrêt dans une seule fenêtre, ce qui facilite leur utilisation. Choisissez View | Breakpoint pour afficher la fenêtre des points d'arrêt.

Figure 6.3
Fenêtre des points d'arrêt



Dans cette fenêtre, vous pouvez définir et visualiser les propriétés des points d'arrêt. Double-cliquez sur un point d'arrêt pour afficher la boîte de dialogue Breakpoint Properties.

Figure 6.4
Boîte de dialogue
Breakpoint
Properties



Suppression de points d'arrêt

Pour supprimer un point d'arrêt, sélectionnez-le dans la fenêtre Points d'arrêt et choisissez Delete Breakpoint dans le TurboMenu. Pour les supprimer tous, choisissez Delete All.

Vous pouvez aussi utiliser les commandes équivalentes du TurboMenu des Breakpoint Properties.

Désactiver et activer les points d'arrêt

Il est possible de désactiver momentanément un point d'arrêt en désélectionnant la case à cocher du point d'arrêt en question dans la fenêtre Breakpoint. Pour le réactiver, cochez à nouveau cette case.

Vous pouvez aussi utiliser les commandes du TurboMenu. Pour en désactiver ou en activer plusieurs en même temps, faites glisser la souris dessus pour les sélectionner dans la fenêtre Breakpoint (ou appuyez sur *Ctrl* en cliquant sur l'un après l'autre), puis choisissez la commande appropriée dans le TurboMenu.

Visualisation et modification du code sous un point d'arrêt

Même si le point d'arrêt recherché n'est pas affiché dans la fenêtre active, il est facile et rapide de positionner le curseur dessus : sélectionnez-le dans la fenêtre Breakpoint puis choisissez View Source dans le TurboMenu. La fenêtre Breakpoint reste la fenêtre active à l'écran.

Si vous désirez *éditer* le code source du point d'arrêt, choisissez plutôt Edit Source dans le TurboMenu de la fenêtre Breakpoint.

Réinitialisation de points d'arrêt incorrects

Un point d'arrêt, pour être valide, doit être placé sur du code exécutable. Si vous placez un point d'arrêt sur un commentaire, une ligne vierge ou une déclaration, le débogueur vous signale l'erreur en affichant la boîte de dialogue Invalid Breakpoint lors de l'exécution du programme. Ouvrez alors la fenêtre des points d'arrêt et supprimez le point d'arrêt erroné.

Vous pouvez aussi choisir d'ignorer ce message, refermer la boîte de dialogue et continuer l'exécution de votre programme. L'EDI désactive alors le point d'arrêt en question.

Visualisation et modification des propriétés d'un point d'arrêt

Pour visualiser et modifier les propriétés d'un point d'arrêt, faites un double-clic dessus et choisissez Edit Breakpoint dans le TurboMenu. La boîte de dialogue Breakpoint Properties s'ouvre.

Outre examiner les propriétés d'un point d'arrêt, vous pouvez ici en définir un nouveau, en modifier un ou rendre un point d'arrêt conditionnel.

Les sections qui suivent décrivent comment procéder.

Pose d'un point d'arrêt

L'option Toggle Breakpoint du TurboMenu de la fenêtre d'édition est la façon la plus simple de poser un point d'arrêt. Si vous désirez poser un point d'arrêt conditionnel, cependant, utilisez la boîte de dialogue Breakpoint Properties :

1. Placez le curseur à l'endroit où vous désirez poser le point d'arrêt.
2. Choisissez Add Breakpoint dans le menu Debug ou dans le TurboMenu de la fenêtre Points d'arrêt.

La boîte de dialogue Breakpoint Properties s'affiche.

Pour poser un point d'arrêt dans un fichier non ouvert :

1. Dans la boîte de dialogue Breakpoint Properties, entrez le nom du fichier en question.
2. Dans la case Line Number, entrez le numéro de la ligne de code source sur laquelle vous désirez placer le point d'arrêt.

Modification d'un point d'arrêt

Vous pouvez modifier un point d'arrêt existant en changeant ses propriétés dans la boîte de dialogue Breakpoint Properties.

Pose de points d'arrêt après le lancement du programme

Lorsque votre programme s'exécute, vous pouvez passer dans le débogueur et poser un point d'arrêt. Au retour dans votre application, ce nouveau point d'arrêt sera opérationnel.

Points d'arrêt conditionnels

Les points d'arrêt posés avec la commande Toggle Breakpoint du TurboMenu de la fenêtre d'édition sont inconditionnels : dès qu'il arrive sur la ligne en question, le débogueur s'arrête. Cependant, lorsque vous éditez un point d'arrêt, vous disposez d'options dans la boîte de dialogue Breakpoint Properties vous permettant de créer des points d'arrêt *conditionnels*. Vous pouvez définir deux types de conditions pour les points d'arrêt : des *expressions booléennes* ou un *compteur de passages*.

Vous pouvez utiliser une expression booléenne comme condition pour un point d'arrêt. Vous pouvez, par exemple, tester si une variable est comprise dans une certaine fourchette de valeurs, ou si un drapeau est activé. Si la condition spécifiée est vraie, le débogueur s'arrête et vous donne la main.

En spécifiant un nombre de passages sur un point d'arrêt, vous indiquez au débogueur qu'il ne doit s'arrêter au point d'arrêt que lorsque la condition est vraie pour la *xième* fois. Si vous ne spécifiez pas de condition, le débogueur s'arrête à chaque passage sur le point d'arrêt.

Consignation des expressions

Vous pouvez faire en sorte que la valeur d'une expression soit écrite dans la fenêtre Event Log à chaque fois qu'un point d'arrêt est atteint :

1. Sélectionnez Log Expression dans la boîte de dialogue Breakpoint Properties.
2. Dans la case Expression to Log, entrez l'expression à évaluer.

Par exemple, si vous entrez le nom d'une variable dans la boîte Expression to Log de la boîte de dialogue Breakpoint Properties, le débogueur écrit la valeur de cette expression dans la fenêtre Event Log lorsqu'il atteint le point d'arrêt.

Pour de plus amples informations sur la fenêtre Event Log, voir la section "Utilisation de la fenêtre Event Log".

**Personnalisation
des points d'arrêt
et du point
d'exécution**

Vous pouvez utiliser des couleurs pour marquer les points d'arrêt activés, désactivés et incorrects :

1. Choisissez Options | Environment puis Syntax Highlighting | Customize Syntax Highlighting.
2. Dans la liste des éléments, sélectionnez le point d'arrêt à modifier puis choisissez la couleur de fond et de premier plan.

Voir la section "Mise en surbrillance de la syntaxe" du chapitre 1 pour de plus amples informations sur la mise en couleur de la syntaxe.

Procédez de même pour modifier la couleur du point d'exécution.

Gestion des erreurs GPF

Si une erreur GPF (General Protection Fault, Erreur de protection générale) se produit lorsque vous exécutez votre programme dans l'EDI, votre programme s'arrête et la boîte de dialogue General Protection s'affiche. Si vous cliquez sur OK pour fermer la boîte de dialogue, le débogueur affiche le code responsable de l'erreur dans la fenêtre d'édition. Le point d'exécution est positionné sur la partie du code qui est en faute.

Choisissez Debug | Terminate Program pour que votre programme ne se bloque pas, puis corrigez l'erreur et relancez le programme.

Le débogueur sait intercepter la majorité des GPF ; nous ne saurions garantir, toutefois, qu'il les intercepte toutes.

Utilisation de la fenêtre Event Log

Pour afficher la fenêtre Event Log, choisissez View | Event Log.

Dans la section précédente, vous avez appris à consigner la valeur de certaines expressions dans la fenêtre Event Log.

Vous pouvez aussi consigner des messages de fenêtres et de sortie dans la fenêtre Event Log. Pour sélectionner les événements à consigner, choisissez Set Options dans le TurboMenu de la fenêtre Event Log. Sélectionnez le sujet Debugger puis sélectionnez des options Event Capture.

Le tableau ci-dessous décrit toutes les commandes du TurboMenu de la fenêtre Event Log :

Tableau 6.3
TurboMenu de la
fenêtre Event Log

Commande	Description
Save Events to File	Affiche une boîte de dialogue dans laquelle vous spécifiez un nom de fichier. Tous les événements affichés dans la fenêtre Event Log sont alors enregistrés dans ce fichier, ce qui vous permet de les visualiser ultérieurement.
Add Comment	Vous permet d'entrer une ligne de commentaire. Ce texte est ensuite inséré dans la fenêtre Event Log.
Clear Events	Efface la fenêtre Event Log.
Set Options	Affiche la boîte de dialogue Environment Options, qui vous permet de sélectionner les options du débogueur et de définir certaines options de capture des événements ("Event Capture").

Débogage des DLL

Lorsque vous déboguez une DLL en mode pas à pas ou en mode suivi, le débogueur charge automatiquement sa table des symboles (la liste servant à mémoriser les variables, constantes, types et fonctions utilisés dans la DLL).

Etant donné qu'il est impossible de charger plusieurs tables des symboles en mémoire en même temps, la table des symboles de votre fichier .EXE est déchargée lorsque celle de la DLL est chargée. Vous ne pouvez donc pas suivre les variables, inspecter les données, etc, du code de votre .EXE. Vous pourrez le faire, bien entendu, sur le code de la DLL.

Pour savoir quelle table des symboles est utilisée par le débogueur, choisissez Debug | Load Symbol Table ; la liste de la table courante (et des autres tables éventuellement disponibles) s'affiche à l'écran. Si, par exemple, votre EXE s'appelle MONAPPLI.EXE et utilise la DLL appelée MADLL.DLL, lorsque vous déboguez la DLL, MADLL.DLL est listée comme table des symboles courante et MONAPPLI.EXE est listée comme table des symboles disponible.

Il est possible de passer d'une table des symboles à une autre. Par exemple, si vous exécutez pas à pas MADLL.DLL et que vous souhaitez examiner la valeur d'une variable de MONAPPLI.EXE; vous pouvez le faire en chargeant la table des symboles de MONAPPLI.EXE :

1. Choisissez Debug | Load Symbol Table.
2. Dans la liste des tables disponibles, choisissez la table à charger.

N'oubliez pas de revenir à la table originale lorsque vous avez fini votre étude de cette nouvelle table.

Débogage en mode soft et en mode hard

Windows traite et génère continuellement des messages. Tout programme Windows actif (tel que Borland C++ pour Windows) doit faire de même. Vos propres programmes, lorsqu'ils s'exécutent, envoient des messages et gèrent ceux qu'ils reçoivent.

Lorsque vous posez un point d'arrêt dans votre code, cependant, vous désirez que votre programme s'interrompe à ce point d'arrêt. Mais, dans l'environnement Windows, le simple fait de modifier une fenêtre, d'ouvrir une boîte de dialogue, de déplacer le pointeur de la souris ou d'ouvrir un menu génère des messages qui sont envoyés à votre programme en cours d'exécution. Si le débogueur ne gérait pas ces messages pour vous, ils pourraient forcer votre programme à s'exécuter, et même, dans des cas extrêmes, bloquer votre ordinateur.

Heureusement, le débogueur intégré se charge de ces problèmes pour vous, ce qui vous permet de déboguer votre programme tout en profitant de toutes les fonctionnalités de Windows. Lorsque le débogueur fonctionne en même temps que d'autres programmes Windows, on dit qu'il est en mode *soft*.

A cause de certaines limites de Windows, il se peut que vous deviez interrompre tout autre programme que le débogueur. Ce mode s'appelle le mode *hard*. Lorsque le débogueur est dans ce mode, vous n'avez pas accès à diverses fonctionnalités de Windows. Vous ne pourrez pas, par exemple, passer d'une tâche à une autre. Le débogueur considère qu'il est le seul programme en mémoire.

Vous déboguerez vos programmes en mode hard s'ils envoient des messages inter-tâches (des messages DDE, par exemple) ou si vous voulez un débogage de très bas niveau. Pour passer en mode hard :

1. Choisissez Options | Environment ; la boîte de dialogue Environment Options s'ouvre.
2. Choisissez Debugger.
3. Cochez l'option Hard Mode on All Stops.

Si l'option Smart est cochée, le débogueur choisit entre le mode soft et hard en fonction des événements qu'il détecte dans l'environnement Windows. Nous vous recommandons d'utiliser cette option. Pour ce faire :

1. Choisissez Options | Environnement.
2. Choisissez Debugger.
3. Sélectionnez l'option Smart.

WinSight

WinSight est un outil de débogage qui vous permet de visualiser des informations au sujet des fenêtres, des classes de fenêtres et des messages. Utilisez-le pour étudier une application Windows, pour voir comment les fenêtres et les classes de fenêtres sont créées et exploitées, ainsi que pour visualiser tous les messages qui sont transmis d'une fenêtre à l'autre.

Vous pouvez configurer WinSight pour visualiser les messages suivants :

- par fenêtre ;
- par classe de fenêtre ;
- par type de message ;
- selon une combinaison des trois critères ci-dessus.

Sachez que WinSight est un outil passif : il intercepte et affiche de l'information au sujet des messages mais il n'empêche pas la transmission des messages d'une application à l'autre.

Démarrage de WinSight



Il suffit de double-cliquer l'icône WinSight pour faire apparaître sa fenêtre principale dans la configuration standard. Dans cette configuration, la fenêtre affiche la liste de toutes les fenêtres actives sur le bureau. Lorsque vous quittez WinSight, la configuration de travail est automatiquement enregistrée.

WinSight comporte trois volets qui peuvent apparaître dans la fenêtre principale : une arborescence de fenêtres (Window Tree), une liste de classes (Class List) et un tracé de messages (Message Trace). Pour modifier l'emplacement et la dimension des volets, choisissez View | Split Horizontal ou View | Split Vertical.

Vous pouvez spécifier quels messages doivent être tracés par WinSight (voir plus loin), et à quel moment le tracé doit être activé et désactivé. Les sections qui suivent décrivent toutes ces fonctions en détail.

Activation et désactivation de la mise à jour

Pour activer le tracé, choisissez Start! dans le menu (cette option se transforme alors en Stop!). En règle générale, les trois vues sont conservées lors de l'enregistrement des classes, lors de la création ou de la destruction des fenêtres, et lors de la réception des messages. Vous pouvez cependant, en choisissant Messages | Trace Off, suspendre le tracé des messages uniquement. La liste des classes et l'arborescence des fenêtres seront toujours mis à jour.

Utilisez Stop! et Start! pour :

- Etudier une situation particulière
- Minimiser les ressources utilisées par WinSight lorsqu'il met constamment les informations à jour.

Désactivation du tracé des messages

Pour désactiver le tracé du type des messages, choisissez Messages | Trace Off. La vue du tracé des messages reste à l'écran, et le tracé reprend lorsque vous choisissez Messages | Selected Classes, Selected Windows ou All Windows (il faut pour cela que le tracé soit activé).

Les sections qui suivent expliquent comment utiliser les trois vues pour obtenir les données nécessaires au débogage de votre application. Pour quitter WINSIGHT, choisissez Spy | Exit.

Choix d'une vue

WinSight comporte trois volets qui peuvent apparaître dans la fenêtre principale : une arborescence de fenêtres (Window Tree), une liste de classes (Class List) et un tracé de messages (Message Trace). Vous pouvez visualiser un seul ou tous les volets. WinSight redistribue automatiquement les volets par un effet de mosaïque au sein de la fenêtre principale.

Vous pouvez montrer ou cacher n'importe quel volet à tout moment au moyen des commandes du menu View. Les informations et les sélections ne sont pas perdues lorsqu'un volet est caché.

- Le volet Window Tree présente la hiérarchie des fenêtres sur le bureau. Il correspond à l'affichage standard au démarrage de WinSight.
- Le volet Class List montre toutes les classes de fenêtres actuellement recensées au niveau du système.
- Le volet Message Trace affiche des informations au sujet des messages reçus par les fenêtres ou les classes de fenêtres sélectionnées.
- Pour de plus amples informations sur les éléments au sein des volets Window Tree et Class List :
 - Sélectionnez une fenêtre ou une classe puis choisissez Spy | Open Detail.
 - Double-cliquez sur une fenêtre ou une classe.

La fenêtre Window Detail donne des informations détaillées au sujet de cette fenêtre, informations qui viennent en complément de celles de la classe de fenêtres.

Liste des classes

Une classe est le nom sous lequel la fenêtre de classe a été enregistrée auprès de Windows.

Parfois, au lieu de choisir quelles fenêtres tracer, vous désirez voir les messages de classes entières de fenêtres. Ceci est faisable grâce au volet Liste de classes.

Utilisation du volet Liste de classes

Le volet Liste de classes montre toutes les classes de fenêtre actuellement recensées au niveau Windows. Vous pouvez obtenir des détails au sujet d'une classe en double-cliquant son nom ou en validant par *Entrée* lorsqu'elle est sélectionnée.

Les losanges à côté des classes deviennent noirs lorsque la fenêtre reçoit un message. Si une fenêtre enfant cachée reçoit un message, le losange de son parent change de couleur.

Format Classe (Module) Fonction Styles

Classe est le nom de la classe. Certaines classes Windows prédéfinies ont un code numérique (la classe des menus Popup, par exemple, a le code 32768). Ces classe prédéfinies sont indiquées avec leur nom et leur code (#32768:PopupMenu, dans ce cas). Le nom de la classe n'est alors que sa valeur numérique, qui utilise le format MAKEINTRESOURCE, également utilisé pour les identificateurs de ressources.

Module est le nom du module exécutable (.EXE ou .DLL) ayant enregistré la classe.

Fonction est l'adresse de la fonction de fenêtre de classe.

Style est la liste des "cs_styles" de la classe. Les noms sont les mêmes que ceux des "cs_definitions" des WinTypes, mais les caractères "cs_" sont supprimés et le nom comporte à la fois des minuscules et des majuscules.

Tracé des classes

Pour tracer des classes, sélectionnez-les dans la Liste des classes (par *Maj+Clic* ou *Ctrl+Clic*), puis choisissez Messages | Selected Classes. Si le volet Message Trace est masqué à ce moment-là, il redevient visible.

Le fait de tracer des messages vers une classe vous permet de voir tous les messages à destination des fenêtres de cette classe, notamment les messages de création, qui resteraient sinon invisibles.

Pour tracer une autre classe, sélectionnez-en une autre dans la Liste de classes. Choisissez Messages | Trace Off pour désactiver le tracé des messages en direction du volet Message.

Arborescence de fenêtres

L'arborescence de fenêtres affiche la hiérarchie de toutes les fenêtres présentes sur le bureau. Ceci vous permet de :

- Connaître le statut des fenêtres, y compris de celles qui sont cachées.
- Savoir quelles fenêtres reçoivent des messages.
- Sélectionner des fenêtres pour le tracé des messages.

Les traits à gauche de l'arborescence montrent sa structure. Chaque fenêtre est reliée à son parent, à ses descendants et à ses enfants par ces traits. Le losange placé à côté de chaque fenêtre indique si la fenêtre a des enfants. Lorsqu'une fenêtre reçoit un message, son losange (ou le losange de sa fenêtre parent si l'arborescence est condensée) devient noir.



La fenêtre n'a pas d'enfants.



La fenêtre a des enfants mais ils ne sont pas affichés. Pour voir le niveau d'enfants suivants, cliquez sur le losange à côté de la fenêtre. Pour voir tous les niveaux des fenêtres enfants (les enfants des enfants, etc.), cliquez, avec le bouton droit, sur le losange.



La fenêtre a des enfants, et ceux-ci sont affichés (un niveau au moins l'est). Pour masquer toutes les fenêtres enfants, cliquez (ou double-cliquez) sur le losange à côté de la fenêtre.

Format `Pointeur (Classe) Module Position "Titre"`

Pointeur est le pointeur de la fenêtre retourné par `CreateWindow`.

Classe est le nom de la classe, décrit dans le volet `Class List`.

Module est le nom du module exécutable (.EXE ou .DLL) ayant créé la fenêtre. Il s'agit du nom du module auquel appartient le segment de données passé comme paramètre *Instance* à `CreateWindow`.

Position est "hidden" (masqué) si la fenêtre est masquée. Dans le cas contraire, *Position* est spécifié en utilisant les coordonnées écran (pour les fenêtres parents) ou les coordonnées dans la zone client du parent (pour les fenêtres enfants). Le format de *Position* est le suivant :

`xBegin, yBegin- xEnd, yEnd`

Titre est le titre de la fenêtre, retourné par `GetWindowText` ou un message `wm_GETTEXT`. S'il s'agit d'une chaîne vide, les guillemets sont omis.

Recherche d'une fenêtre

WinSight a un mode propre de recherche de fenêtres. Il peut s'utiliser de deux façons : soit en identifiant la ligne de l'arborescence de fenêtres qui correspond à la fenêtre sur laquelle vous pointez, soit en mettant en surbrillance la fenêtre que vous sélectionnez dans l'arborescence de fenêtres.

Important ! Lorsque vous êtes en mode de recherche de fenêtre, toutes les autres tâches sont interrompues.

Passez en mode de recherche de fenêtre en choisissant `Spy | Find Window`. Dans ce mode, dès que la souris passe sur une fenêtre, une bordure épaisse entoure cette fenêtre et elle est sélectionnée dans l'arborescence de fenêtres.

De plus, une fois en mode recherche, vous pouvez sélectionner des fenêtres dans l'arborescence avec la souris ou les touches de déplacement pour obtenir le même effet. Si vous appuyez sur *Entrée*, la fenêtre `Window Detail` s'ouvre.

Sortie du mode de recherche de fenêtres

Une fois que vous avez localisé la fenêtre de votre choix, vous pouvez sortir du mode recherche en cliquant ou en appuyant sur `Echap`. La bordure s'efface et la description de la fenêtre courant reste sélectionnée dans l'arborescence.

Tracé des fenêtres

Pour analyser une ou plusieurs fenêtres, sélectionnez-les (avec la souris et les touches *Maj* ou *Ctrl*) puis choisissez `Messages | Selected Windows`. Pour tracer d'autres fenêtres, sélectionnez d'autres fenêtres dans l'arborescence.

Pour analyser toutes les fenêtres, quelle que soit la sélection dans la Liste des classes ou dans l'arborescence, choisissez `Messages | All Windows`.

Le Tracé des messages devient visible si vous choisissez `Messages | Selected Windows` ou `Windows | All Windows`.

Choisissez `Messages | Trace Off` pour désactiver le tracé des messages sans masquer la fenêtre `Message Trace`.

Choix des messages à tracer

Le Tracé des messages affiche les messages reçus par les classes de fenêtres ou par les fenêtres sélectionnées. Les messages reçus via `SendMessage` sont indiqués deux fois : quand ils sont envoyés et lorsqu'ils sont retournés pour montrer la valeur de retour.

Les messages envoyés ne sont affichés qu'une fois, leur valeur de retour étant sans importance. L'affichage est mis en retrait pour montrer l'imbrication des messages.

**Utilisation du volet
Tracé de messages**

Par défaut, WINSIGHT trace tous les messages et les affiche dans le volet Tracé de messages. Il existe plusieurs façons de restreindre le tracé des messages :

- Choisissez Messages | Selected Classes ou Messages | Selected Windows puis sélectionnez des classes (dans la liste des classes) ou des fenêtres (dans l'arborescence de fenêtres) avec la souris et les touches *Maj* ou *Ctrl*.
- Choisissez Message | All Windows.
- Choisissez Message | Options puis sélectionnez un ou plusieurs groupes de messages (ces groupes sont décrits aux tableaux 7.1 à 7.14). Cochez All Messages dans la boîte de dialogue Options pour réactiver le tracé de tous les messages.

**Autres options de
tracé**

La boîte de dialogue Message Trace Options vous permet de changer le format des messages, ainsi que de les rediriger sur une imprimante, dans un fichier ou vers un moniteur ou une fenêtre auxiliaire.

- Normalement, le Tracé des messages interprète les paramètres de chaque message et les affiche dans un format lisible (l'option Interpret Values est cochée). Cochez Hex Values si vous désirez voir les paramètres de messages sous forme de valeurs hexadécimales de *wParam* et de *lParam*.
- Les informations sur les messages tracés s'affichent normalement dans le volet de Tracé des messages. Il est possible, cependant, de les rediriger sur une imprimante, dans un fichier ou vers un moniteur auxiliaire en cochant Log File dans la boîte de dialogue Message Trace Options et en faisant l'une des opérations ci-dessous :

- Entrez le nom du fichier journal. s'il existe déjà, les messages lui sont ajoutés.
- Entrez le nom du périphérique (PRN, par exemple) sur lequel les sorties doivent être redirigées.
- Entrez AUX pour rediriger les sorties sur un moniteur ou une fenêtre auxiliaire. Pour pouvoir faire ceci, WINOX.SYS ou OX.SYS doivent être installés comme périphériques dans votre CONFIG.SYS.

Pour arrêter la redirection des sorties, désélectionnez Log File dans la boîte de dialogue Message Trace Options.

Format Pointeur ["Titre" ou {Classe}] Message Etat

Pointeur est le pointeur de la fenêtre recevant le message.

Titre est le titre de la fenêtre. S'il s'agit d'une chaîne vide, la classe est indiquée entre accolades.

Message est le nom du message défini par Windows. Ces noms sont affichés en majuscules par WINSIGHT. Les messages connus non documentés sont affichés en minuscules. Les numéros de messages inconnus définis par l'utilisateur) sont indiqués sous la forme *wm_User+0xXXXX* s'ils sont supérieurs ou égaux à *wm_User*, et sous la forme *wm_0xXXX* s'ils sont inférieurs à *wm_User*. Le nom de l'identificateur des messages enregistrés est placé entre guillemets simples.

Etat peut prendre l'une des valeurs suivantes :

- *Dispatched* si le message a été reçu via `DispatchMessage`.
- *Sent [from XXXX]* si le message a été reçu via `SendMessage`. S'il provient d'une autre fenêtre, `from XXXX` indique le pointeur de cette fenêtre. Si l'émetteur et le récepteur sont la même fenêtre, cette chaîne devient `from self`. Si le message a été envoyé par Windows, le mot "from" est omis.
- *Returns* si le message a été reçu via `SendMessage` et est maintenant en phase de retour.
- Il se peut que d'autres messages incluent une valeur de retour numérique ou un message textuel tel que *wm_GetText*. Pour les messages envoyés et expédiés (*Sent* et *Dispatched*), WINSIGHT interprète les paramètres et les affiche de façon lisible. Si un message est associé à des structures de données (*wm_Create*, par exemple), WINSIGHT inclut ces structures dans les informations qu'il affiche.

Tableau 7.1: Messages souris

WM_HSCROLL	WM_MBUTTONDOWN	WM_RBUTTONDOWN
WM_LBUTTONDOWNBLCK	WM_MOUSEACTIVATE	WM_RBUTTONUP
WM_LBUTTONDOWN	WM_MOUSEFIRST	WM_SETCURSOR
WM_LBUTTONUP	WM_MOUSELAST	WM_VSCROLL
WM_MBUTTONDOWNBLCK	WM_MOUSEMOVE	
WM_MBUTTONDOWN	WM_RBUTTONDOWNBLCK	

Tableau 7.2 : Messages fenêtres

WM_ACTIVATE	WM_GETDLGCODE	WM_QUERYNEWPALETTE
WM_ACTIVATEAPP	WM_GETFONT	WM_QUERYOPEN
WM_CANCELMODE	WM_GETMINMAXINFO	WM_QUIT
WM_CHILDACTIVATE	WM_GETTEXT	WM_SETFOCUS
WM_CLOSE	WM_GETTEXTLENGTH	WM_SETFONT
WM_CREATE	WM_ICONERASEBKGD	WM_SETREDRAW
WM_CTLCOLOR	WM_KILLFOCUS	WM_SETTEXT
WM_DDE_FIRST	WM_MOVE	WM_SHOWWINDOW
WM_DESTROY	WM_PAINT	WM_SIZE
WM_ENABLE	WM_PAINTICON	WM_WINDOWPOSCHANGED
WM_ENDSESSION	WM_QUERYDRAGICON	WM_WINDOWPOSCHANGING
WM_ERASEBKGD	WM_QUERYENDSESSION	

Tableau 7.3 : Messages clavier

WM_CHAR	WM_KEYUP	WM_SYSKEYDOWN
WM_CHARTOITEM	WM_MENUCHAR	WM_SYSKEYUP
WM_COMMAND	WM_MENUSELECT	WM_TIMER
WM_DEADCHAR	WM_PARENTNOTIFY	WM_VKEYTOITEM
WM_KEYDOWN	WM_SYSCHAR	
WM_KEYLEAST	WM_SYSDEADCHAR	

Tableau 7.4 : Messages système

WM_COMPACTING	WM_PALETTECHANGED	WM_SYSCOLORCHANGE
WM_DEVMODECHANGE	WM_PALETTEISCHANGING	WM_SYSCOMMAND
WM_ENTERIDLE	WM_POWER	WM_TIMECHANGE
WM_FONTCHANGE	WM_QUEUESYNC	WM_WININICHANGE
WM_NULL	WM_SPOOLERSTATUS	

Tableau 7.5 : Messages d'initialisation

WM_INITDIALOG	WM_INITMENU	WM_INITMENUPOPUP
---------------	-------------	------------------

Tableau 7.6 : Messages du presse-papiers

WM_ASKCBFORMATNAME	WM_DESTROYCLIPBOARD	WM_RENDERALLFORMATS
WM_CHANGECHAIN	WM_DRAWCLIPBOARD	WM_RENDERFORMAT
WM_CLEAR	WM_HSCROLLCLIPBOARD	WM_SIZECLIPBOARD
WM_COPY	WM_PAINTCLIPBOARD	WM_UNDO
WM_CUT	WM_PASTE	WM_VSCROLLCLIPBOARD

Tableau 7.7 : Messages DDE

WM_DDE_ACK	WM_DDE_EXECUTE	WM_DDE_REQUEST
WM_DDE_ADVISE	WM_DDE_INITIATE	WM_DDE_TERMINATE
WM_DDE_DATA	WM_DDE_POKE	WM_DDE_UNADVISE

Tableau 7.8 : Messages non clients

WM_NCACTIVATE	WM_NCLBUTTONDOWN	WM_NCPAINT
WM_NCCALCSIZE	WM_NCLBUTTONUP	WM_NCRBUTTONDBLCLK
WM_NCCREATE	WM_NCMBUTTONDBLCLK	WM_NCRBUTTONDOWN
WM_NCDESTROY	WM_NCMBUTTONDOWN	WM_NCRBUTTONUP
WM_NCHITTEST	WM_NCMBUTTONUP	
WM_NCLBUTTONDBLCLK	WM_NCMOUSEMOVE	

Tableau 7.9 : Messages d'impression

DM_COLOR	DM_MODIFY	DM_printquality
DM_COPIES	DM_ORIENTATION	DM_prompt
DM_COPY	DM_OUT_BUFFER	DM_scale
DM_DEFAULTSOURCE	DM_out_default	DM_specversion
DM_DUPLEX	DM_paperlength	DM_ttoption
DM_IN_BUFFER	DM_papersize	DM_update
DM_IN_PROMPT	DM_paperwidth	DM_yresolution

Tableau 7.10 : Messages de contrôle

BM_GETCHECK	CBN_SELCHANGE	EN_VSCROLL
BM_GETSTATE	CBN_SELENDCANCEL	
BM_SETCHECK	CBN_SETFOCUS	LB_ADDSTRING
BM_SETSTATE		LB_DELETESTRING
BM_SETSTYLE	DM_GETDEFID	LB_DIR
	DM_SETDEFID	LB_FINDSTRING
BN_CLICKED		LB_FINDSTRINGEXACT
BN_DISABLE	EM_CANUNDO	LB_GETCAREINDEX
BN_DOUBLECLICKED	EM_EMPTYUNDOBUFFER	LB_GETCOUNT
BN_HILITE	EM_FMTLINES	LB_GETCURSEL
BN_PAINT	EM_GETFIRSTVISIBLELINE	LB_GETHORIZONTALEXTENT
BN_UNHILITE	EM_GETHANDLE	LB_GETITEMDATA
	EM_GETLINE	LB_GETITEMHEIGHT
CB_ADDSTRING	EM_GETLINECOUNT	LB_GETITEMRECT
CB_DELETESTRING	EM_GETMODIFY	LB_GETSEL
CB_DIR	EM_GETPASSWORDCHAR	LB_GETSELCOUNT
CB_FINDSTRING	EM_GETRECT	LB_GETSELITEMS
CB_FINDSTRINGEXACT	EM_GETSEL	LB_GETTEXT
CB_GETCOUNT	EM_GETTHUMB	LB_GETTEXTLEN
CB_GETCURSEL	EM_GETWORDBREAKPROC	LB_GETTOPINDEX
CB_GETDROPPEDCONTROLRECT	EM_LIMITTEXT	LB_INSERTSTRING
CB_GETDROPPEDSTATE	EM_LINEFROMCHAR	LB_MSGMAX
CB_GETEDITSEL	EM_LINEINDEX	LB_RESETCONTENT
CB_GETEXTENDEDUI	EM_LINELENGTH	LB_SELECTSTRING
CB_GETITEMDATA	EM_LINESCROLL	LB_SELITEMRANGE
CB_GETITEMHEIGHT	EM_MSGMAX	LB_SETCAREINDEX
CB_GETLBTEXT	EM_REPLACESEL	LB_SETCOLUMNWIDTH
CB_GETLBTEXTLEN	EM_SCROLL	LB_SETCURSEL
CB_INSERTSTRING	EM_SETFONT	LB_SETHORIZONTALEXTENT
CB_LIMITTEXT	EM_SETHANDLE	LB_SETITEMDATA
CB_MSGMAX	EM_SETMODIFY	LB_SETITEMHEIGHT
CB_RESETCONTENT	EM_SETPASSWORDCHAR	LB_SETSEL
CB_SELECTSTRING	EM_SETRECT	LB_SETTABSTOPS
CB_SETCURSEL	EM_SETRECTNP	LB_SETTOPINDEX
CB_SETEDITSEL	EM_SETSEL	
CB_SETITEMDATA	EM_SETTABSTOPS	LBN_DBLCLK
CB_SETITEMHEIGHT	EM_SETWORDBREAK	LBN_KILLFOCUS
CB_SHOWDROPDOWN	EM_UNDO	LBN_SELCANCEL
CBN_CLOSEUP	EN_CHANGE	LBN_SELCHANGE
CBN_DBLCLK	EN_ERRSPACE	LBN_SETFOCUS
CBN_DROPDOWN	EN_HSCROLL	
CBN_EDITCHANGE	EN_KILLFOCUS	STM_GETICON
CBN_EDITUPDATE	EN_MAXTEXT	STM_SETICON
CBN_ERRSPACE	EN_SETFOCUS	
CBN_KILLFOCUS	EN_UPDATE	

Tableau 7.11 : Messages stylo

WM_USER	WM_HOOKRCRESULT	WM_RCRESULT
WM_GLOBALRCCHANGE	WM_PENWINFIRST	WM_SKB
WM_HEDITCTL	WM_PENWINLAST	

Tableau 7.12 : Messages multimédia

MM_ADLIB	MM_MIM_CLOSE	MM_SNDBLST_MIDIIN
MM_JOY1BUTTONDOWN	MM_MIM_DATA	MM_SNDBLST_MIDIOUT
MM_JOY1BUTTONUP	MM_MIM_ERROR	MM_SNDBLST_SYNTN
MM_JOY1MOVE	MM_MIM_LONGDAT	MM_SNDBLST_WAVEIN
MM_JOY1ZMOVE	MM_MIM_LONGERROR	MM_SNDBLST_WAVEOUT
MM_JOY2BUTTONDOWN	MM_MIM_OPEN	MM_WAVE_MAPPER
MM_JOY2BUTTONUP	MM_MOM_CLOSE	MM_WIM_CLOSE
MM_JOY2MOVE	MM_MOM_DONE	MM_WIM_DATA
MM_JOY2ZMOVE	MM_MOM_OPEN	MM_WIM_OPEN
MM_MCIINOTIFY	MM_MPU401_MIDIIN	MM_WOM_CLOSE
MM_MICROSOFT	MM_MPU401_MIDIOUT	MM_WOM_DATA
MM_MIDI_MAPPER	MM_PC_JOYSTICK	MM_WOM_OPEN

Tableau 7.13 : Autres messages

WM_COALESCE_FIRST	WM_MDIACTIVATE	WM_MDIRESTORE
WM_COALESCE_LAST	WM_MDICASCADE	WM_MDISETMENU
WM_COMMNOTIFY	WM_MDICREATE	WM_MDITILE
WM_COMPAREITEM	WM_MDIDESTROY	WM_MEASUREITEM
WM_DELETEITEM	WM_MDIGETACTIVE	WM_NEXTDLGCTL
WM_DRAWITEM	WM_MDIICONARRANGE	WM_SYSTEMERROR
WM_DROPFILES	WM_MDIMAXIMIZE	
WM_KEYFIRST	WM_MDINEXT	

Tableau 7.14 : Messages non documentés par Microsoft

WM_ALTTABACTIVE	WM_ENTERSIZEMOVE	WM_QUERYPARKICON
WM_BEGINDRAG	WM_EXITMENULOOP	WM_SETHOTKEY
WM_CONVERTREQUEST	WM_EXITSIZEMOVE	WM_SETVISIBLE
WM_CONVERTRESULT	WM_FILESYSCHANGE	WM_SIZEWAIT
WM_DRAGLOOP	WM_GETHOTKEY	WM_SYNCPAINT
WM_DRAGMOVE	WM_ISACTIVEICON	WM_SYNCTASK
WM_DRAGSELECT	WM_LBTRACKPOINT	WM_SYSTIMER
WM_DROPOBJECT	WM_NEXTMENU	WM_TESTING
WM_ENTERMENULOOP	WM_QUERYDROPOBJECT	WM_YOMICHR

WinSpector

WinSpector, accompagné de ses utilitaires, vous permet de réaliser un test postmortem des UAE ("Unrecoverable Application Errors") et des GPF ("General Protection Faults"). Lorsqu'une UAE ou une GPF se produit, WinSpector écrit des informations dans un fichier journal. Ce fichier fournit des informations concernant la raison de l'erreur. Voici le type d'informations que vous trouverez dans ce fichier journal :

- La pile d'appel qui était active au moment de l'erreur.
- Les noms des fonctions et procédures de la pile d'appel.
- Les registres de la CPU.
- Un désassemblage des instructions machine pour lesquelles l'erreur a été détectée.
- Les informations Windows concernant l'environnement d'exécution du programme.

Utilisation de WinSpector

Avant d'utiliser WinSpector, assurez-vous que le fichier TOOLHELP.DLL (de Windows 3.1 ou d'une version ultérieure) figure bien dans un des répertoires normalement accessibles. Ce fichier est une bibliothèque DLL Windows, fournie avec Borland C++, qui permet aux utilitaires d'accéder aux informations système de bas niveau. WinSpector utilise TOOLHELP.DLL pour détecter si une erreur a été mise en évidence et pour obtenir les informations système qu'il écrit alors dans le fichier journal. N'utilisez aucun autre outil que Turbo Debugger en même temps que WinSpector. Vous pouvez lancer WinSpector de trois façons :



WinSpector

- L'inclure dans la section "load=" de votre WIN.INI.
- L'inclure dans le dossier Ouverture au démarrage de Windows
- Cliquer sur son icône depuis Windows.

Lorsqu'une erreur UAE ou GPF est détectée, WinSpector crée un rapport dans le fichier texte WINSPECTR.LOG pour vous aider à déterminer la cause de l'erreur. Il crée aussi le fichier binaire WINSPECTR.BIN, que l'utilitaire DFA traduit en un fichier texte du nom DFA.OUT (voir la description de DFA.EXE dans les pages qui suivent).

Lorsqu'une erreur se produit, WinSpector affiche une boîte de dialogue contenant un rapport succinct. Vous pouvez cliquer OK pour désactiver l'affichage de la boîte de dialogue et lire ensuite le fichier journal pour prendre connaissance de la cause de l'erreur. Pour modifier les options de sortie dans le fichier journal, reportez-vous à la section qui suit.

Configuration de WINSPECTR.LOG

Il existe deux façons de définir les options de WinSpector qui contrôlent le contenu de WINSPECTR.LOG.

- Pour utiliser la boîte de dialogue WinSpector Preferences, lancez WinSpector, cliquez sur l'icône WinSpector et choisissez Preferences dans le menu surgissant.
- Pour éditer les commandes de WINSPECTR.INI, modifiez-le à l'aide d'un éditeur de texte puis enregistrez-le et relancez WinSpector

Les sections qui suivent décrivent chaque option de la boîte de dialogue Preferences. Les options de WINSPECTR.INI sont listées à gauche.

LogDir=[répertoire]

Le répertoire dans lequel est situé le fichier WINSPECTR.LOG. Vous pouvez spécifier le répertoire de votre choix ; par défaut, il s'agit de C:\WINDOWS.

LogViewer=[visualiseur]

- Le visualiseur est le programme que WinSpector utilise pour afficher le fichier journal. Entrez le chemin d'accès et le nom du programme de votre choix (NOTEPAD.EXE par défaut). Si WinSpector ne trouve pas le programme, il affiche le message

Error: Unable to execute: [programme]

programme est le nom du fichier de l'éditeur. Vérifiez que celui-ci est bien indiqué dans le répertoire mentionné.

CreateNewLog=0
(ajouter) ou 1 (effacer)

- Les options Append New Reports et Overwrite Previous Reports vous permettent de spécifier si les nouvelles informations doivent s'ajouter aux précédentes ou si elles doivent les effacer.

ShowSystemInfo=0
(non) ou 1 (oui)

- Cochez System Information pour ajouter la Task List, la Module List et les informations concernant les tas USER et GDI au fichier journal.

LogToStdAux=0 (non)
ou 1 (oui)

■ Cochez AUX Summary pour voir, sur le périphérique AUX, une forme abrégée des informations envoyées dans le fichier journal. Il faut pour cela qu'un terminal soit connecté à la sortie AUX ou qu'un pilote tel que OX.SYS redirige les sorties du périphérique AUX sur un second moniteur.

PostMortemDump=0
(non) ou 1 (oui)

■ Cochez l'option PostMortem Dump pour générer le fichier WINSPCTR.BIN. Utilisez DFA.EXE pour traduire ce fichier en fichier texte lisible.

ShowStackInfo=0 (non)
ou 1 (oui)

■ Cochez Stack Frame Data pour ajouter au fichier journal un listing détaillé ("verbeux") du tracé de pile. Pour chaque cadre de pile ne dépassant pas 256 octets, WinSpector effectue un dump hexadécimal en commençant au SS:BP du cadre. Si deux cadres de pile comptent plus de 256 octets, l'affichage de la mémoire n'est pas effectué pour ce cadre. Ces valeurs vous permettent de connaître les paramètres passés à la fonction.

Il est en général plus simple de laisser DFA travailler à l'interprétation des paramètres. Cependant, si les informations de Turbo Debugger ne sont pas disponibles, ces données vous seront très utiles.

ShowUserInfo=0 (non)
ou 1 (oui)

■ Cochez User Comments si vous désirez ajouter au fichier journal des informations concernant l'opération en cours lors de l'erreur. Si cette option est cochée, WinSpector affiche une boîte de dialogue juste après l'erreur. Ce que vous saisissez est envoyé au fichier journal.

WINSPECTR.LOG

Références

Les rapports générés par WinSpector ont plusieurs sections qui vous aident à savoir ce qui s'est passé lors de l'exception. La première ligne du rapport figurant dans WINSPECTR.LOG donne la date de valeur (date et heure) de l'exception. Vous obtiendrez par exemple :

```
WinSpector failure report - 01/18/1994 11:04:25
```

La deuxième ligne donne les informations suivantes :

- Type d'exception survenue.
- Nom du module.
- Adresse logique.
- Adresse physique.
- Tâche active au moment de l'erreur.

La deuxième ligne peut contenir des informations de ce type :

```
Exception 13 at USER 002A:0429 (079F:0429) (TASK=BAD)
```

Tableau 8.1 : Types d'exceptions

Numéro	Nom	Description
0	Division par zéro	Se produit lors d'une interaction DIV ou IDIV si le diviseur est nul.
12	Erreur de pile	Cette erreur est généralement provoquée lorsque l'espace de pile s'avère insuffisant pour permettre la poursuite de l'opération en cours.
13	Erreur de protection générale	Toutes les erreurs de protection ne générant pas d'autre erreur provoquent une exception du type 13.

Les exceptions du type 13 incluent, entre autres, les erreurs suivantes :

- Sélecteur incorrect chargé dans un registre de segment.
- Limite de segment dépassée. Bien que le sélecteur soit valide, la valeur de déplacement est supérieure à la limite du segment (par exemple, un index de tableau ne respectant pas les limites autorisées dans DS, ES, ou dans d'autres segments).
- L'exécution est transférée à un segment non exécutable, tel qu'un pointeur invalide sur une fonction.
- Accès aux registres DS, ES, FS ou GS contenant un sélecteur nul. (Cette erreur peut générer la présence d'un 0 dans le registre de segment du fichier journal).

Le fichier journal mentionne l'adresse physique et l'adresse logique où l'exception s'est produite. Ces deux types d'adresses sont indispensables à la bonne exécution des programmes Windows pour les raisons suivantes :

- Lorsqu'un programme est chargé, Windows alloue de l'espace pour chaque segment logique et affecte un sélecteur unique à chaque segment. Le sélecteur et son déplacement sont combinés à partir d'une adresse physique.
- Lorsqu'un fichier .EXE Windows est lié, chaque segment est placé dans une section différente du fichier et une table des segments est alors créée.
- Une adresse logique, qui correspond effectivement à la position du segment dans la table des segments Windows, se présente comme suit : nom du module, segment logique et déplacement. Vous pouvez exécuter TDUMP sur le fichier pour déterminer la taille du segment ainsi que d'autres informations. Vous pouvez également générer un fichier .MAP qui contient le même type d'informations.

Si le pointeur de pile est trop petit au moment de l'erreur, TOOLHELP.DLL permute automatiquement la pile et le message `Stack Switched` est ajouté à la fin de la deuxième ligne du fichier journal.

**Section
Disassembly**

La première ligne de la section Disassembly du fichier journal identifie l'instruction du langage d'assemblage générant l'exception. Sous cette ligne figurent quelques instructions du programme. Ces commandes sont mentionnées afin d'être utilisées comme repère lors de la détermination de la tâche qui a provoqué l'exception.

Dans l'exemple suivant, ES correspond au registre de segment qui contient un sélecteur et BX correspond au déplacement du segment ; une exception de type 13 s'est produite parce que la valeur du déplacement BX est supérieure à la limite du segment référencé par ES :

```
079F:0429  CMP     BYTE  PTR  ES: [BX], FF
079F:047D  JNE     043A
079F:042F  CMP     WORD  PTR  [BP+406], 03
079F:0435  MOV     DI, 0001
```

Section Stack Trace

La première ligne de la section Stack Trace (tracé de pile) du fichier journal identifie la fonction ou la procédure exécutée au moment de l'exception. Les informations fournies incluent les éléments suivants :

- Numéro du cadre.
- Nom du module.
- Nom de la fonction la plus proche de l'adresse de celle qui a provoqué l'erreur, suivi d'un nombre indiquant à quelle distance vous vous trouviez par rapport à cette fonction (ces informations sont données uniquement si le fichier .SYM existe).
- Adresses physiques et logiques du cadre de pile.
- Adresse de retour du programme après l'appel.

Lorsque WinSpector donne les noms des fonctions, il recherche dans le fichier .SYM le nom symbolique le plus proche précédant l'adresse contenue dans la pile d'appel. Certains fichiers .SYM ne contiennent pas les informations pour toutes les fonctions. Par conséquent, le nom de la fonction figurant dans le fichier journal correspond à celui de la fonction la plus proche figurant dans le fichier .SYM avec une adresse précédant l'adresse de cadre. Si la valeur du champ de déplacement est trop élevée, les noms des fonctions peuvent ne pas être valables.

Les informations suivantes concernant l'analyse de la pile donnent les noms des fonctions exécutées au moment où la tâche BAD (notre exemple) a généré une exception :

```
Stack Trace:
0  User  <no info>
   CS:IP  002A:0429 (079F:0429)   SS:BP  10FF:18CA
   C:\WIN31\SYSTEM\USER.EXE
   .
   .
   .

3  BAD function5(unsigned long, unsigned long, unsigned
long) + 0014
   CS:IP 0001:0184 (1107:0184)   SS:BP  10 FF:1952
   C\BIN\BAD.EXE
   .
   .
   .
```

Section Registers

La section Registers de WINSPECTR.LOG donne la liste des valeurs stockées dans les registres du processeur au moment où l'erreur s'est produite, comme le montre l'exemple suivant :

```
Registers:
AX  0037
BX  0000
CX  0008
DX  10EE
SI  0037
DI  0028
```

Les restrictions et les droits d'accès sont donnés pour les registres CS, DS, ES et SS.

**Section
Message Queue :**

Les informations L enregistrées dans la section File d'attente des messages ne correspondent pas obligatoirement à celles du dernier message que le programme a reçu. L'utilisation de la fonction SendMessage permet par exemple à Windows de court-circuiter la file d'attente des messages.

La section Message Queue du fichier journal contient le dernier message reçu pendant le traitement. Cette section contient aussi les messages figurant dans la file d'attente au moment de l'exception. Pour chaque message, WinSpector donne les informations suivantes :

- Le descripteur de fenêtre de la fenêtre de destination.
- Le numéro d'identification du message.
- Deux paramètres contenant des informations supplémentaires concernant le message.

Dans cet exemple, nous voyons un message reçu et un message en attente :

```
Message Queue:
Last message received:
  hWnd: 0000 msg: 001   wParam: 0002 lParam: 00000003
Waiting in queue:
  hWnd: 0000 msg: 0001  wParam : 0002 lParam : 00000003
```

Section Tasks

Dans la section Tasks du fichier journal, WinSpector donne la liste des programmes exécutés au moment de l'erreur. Cette section contient les informations suivantes :

- Chemin complet du fichier exécutable.
- Nom du module.
- Descripteur du module Windows.
- Descripteur de tâche.
- Valeur du segment de données associé à la tâche (descripteur d'instance).

Voici certaines des tâches exécutées lorsque notre application, BAD, a généré une exception :

```
C:\WIN3\SYSTEM\NWPOPOP.EXE
  Module: NWPOPOP  hModule: 142F  hTask: 141F  hInstance: 13F6
.
.
.
C:\BIN\WINSPECTR.EXE
  Module: WINSPECTR  hModule: 1397  hTask: 1367  hInstance: 135E
.
.
.
C:\BIN\BAD.EXE
  Module:BAD        hModule: 1467  hTask: 1127  hInstance: 10FE
```

Section Modules

Dans la section Modules du fichier journal, WinSpector donne la liste des modules exécutés au moment de l'exception. Pour chaque module, les informations suivantes sont données :

- Chemin du fichier exécutable.
- Date de création du fichier exécutable.
- Taille du fichier.
- Nom du module.
- Descripteur du module.
- Compteur de référence indiquant combien de fois ce module a été appelé.

Voici trois des modules exécutés lorsque notre application exemple, BAD, a généré une exception :

```
C:\WIN31\SYSTEM\KRNL386.EXE      Date: 03/02/1992   Size: 116132
  Module: KERNEL      hModule: 010F   reference count: 21
C:\WIN31\SYSTEM\SYSTEM.DRV      Date: 03/01/1992   Size: 2304
  Module: SYSTEM      hModule: 013F   reference count: 13
.
.
.
C:\BIN\WINSPECTR.EXE            Date: 06/02/1992   Size: 46256
  Module: WINSPECTR   hModule: 1397   reference count: 1
.
.
.
```

Section des tas USER et GDI

Cette section du fichier journal indique quel était le pourcentage de mémoire de tas attribuée à USER et GDI (Graphics Device Interface) au moment de l'exception. Etant donné que Windows ne dispose que d'un espace de tas interne de 64 ko qui doit être partagé entre toutes les applications, il s'avère dès lors souvent utile d'effectuer le suivi de l'espace effectivement utilisé. Si vous estimez que USER et GDI utilisent une trop grande partie du tas, vérifiez que vous avez libéré les ressources que vous n'utilisez pas.

Dans notre exemple, cette section se présente ainsi :

```
USER      Free   91%
GDI       Free   83%
```

**Section
System Info**

Cette section indique le mode et la version de Windows. Elle contient aussi les informations suivantes :

- Type de CPU.
- Plus grand bloc libre de mémoire linéaire contiguë disponible.
- Espace total de l'adressage linéaire représenté sous forme de pages.
- Nombre de pages de mémoire libre dans l'espace d'adressage linéaire.
- Nombre de pages dans le fichier d'échange système.

Par exemple, pour un ordinateur 486 tournant en mode étendu Windows 3.1, la quantité de mémoire disponible est indiquée dans la page-écran suivante :

```
System info: Running in enhanced mode under Windows 3.1
debug version
CPU : 80486
Largest free memory block: 3457024 bytes
Total linear memory space: 19696 K
Free linear memory space: 18212 K
Swap file Pages: 0 (0K)
```

Traitement des données WinSpector

DFA est un utilitaire qui récupère un fichier WINSPECTR.BIN et les informations issues de Turbo Debugger (fichiers .EXE, .DLL ou .TDS) et transforme les données binaires pour les rendre lisibles. Il crée ainsi un fichier contenant non seulement les informations d'analyse de pile, comme le fichier journal, mais aussi les noms de fonctions, les numéros de ligne et les variables locales et globales.

L'utilitaire DFA effectue un post-traitement des informations de Turbo Debugger que WinSpector a regroupées au moment de l'exception. Si vous choisissez d'envoyer les informations du rapport à un rapport postmortem, WinSpector écrit un fichier WINSPECTR.BIN au moment de l'erreur. Vous pouvez ensuite utiliser DFA pour traduire les données binaires du fichier WINSPECTR.BIN en informations utilisables qui seront stockées dans le fichier DFA.OUT .

Puisque le système n'écrit qu'un seul fichier WINSPECTR.BIN par session Windows, assurez-vous que vous exécutez DFA rapidement. Exemple : si vous obtenez trois UAE successivement, WinSpector écrit alors trois rapports dans le fichier journal ; les données n'existeront que pour le premier rapport. Nous vous recommandons d'exécuter DFA immédiatement après avoir reçu le premier message UAE. Vous pourrez ensuite renommer le fichier DFA.OUT et supprimer les fichiers WINSPECTR.BIN et WINSPECTR.LOG avant de poursuivre les opérations.

Sortie de DFA

DFA n'écrit un fichier que s'il trouve des informations de Turbo Debugger pour le fichier dans le cadre de pile. Le fichier de sortie DFA (DFA.OUT) contient un tracé de pile identique à celui contenu dans le fichier journal WinSpector. Ce fichier contient de plus les éléments suivants :

- Noms des fonctions.
- Numéros de lignes.
- Variables locales et globales.
- Segments de données et leurs valeurs (y compris le segment de pile).

Utilisation de DFA avec le fichier WINSPECTR.LOG

Lorsque DFA est utilisé uniquement avec le fichier WINSPECTR.LOG , il donne les informations minimales concernant l'analyse de la pile, telles que les adresses. Si les informations de Turbo Debugger (contenues dans un fichier .EXE, .DLL ou .TDS) figurent dans un fichier exécutable ou dans un fichier séparé, les noms des fichiers sources ainsi que les numéros de lignes sont ajoutés au rapport.

Utilisation de DFA avec WINSPECTR.BIN

Lorsque DFA est utilisé avec le fichier WINSPECTR.BIN , il réalise les traitements suivants :

- Ajout des variables de pile au fichier journal (variables locales, paramètres transmis à la fonction, structures et tableaux).
- Mention des types de variables, des valeurs ainsi que des adresses par fonction.

De plus, si les informations de Turbo Debugger sont disponibles, DFA indique pour chaque cadre de pile :

- Dans la première section :
 - Le fichier source.
 - Le numéro de ligne.
 - Les variables locales.
 - Les paramètres.
- Dans la deuxième section :
 - Le nom du module pour la tâche défaillante.
 - Le nom des fichiers.
 - Les segments logiques.
 - Les sélecteurs de segments.
 - Le type des segments : données ou code .
- Dans la troisième section :
 - Les variables globales.
 - Les variables statiques.
 - Les valeurs des variables au moment de l'erreur.

Format DFA [option] WINSPECTR.LOG [WINSPECTR.BIN]

Si le fichier WINSPECTR.LOG (fichier obligatoire) existe, vous n'obtenez que les informations suivantes : nom du fichier source et numéros de ligne. Si WINSPECTR.BIN (fichier facultatif) existe, vous obtenez les informations complémentaires concernant les variables.

Tableau 8.2
Options de DFA

Option	Résultat
<i>/O</i> <i>fichier_sortie</i>	Renomme le fichier de sortie (DFA.OUT est le nom implicite)
<i>/D</i>	Demande à DFA d'écrire une liste hexadécimale des segments de données sauvegardées.

Autres outils de WinSpector

Vous pouvez utiliser les trois utilitaires ci-dessous pour enrichir encore les informations que WinSpector fournit pour une exception :

- EXEMAP crée un fichier .MAP à partir d'un fichier .EXE Windows. Le fichier .MAP est requis pour la création du fichier .SYM, qui améliore le compte-rendu d'erreurs pour l'EXE original.
- TMAPSYM, utilisé avec EXEMAP.EXE, crée un fichier .SYM depuis un fichier .MAP.
- BUILDSYM utilise EXEMAP.EXE et TMAPSYM.EXE pour créer un fichier .SYM depuis un fichier .EXE Windows.

Utilisation de EXEMAP.EXE

EXEMAP crée des fichiers .MAP des exécutables Windows. Un fichier .MAP peut être utilisé pour créer un fichier .SYM. WinSpector peut alors utiliser ce fichier pour enrichir le rapport d'erreurs. Si vous utilisez des DLL ou d'autres programmes dont vous ne disposez pas du code source, ces informations peuvent s'avérer particulièrement utiles.

Pour créer un fichier .MAP depuis un fichier .EXE, tapez la commande `EXEMAP fichier.EXE nouveaunom.MAP`. Si le second fichier est omis, le nom du fichier .EXE est conservé pour le fichier .MAP.

Bien que le fichier .MAP résultant ne soit pas aussi complet que celui qui est généré lors de la phase de lien du processus de compilation, il contient les adresses des fonctions publiques exportées.

Utilisation de TMAPSYM

TMAPSYM crée un fichier .SYM à partir d'un fichier .MAP existant (lequel est créé à l'aide du compilateur ou de l'utilitaire EXEMAP). Le fichier .SYM résultant permet à WinSpector d'utiliser les fonctions publiques, les noms de variables ainsi que les fonctions figurant dans la table d'entrée de l'exécutable. Toutefois, les informations concernant les constantes et les numéros des lignes ne sont pas incluses dans ce fichier.

Pour créer un fichier .SYM à partir d'un fichier .MAP, tapez `TMAPSYM fichier.MAP` (l'extension .MAP est obligatoire).

**Utilisation de
BUILDSYM**

BUILDSYM crée des fichiers .SYM à partir de fichiers .EXE. La sortie générée est la même qu'en utilisant à la fois EXEMAP et TMAPSYM, puisqu'il charge ces deux utilitaires, mais il supprime le fichier .MAP de votre répertoire. BUILDSYM reconnaît les caractères jokers ? et *, ce qui permet de traiter tout un répertoire en une seule commande.

Pour exécuter BUILDSYM, les deux utilitaires EXEMAP et TMAPSYM doivent être accessibles. BUILDSYM stocke les fichiers .SYM résultants dans le répertoire actif. Pour que WinSpector puisse trouver un fichier .SYM, celui-ci doit se trouver dans le même répertoire que celui de l'exécutable qui cause l'erreur.

BUILDSYM exécute les tâches suivantes :

- Il vérifie que les fichiers sont effectivement des fichiers Windows. Dans le cas contraire, il ne crée aucun fichier .MAP ou .SYM pour ces fichiers.
- Il lance EXEMAP pour créer les fichiers .MAP.
- Il s'assure que les fichiers .MAP ont bien été créés.
- Il lance TMAPSYM et transmet les noms des nouveaux fichiers .MAP afin de créer les fichiers .SYM.
- Il supprime les fichiers .MAP puisqu'ils ne sont plus d'aucune utilité.

Pour créer un fichier .SYM à partir d'un .EXE, tapez BUILDSYM
fichier.EXE.

Vous pouvez utiliser les caractères jokers habituels dans la partie *fichier* de la commande. Par exemple, tapez BUILDSYM * .EXE pour créer des fichiers .SYM pour tous les fichiers .EXE du répertoire actif.

Utilisation de l'éditeur de liens TLINK

L'annexe A donne la liste des messages de l'éditeur de liens générés par TLINK et par l'éditeur de liens intégré de l'EDI.

TLINK et TLINK32 sont des outils de ligne de commande qui associent des modules objets (fichiers .OBJ) et des modules bibliothèques (fichiers .LIB) pour produire des fichiers exécutables. L'EDI utilise des versions intégrées d'éditeurs de liens. Du fait que le compilateur appelle automatiquement l'éditeur de liens, vous n'avez pas besoin d'utiliser TLINK *sauf* si vous supprimez l'étape d'édition de liens de la compilation (reportez-vous à l'option **-c** du compilateur). Sauf indication contraire, les instructions et les options de TLINK s'appliquent également à TLINK32.

Notions de base sur TLINK

Les options de TLINK et TLINK32 sont sensibles à la différence majuscules/minuscules.

TLINK utilise un fichier de configuration intitulé TLINK.CFG, un fichier de recherche (facultatif), et des options de lignes de commandes pour lier des modules objets, des bibliothèques et des ressources en vue de créer un fichier exécutable (.EXE ou .DLL). L'éditeur de liens de l'EDI utilise les options indiquées dans la boîte de dialogue Project Options de la section Linker. La syntaxe de TLINK est la suivante :

```
TLINK [@f_rech][options] f_démarr objetss, nom_exe,  
[f_map],[biblios] bib_runtime [import], [f_def], [f_res]
```

où

- *options* sont des options de TLINK qui commandent le fonctionnement de TLINK. Par exemple des options permettent d'indiquer si on veut produire un fichier .EXE ou DLL. Les options de TLINK doivent être précédées soit d'une barre oblique (/), soit d'un trait d'union (-). Pour désactiver une option par défaut, mettez un trait d'union à la suite de l'option (par exemple -P-). Les options TLINK sont énumérées dans le tableau 9.3.

Reportez-vous aux tableaux 9.1 et 9.2 pour savoir quel module utiliser.

- *f_démarr* est un module d'initialisation Borland pour fichiers exécutables ou fichiers DLL, qui s'occupe du classement des divers segments du programme. Le module d'initialisation doit venir en premier dans la liste du fichier objet. S'il n'est pas en première position, les segments du programme risquent de ne pas être placés au bon endroit dans la mémoire et de provoquer des bugs désagréables. Si le module d'initialisation n'est pas correctement enchaîné, vous risquez d'avoir une longue liste de messages d'erreur vous indiquant que certains identificateurs n'ont pas été traduits ou qu'aucune pile n'a été créée.
- *objets* sont les fichiers .OBJ que vous voulez enchaîner. Indiquez le chemin d'accès si ces fichiers ne se trouvent pas dans le répertoire en cours.
- *nom_exe* est le nom que vous voulez donner au fichier exécutable (.EXE or.DLL). Si vous ne précisez pas le nom du fichier exécutable, TLINK lui donne un nom en ajoutant .EXE ou .DLL au nom du premier fichier objet de la liste. Veillez à donner un nom explicite au fichier exécutable sur la ligne de commande de TLINK. Sinon, le nom de votre programme sera du type C02.EXE, qui ne correspondra probablement pas à ce que vous souhaitez.
- *f_map* (facultatif) est le nom que vous voulez donner au fichier de mappage. Si vous n'indiquez pas de nom, le nom du fichier est celui d'*exefile*(mais avec l'extension .MAP).
- *biblios* (facultatif) sont les fichiers bibliothèques que vous voulez inclure au moment d'éditer les liens. Si ces fichiers ne sont pas dans le répertoire ou le chemin de recherche en cours (voir l'option /L), vous devez indiquer leur chemin d'accès.
- *bib_runtime* est la bibliothèque d'exécution Borland. Si aucune bibliothèque n'est incluse, aucune ne sera enchaînée.
- *import* est la bibliothèque d'importation Windows qui permet d'accéder aux fonctions API de Microsoft Windows.
- *f_def* est le fichier de définition des modules (.DEF) des fichiers exécutables Windows. Si vous n'indiquez pas de fichier .DEF, TLINK crée une application basée sur des paramètres par défaut.
- *f_res* sont une liste de fichiers .RES à enchaîner au fichier exécutable.

TLINK utilise par défaut ou ajoute les extensions suivantes aux noms de fichiers qui n'en ont pas :

- .OBJ pour les fichiers objets
- .EXE pour les fichiers exécutables
- .DLL pour les bibliothèques à liaisons dynamiques
- .MAP pour les fichiers de mappage
- .LIB pour les fichiers bibliothèques
- .DEF pour les fichiers de définition de modules
- .RES pour les fichiers de ressources

TLINK.CFG

L'EDI utilise les options d'édition de liens indiquées dans les options du projet et les feuilles de style.

Reportez-vous au chapitre 2 pour plus d'informations sur les valeurs à donner aux options du projet.

TLINK utilise un fichier de configuration intitulé TLINK.CFG (ou TLINK32.CFG) pour les options que vous devriez normalement taper sur la ligne de commande (notez que TLINK.CFG ne peut contenir que des options, pas de noms de fichiers). Les fichiers de configuration vous permettent de sauvegarder les options que vous utilisez souvent pour vous éviter de les retaper à chaque fois.

TLINK cherche TLINK.CFG dans le répertoire en cours, puis dans le répertoire à partir duquel TLINK a été chargé.

Le fichier TLINK.CFG suivant dit à TLINK de chercher les bibliothèques d'abord dans le répertoire C:\BC4\LIB puis dans C:\WINAPPS\LIB, pour inclure des informations de débogage dans les fichiers exécutables qu'il crée, pour créer un mappe détaillé des segments et pour produire un fichier Windows exécutable (.EXE et non .DLL).

TLINK

```
/Lc:\bc4\lib;c:\winapps\lib  
/v /s  
/Twe
```

TLINK32

```
/Lc:\bc4\lib;c:\winapps\lib  
/v /s  
/Tpe
```

Fichiers de recherche

Les compilateurs en ligne de commande utilisent également des fichiers de recherche. Reportez-vous à la section "Fichiers de recherche" du chapitre 3 pour plus d'informations.

Les fichiers de recherche sont des fichiers ASCII qui contiennent les options et noms de fichiers destinés à TLINK.EXE (et TLINK32.EXE) que vous devriez normalement taper sur la ligne de commande. Les fichiers de recherche vous permettent d'avoir une ligne de commande plus longue que ne le permettent la plupart des systèmes d'exploitation. Ils peuvent contenir les mêmes informations que les fichiers de configuration (options de lignes de commande), mais peuvent en plus contenir des noms de fichiers.

Contrairement aux lignes de commande, les fichiers de recherche peuvent contenir plusieurs lignes. Pour avoir plusieurs lignes dans votre fichier de recherche, terminez chaque ligne par un caractère (+). Notez que si une ligne se termine par une option activée par le signe plus (par exemple /**v+**), le + n'est pas considéré comme un caractère de continuation de ligne (pour prolonger la ligne, utilisez /**v++**).

Si, à l'intérieur d'un fichier de recherche, vous séparez des éléments de la ligne de commande (par exemple des fichiers .OBJ et des fichiers .LIB) par des lignes, vous ne devez pas taper la virgule qui les sépare normalement quand ils sont sur une même ligne de commande. Par exemple :

```
/c c0ws+
    monprog,monexe
    monmap
    mabib cws
```

Cette syntaxe omet la virgule que vous devriez taper si les informations étaient toutes sur la même ligne, comme ceci :

```
TLINK /c c0ws monprog,monexe,monmap,mabib cws
```

Pour utiliser des fichiers de recherche,

1. Tapez les options de lignes de commande et les noms des fichiers dans un fichier texte ASCII que vous sauvegardez.
2. Tapez TLINK @@[chemin]F_RECH.RSP, où F_RECH.RSP est le nom de votre fichier de recherche.

Vous pouvez indiquer plusieurs fichiers de recherche, comme ceci :

```
tlink /c @@listobjs,monexe,monmap,@@listlibs
```

Si vous utilisez un fichier de recherche en plus d'options de lignes de commande, ces dernières se substitueront à toute option du fichier de recherche. Par exemple, si vous insérez un -v dans un fichier de recherche mais utilisez -v- sur la ligne de commande, TLINK utilise l'option de ligne de commande -v-.

Utilisation de TLINK avec BCC.EXE

BCC lance toujours TLINK avec l'option /c (différence majuscules/minuscules).

Vous pouvez transmettre des options et des fichiers à TLINK par l'intermédiaire des compilateurs en ligne de commande (BCC.EXE et BCC32.EXE), en tapant les noms de fichiers sur la ligne de commande, sans oublier leur extension .OBJ et .LIB. Par exemple,

```
BCC princip.obj sub1.obj mabib.lib
```

effectue l'édition de liens PRINCIP.OBJ, SUB1.OBJ et MABIB.LIB, puis produit le fichier exécutable PRINCIP.EXE.

BCC lance TLINK avec les fichiers C0WS.OBJ, CWS.LIB et IMPORT.LIB (module d'initialisation, bibliothèque d'exécution et bibliothèque d'importation Windows). BCC32 lance TLINK32 avec les fichiers C0W32.OBJ, CW32.LIB et IMPORT32.LIB par défaut.

Edition de liens pour les bibliothèques

Vous devez obligatoirement enchaîner la bibliothèque d'exécution Borland C++ qui contient les fonctions de bibliothèque C/C++ standard pour le type d'application que vous êtes en train d'enchaîner. Vous devez aussi insérer la bibliothèque d'importation qui convient (IMPORT.LIB pour les applications Windows 16 bits, IMPORT32.LIB pour les applications machine, ou IMPRTW32.LIB pour les applications Windows 32 bits).

Le tableau 9.1 décrit les bibliothèques Windows 3.x 16 bits et les fichiers .OBJ fournis par Borland. Reportez-vous au *Guide de référence* pour avoir la liste complète des bibliothèques Windows, et au *Guide de référence DOS* pour la liste complète des bibliothèques et fichiers de démarrage DOS.

Tableau 9.1
Bibliothèques 16 bits et
fichiers de démarrage
Borland

Bibliothèques et fichiers .OBJ	Description
Cn.LIB	Bibliothèque d'exécution pour applications DOS où <i>n</i> est S, C, M, L ou H pour désigner le modèle de mémoire : SMALL, COMPACT, MEDIUM, LARGE ou HUGE.
CWn.LIB	Bibliothèque d'exécution pour applications Windows 3.x, où <i>n</i> est S, C, M ou L pour désigner le modèle de mémoire : SMALL, COMPACT, MEDIUM ou LARGE.
CRTL DLL.LIB	Bibliothèque d'exécution pour applications Windows 3.x à enchaîner comme fichier .DLL.
IMPORT.LIB	Bibliothèque d'importation pour fonctions API de Windows 3.x.
C0n.OBJ	Code de démarrage pour applications .EXE DOS où <i>n</i> est T, S, C, M, L ou H pour désigner le modèle de mémoire : TINY, SMALL, COMPACT, MEDIUM, LARGE ou HUGE.

Tableau 9.1
Bibliothèques 16 bits et
fichiers de démarrage
Borland

Bibliothèques et fichiers .OBJ	Description
C0Wn.OBJ	Code de démarrage pour applications Windows 3.x où <i>n</i> est S, M, C ou L pour désigner le modèle de mémoire : SMALL, MEDIUM, COMPACT ou LARGE.
C0Dn.OBJ	Code de démarrage pour modules .DDL Windows 3.x où <i>n</i> est S, M ou L pour désigner le modèle de mémoire : SMALL, MEDIUM ou LARGE.
MATHWS.LIB	Si votre programme utilise des virgules flottantes, vous devez inclure une bibliothèque mathématique. MATHWS.LIB est destinée aux modèles SMALL et TINY.
MATHWC.LIB	Bibliothèque mathématique pour modèles COMPACT.
MATHWM.LIB	Bibliothèque mathématique pour modèles MEDIUM.
MATHWL.LIB	Bibliothèque mathématique pour modèles LARGE.

Le tableau 9.2 décrit les bibliothèques et fichiers .OBJ 32 bits fournis par Borland ; ils sont utilisés par TLINK32. Reportez-vous au *Guide de référence* pour avoir la liste complète des bibliothèques.

Tableau 9.2
Bibliothèques et
fichiers de
démarrage Borland
32 bits

Bibliothèques et fichiers .OBJ	Description
CW32.LIB	Bibliothèque d'exécution pour applications Win32.
IMPORT32.LIB	Bibliothèque d'importation pour applications machine et applications Windows 32 bits.
C0X32.OBJ	Code de démarrage pour applications machine.
C0W32.OBJ	Code de démarrage pour applications Win32.
C0D32.OBJ	Code de démarrage pour modules DLL 32 bits.

Options de TLINK

Sauf indication contraire, les options fonctionnent à la fois avec TLINK et avec TLINK32. Elles sont sensibles à la différence majuscules/minuscules et doivent être précédées d'une barre oblique (/) ou d'un trait d'union (-). Pour désactiver une option par défaut, mettez un trait d'union à la suite de l'option sur la ligne de commande (par exemple, **-P-** ou **/P-**). Vous pouvez placer des options à n'importe quel endroit de la ligne de commande. Il n'est pas nécessaire de séparer les options par des espaces (**/m/f/c** est la même chose que **/m /f /c**), mais par contre un espace est obligatoire pour séparer options et fichiers.

Le tableau 9.3 énumère les options de ligne de commande de TLINK et les options EDI correspondantes (notez que toutes les options n'ont pas un équivalent dans l'EDI). Les options de ligne de commande par défaut sont signalées par le symbole suivant : ■. Vous trouverez à la suite du tableau des explications plus détaillées sur ces options, notamment le nom des options EDI.

Tableau 9.3 : Options TLINK

Par défaut	Option	Option de l'éditeur de liens EDI	Pour	Description
	/3	Linker 16-bits Linker Enable 32-bit processing	16 bits	Accepte et enchaîne un code 32 bits produit par TASM ou un assembleur compatible.
	/ax	Target Attributes Target Model <néant>	32 bits	Indique le type d'application, où /aa cible des applications Windows /ap cible des applications machine.
	/A:dd	16-bit Segment Alignment	16/32 bits	Indique l'alignement des pages à l'intérieur du fichier .EXE.
	/B:xxxxxx	32-bit Linker Image based address	32 bits	Indique l'adresse de base des images (en hexadécimal).
	/c	General Case-sensitive link	16/32 bits	Considère les caractères comme significatifs dans les symboles.
	/C	General Case-sensitive exports, imports	16 bits	Considère les caractères comme significatifs dans les sections EXPORTS et IMPORTS du fichier de définition des modules.
	/d	Warnings Warn duplicate symbol in .LIB	16 bits	Vous avertit en cas de symboles en double dans des bibliothèques.

Tableau 9.3 : Options TLINK (suite)

Par défaut	Option	Option de l'éditeur de liens EDI	Pour	Description
	/c	General Case-sensitive link	16/32 bits	Considère les caractères comme significatifs dans les symboles.
	/C	General Case-sensitive exports, imports	16 bits	Considère les caractères comme significatifs dans les sections EXPORTS et IMPORTS du fichier de définition des modules.
	/d	Warnings Warn duplicate symbol in .LIB	16 bits	Vous avertis en cas de symboles en double dans des bibliothèques.
	/E	16-bit Linker Process extended dictionaries	16 bits	Permet le traitement de dictionnaires étendus dans les bibliothèques.
	/Enn	32-bit Linker Maximum linker errors	32 bits	Indique le nombre maximum d'erreurs avant interruption.
■	/e	16-bit Process extended dictionaries (unchecked)	16 bits	Ignore les dictionnaires étendus dans les bibliothèques. C'est le contraire de l'option /E.
	/f	16-bit Linker Inhibit optimizing far to near	16 bits	Invalide l'optimisation des appels far à des données near.
	/Gx		16/32 bits	"Super" options où x est n, r, ou m.
	/Gn	16-bit Linker Discard nonresident name table	16 bits	Ne tient pas compte des tables de noms non-résidents.
	/Gr	16-bit Transfer resident to nonresident table	16 bits	Transfère les noms résidents dans une table de noms non-résidents.
	/Gm	Map File Print mangled names in map file	16/32 bits	Met les noms non significatifs dans un fichier mappe.
	/i	16-bit Linker Initialize segments	16 bits	Initialise tous les segments.
	/l	Map File Include source line numbers	16 bits	Insère des numéros de ligne source (L minuscule).
	/L	Directories Library (pas dans Linker de l'EDI)	16/32 bits	Indique les chemins d'accès des bibliothèques.
	/m	Map File Public	16/32 bits	Crée un fichier de mappe avec accès publics.
	/n	General Default Libraries	16 bits	Ne pas utiliser les bibliothèques par défaut.
	/o	Overlay module (Boîte de dialogue des attributs de noeuds)	16 bits	Modules ou bibliothèques de recouvrement.
■	/P	General Pack code segments	16 bits	Stockage de segments de codes.
	/Rk	Resource Pack fast load area (not under Linker)	16 bits	Envoi des options à RLINK.EXE.
	/Rv	<néant>	32 bits	Édition de liens avec des ressources verbeuses.

Tableau 9.3 : Options TLINK (suite)

Par défaut	Option	Option de l'éditeur de liens EDI	Pour	Description
	/Re xxxx	<néant>	32 bits	Renomme le fichier exécutable : xxxx.
	/S :xxxxxx	32-bit Linker Stack size	32 bits	Indique la taille de la pile (en hexadécimal).
	/s	Map File Detailed	16 bits	Crée un mappe détaillé des segments.
	/t	<néant>	16 bits	Crée un fichier .COM DOS de modèle TINY.
	/Tdx	16 bits <néant>		Indique l'application cible où /Tdc signifie créer un fichier .COM DOS. /Tde signifie créer un fichier .EXE DOS.
■		TargetExpert Platform		
	/Tpx	TargetExpert Platform	32 bits	Indique une application cible, où /Tpe signifie créer un fichier .EXE 32 bits. /Tpd signifie créer une DLL 32 bits.
■				
	/Twx	TargetExpert Target Type	16 bits	Indique une application Windows 3.x, où /Twe crée un fichier .EXE Windows. /Twd crée une DLL Windows.
■				
	/v	General Include debug information	16/32 bits	Inclut des informations de débogage symboliques complètes.
	/w xxx	Warnings (voir tableau 9.4).	32 bits	Active ou désactive les mises en garde (voir tableau 9.4).
	/x	Map File Off	16/32 bits	Ne crée pas de fichier de mappe.
	/ye	<néant>	16 bits	Utilise la mémoire paginée pour les permutations.
	/yx	<néant>	16 bits	Configure l'utilisation par TLINK de la permutation avec la mémoire paginée.

/3 vous permet d'enchaîner des modules objets DOS 32 bits produits par TASM ou un assembleur compatible. Cette option augmente la quantité de mémoire requise pour TLINK et ralentit la procédure d'édition de liens.



/a vous permet d'indiquer le type d'image EXE :

- **/aa** cible des applications Windows.
- **/ap** cible des applications machine qui peuvent être exécutées dans une fenêtre.

/A:dd indique l'alignement des pages pour les codes et données du fichier exécutable *dd* doit être une puissance décimale de 2. Par exemple, si vous indiquez un alignement de **/A:12**, les sections de l'image sont enregistrées sur les limites de 4096 octets. Le système d'exploitation se base sur cette valeur pour chercher des pages à charger. La valeur par défaut est **/A:9**, ce qui signifie que les sections sont alignées sur les limites de 512 octets à l'intérieur du fichier exécutable.



/B:xxxxxx indique une adresse de base pour les images d'une application donnée. Si cette option est utilisée, les relocations internes sont supprimées de l'image et l'adresse de chargement du premier objet demandé prend le numéro hexadécimal donné avec l'option. Tous les objets qui suivent sont alignés sur les limites d'adresse linéaires de 64 ko. Cette option réduit la place occupée par les applications sur le disque et améliore à la fois le temps de chargement et les performances d'exécution, dans la mesure où le système d'exploitation n'a plus besoin de procéder à des relocations internes. Du fait que NT charge toutes les images .EXE à 64 ko, nous vous conseillons d'enchaîner tous les fichiers .EXE à l'aide de **/B:0x10000**.

/c rend les majuscules et minuscules significatives dans les symboles publics et externes.



/C rend les majuscules et minuscules significatives dans les sections EXPORTS et IMPORTS des fichiers de définition de modules.



/d vous avertit si un symbole apparaît dans plusieurs fichiers bibliothèques. Si le symbole doit être inséré dans le programme, TLINK utilise le symbole du premier fichier contenant le symbole indiqué sur la ligne de commande (ou dans un fichier de recherche). Cette option vous avertit également si des symboles apparaissent à la fois dans un fichier .OBJ et dans un fichier .LIB (TLINK utilise le premier fichier enchaîné et ignore les autres).



/Enn précise le nombre maximum d'erreurs indiqué par l'éditeur de liens avant interruption. **/EO** (par défaut) indique un nombre infini d'erreurs (c'est-à-dire autant que possible).



/E traite les dictionnaires étendus. Les fichiers bibliothèques de Borland C++ contiennent un dictionnaire étendu avec des informations permettant à TLINK d'utiliser moins de mémoire et d'éditer plus rapidement des liens avec ces bibliothèques. Vous pouvez ajouter le dictionnaire étendu dans d'autres fichiers bibliothèques à l'aide de l'option **/E** de TLIB (reportez-vous à la section sur TLIB dans le chapitre 11). Evitez d'utiliser **/E** avec des programmes ayant des liens avec des bibliothèques créées sans dictionnaire étendu (bibliothèques de tiers fournies sans le code source, par exemple). Pour pouvoir utiliser les dictionnaires étendus, il faut que toutes les bibliothèques liées aient un dictionnaire étendu.



/e ne tient pas compte des dictionnaires étendus. C'est le contraire de l'option **/E**, et c'est la valeur par défaut.



/f invalide l'optimisation des appels far à des données near.

/Gx sont des options où *x* peut être

n = Ignorer la table des noms non-résidents.

r = Transférer les noms résidents dans une table de non-résidents.

m = (TLINK32) Met les noms non significatifs dans le fichier de mappe ; peut vous aider à savoir comment les noms sont modifiés.



/i envoie les segments de fin non initialisés dans le fichier exécutable, même si ces segments ne contiennent pas d'enregistrement.



/l crée une section dans le fichier .MAP pour les numéros de ligne du code source. Les fichiers .OBJ enchaînés doivent être compilés avec les informations de débogage à l'aide de **-y** ou **-v**. Si vous utilisez **/x** pour supprimer la création du fichier de mappe, l'option **/l** (L minuscule) est sans effet.

/L vous permet d'afficher la liste des répertoires que TLINK explorera si vous ne tapez pas le nom de chemin explicite d'une bibliothèque ou du module d'initialisation C ou C++. TLINK explore en premier le répertoire en cours (dans lequel vous avez tapé TLINK). Par exemple,

```
TLINK /Lc:\BC4\lib;c:\mesbibs splash logo,,utils .\logolib
```

explore d'abord le répertoire en cours pour trouver UTILS.LIB, explore ensuite C:\BC4\LIB, puis C:\MESBIBS. Dans la mesure où LOGOLIB désigne explicitement le nom du répertoire en cours, TLINK n'explore pas les bibliothèques indiquées avec l'option **/L** pour trouver LOGOLIB.LIB.

/m crée un mappe plus complet que celui normalement réalisé par TLINK, en ajoutant dans le mappe une liste triée de symboles publics. Ce type de fichier de mappe est utile pour le débogage. De nombreux débogueurs peuvent utiliser la liste des symboles publics, ce qui vous permet de faire référence à des adresses symboliques quand vous déboguez. Si vous ne définissez aucune option du fichier de mappe (**/m**, **/s**, ou **/x**), c'est l'option Map File | Segments qui est utilisée. Voir également **/s**.



/M crée un mappe avec noms publics non significatifs.



/n ignore les bibliothèques par défaut indiquées par certains compilateurs. Utilisez cette option pour enchaîner des modules qui sont écrits dans un autre langage.

/o active le recouvrement pour le code dans tous les modules ou bibliothèques qui suivent l'option sur la ligne de commande (cette option ne fonctionne que pour les applications DOS). Utilisez **/o-** sur la ligne de commande pour désactiver cette fonction. Si vous indiquez un nom de classe à la suite de cette option, tous les segments de cette classe seront recouverts (vous pouvez le faire pour plusieurs classes). Si vous n'indiquez pas de nom à la suite de l'option, tous les segments des classes se terminant par CODE seront recouverts. Cette option utilise le numéro d'interruption de recouvrement par défaut 3FH. Pour indiquer un autre numéro d'interruption, utilisez **/o#xx**, où **xx** est un nombre hexadécimal à deux chiffres.



/P réunit autant de segments de code qu'il est possible dans un segment physique, dans la limite des possibilités de stockage de segments de code (jamais au-delà). TLINK crée un nouveau segment si nécessaire. La limite de stockage par défaut de segments de code est de 8 192 octets (8 ko). Pour la modifier, utilisez **/P=n** où *n* indique un nombre d'octets entre 1 et 65 536. Vous préférerez sans doute que cette limite soit un multiple de 4 ko en mode 386 renforcé, en raison de la granularité des unités de mémoire à accès direct du système.

Bien que la taille optimale des segments en mode amélioré 386 soit de 4 ko, la taille de stockage par défaut est de 8 ko du fait que les segments de code types sont de 4 à 8 ko et qu'il est plus facile de stocker 8 ko.

Du fait que chaque segment est supervisé par un système, le stockage des segments de code améliore en principe les performances. **/P-** désactive le stockage des segments de code (utile si vous l'avez activé dans le fichier de configuration et voulez le désactiver pour un lien donné).



/s crée un fichier de mappe avec les mêmes fonctions que l'option **/m**, mais ajoute un mappe de segment détaillé. Si vous ne définissez pas la valeur des options (**/m**, **/s**, ou **/x**), c'est l'option MapFile | Segments qui est utilisée. Pour chaque segment de chaque module, ce fichier de mappe inclut l'adresse, la longueur en octets, la classe, le nom du segment, le groupe, le module et les informations ACBP. Si le même segment apparaît dans plusieurs modules, chaque module fait l'objet d'une ligne séparée. A l'exception du champ ACBP, les informations du mappe détaillé des segments sont très explicites.

Le champ ACBP code les attributs A (alignement), C (combinaison) et B (big) en un groupe de champs de quatre bits, tel que défini par Intel. TLINK n'utilise que trois de ces champs, les champs A, C et B. La valeur d'ACBP dans le mappe est imprimée en hexadécimal. Les valeurs de champ qui suivent doivent être réunies par des OR pour atteindre la valeur ACBP imprimée.

Champ	Valeur	Description
Champ A (alignement)	00	Segment absolu.
	20	Segment aligné sur les octets.
	40	Segment aligné sur les mots.
	60	Segment aligné sur les paragraphes.
	80	Segment aligné sur les pages.
	A0	Portion de mémoire absolue sans nom.
Champ C (combinaison)	00	Ne peut être combiné.
	08	Segment de combinaison public.
Champ B (big)	00	Segment de moins de 64 ko.
	02	Segment de 64 ko exactement.

Avec l'option **/S** les symboles publics sans référence sont étiquetés "inactifs". Un symbole inactif est un symbole de module publiquement défini auquel il n'a pas été fait référence par un enregistrement EXTDEF d'un autre module inclus dans le lien. Par exemple, ce fragment de la section de symboles publics d'un fichier de mappe montre qu'il n'est pas fait référence aux symboles *Symbol1* et *Symbol3* par l'image qui leur est liée :

```
0002:00000874      Inactif      Symbol1
0002:00000CE4      Inactif      Symbol2
0002:000000E7      Inactif      Symbol3
```



/S:xxxxxx définit la taille en hexadécimal de la pile de l'application où *xxxxxx* est une chaîne hexadécimale. Le fait d'indiquer la taille de la pile avec **/S** annule l'effet de tout réglage de la taille de la pile dans un fichier de définition de modules.

/t crée un fichier .COM DOS de modèle TINY (cette option fonctionne comme **/Tdc**, à cette exception près que vous pouvez utiliser **/t** avec BCC.EXE). Les fichiers .COM DOS ne peuvent dépasser 64 ko, ont des relocations liées aux segments ou définissent un segment de pile. Ils doivent avoir une adresse de départ de 0:100H. Si vous voulez changer l'extension des fichiers (pour .BIN, par exemple), l'adresse de départ peut être soit 0:0, soit 0:100H. L'éditeur de liens ne peut générer d'informations de débogage pour les fichiers .COM ; il vous faudra donc les déboguer comme des fichiers .EXE, puis les recompiler et enchaîner comme des fichiers .COM.

/Tdx produit un fichier .EXE DOS (**/Tde**) ou un fichier .COM DOS (**/Tdc**).



/Tpx produit un fichier .EXE (**/Tpe**) ou .DLL (**/Tpd**) en mode protégé.



/Twx produit un fichier Windows .EXE (**/Twe**) ou .DLL (**/Twd**). Cette option n'est pas nécessaire si vous incluez un fichier de définition de modules avec une instruction EXETYPE Windows ; TLINK crée en effet une application(.EXE) si le fichier de définition de modules contient une instruction NAME, ou une DLL si le fichier de définition de modules contient une instruction LIBRARY.

/v insère des informations de débogage dans le fichier exécutable. Si cette option est trouvée quelque part sur la ligne de commande, les informations de débogage sont insérées dans le fichier exécutable pour tous les modules objets qui contiennent des informations de débogage. Vous pouvez utiliser les options **/v+** et **/v-** pour activer ou désactiver sélectivement les informations de débogage, module par module (mais pas sur la même ligne de commande que **/v**). Par exemple, la commande

```
TLINK mod1 /v+ mod2 mod3 /v- mod4
```

inclut des informations de débogage pour les modules *mod2* et *mod3*, mais ni pour *mod1* ni *mod4*.



/wxxx active (**/wxxx**) ou désactive (**/w-xxx**) les messages de mise en garde de TLINK, où *xxx* peuvent être (l'option par défaut implique que TLINK enverra la mise en garde sans que vous l'ayez spécialement activée) :

Tableau 9.4
Mises en garde
TLINK32 s

Par défaut	Option /w	Description EDI
■	bdl	Utilisation de liaisons implantées dans des DLL (peut provoquer un mauvais fonctionnement de la DLL).
	def	Pas de fichier .DEF ; utilisation des valeurs par défaut.
	dpl	Avertit de l'existence d'un symbole en double dans .LIB.
■	dup	Symbole en double (mise en garde pour les fichiers .OBJ).
■	ent	Pas de point d'entrée.
	imt	Import ne correspond pas à la définition précédente.
	inq	Extern non qualifié avec __import.
■	srf	Relocation différentielle dépassée. Mise en garde "Pas de pile".

/x dit à TLINK de ne pas générer de fichier de mappe. TLINK crée généralement des fichiers de mappe qui énumèrent les segments du programme, l'adresse de démarrage du programme et toute mise en garde ou tout message d'erreur TLINK (par défaut, l'option MapFile | Segments, qui n'a pas d'option de ligne de commande, est activée).



/ye commande l'utilisation par TLINK de la mémoire paginée pour les tampons d'E/S. Si TLINK a besoin d'espace mémoire supplémentaire pour les structures de données actives (en cours de lecture de fichiers objets ou d'écriture dans le fichier exécutable), soit il efface les tampons, soit il les permute avec la mémoire paginée.

Quand il lit des fichiers, TLINK efface le tampon d'entrée de sorte que l'espace disponible puisse être utilisé pour d'autres structures de données. Quand TLINK crée un exécutable, il met le tampon à la bonne place dans le fichier exécutable. Dans les deux cas, vous pouvez augmenter sensiblement la vitesse de l'édition de liens en permutant sur la mémoire paginée. Par défaut, la fonction sur la mémoire paginée est active et celle d'échange sur la mémoire étendue désactivée. Si l'échange est activé et s'il n'y a pas suffisamment d'espace mémoire, l'échange n'a pas lieu.

Cette option prend plusieurs formes qui sont indiquées ci-après ;

- **/ye** ou **/ye+** active sur la mémoire paginée (valeur par défaut).
- **ye-** désactive sur la mémoire paginée (inactive par défaut).



/yx commande l'utilisation par TLINK de la mémoire étendue pour les tampons d'E/S. Par défaut TLINK peut utiliser jusqu'à 8 Mo de mémoire étendue. Vous pouvez modifier l'utilisation de la mémoire étendue par TLINK en sélectionnant l'une des options suivantes :

- **/yx+** utilise toute la mémoire étendue disponible, dans la limite de 8 Mo.
- **/yx*n*** utilise la mémoire étendue uniquement dans la limite de *n* ko.

Fichiers de définition de modules

IMPDEF crée des fichiers de définition de modules, tandis que IMPLIB crée des bibliothèques d'importation à partir de fichiers de définition de modules.

Reportez-vous au chapitre 10 pour tout complément d'information sur ces outils.

Cette section décrit les fichiers de définition de modules et les instructions qui apparaissent dans ces fichiers. Les fichiers de définition de modules donnent à TLINK des informations sur le contenu et les exigences système d'une application Windows. Plus précisément, un fichier de définition de modules :

- Nomme les fichiers .EXE ou .DLL.
- Identifie le type d'application.
- Enumère les fonctions importées et exportées.
- Décrit les attributs des segments de code et de données, et vous permet d'insérer les attributs d'autres segments de code.
- Précise la taille de la pile.
- Permet l'intégration d'un programme de remplacement temporaire.

Reportez-vous au *Guide du programmeur* pour plus de détails sur l'utilisation et pour des exemples de fichiers de définition de modules.

Instruction CODE

CODE définit les attributs par défaut des segments de code. Les segments de code peuvent prendre n'importe quel nom, mais doivent appartenir à des classes de segments dont le nom se termine par CODE (par exemple CODE ou MYCODE). La syntaxe est la suivante :

TLINK	TLINK32
CODE [FIXED MOVEABLE]	[PRELOAD LOADONCALL]
[DISCARDABLE NONDISCARDABLE]	[EXECUTEONLY EXECUTEREAD]
[PRELOAD LOADONCALL]	

- FIXED (valeur par défaut) signifie que le segment reste toujours au même endroit dans la mémoire ; MOVEABLE signifie que le segment peut être déplacé.
- PRELOAD signifie que le code est chargé quand le programme d'appel est chargé. LOADONCALL (valeur par défaut) signifie que le code est chargé quand il est appelé par le programme.
- DISCARDABLE signifie que le segment peut être supprimé s'il n'est plus utile (implique qu'il soit MOVABLE). NONDISCARDABLE (valeur par défaut) signifie que le segment ne peut être supprimé.

- EXECUTEONLY signifie qu'un segment de code peut être uniquement exécuté. EXECUTEREAD (valeur par défaut) signifie que le segment de code peut être lu et exécuté.
- PRELOAD signifie que le segment est chargé quand le module est chargé pour la première fois. LOADONCALL (valeur par défaut) signifie que le segment est chargé quand il est appelé.

Instruction DATA

DATA définit les attributs des segments de données. La syntaxe est la suivante :

```
DATA [NONE | SINGLE | MULTIPLE]
     [READONLY | READWRITE]
     [PRELOAD | LOADONCALL]
     [SHARED | NONSHARED]
```

- NONE signifie qu'il n'y a pas eu de segment de données créé. Cette option n'est disponible que pour les bibliothèques. SINGLE (valeur par défaut des fichiers .DLL) signifie qu'un seul segment de données est créé et partagé par tous les processus. MULTIPLE (valeur par défaut pour les fichiers .EXE) signifie qu'un segment de données est créé pour chaque processus.
- READONLY signifie que le segment de données peut être uniquement lu. READWRITE (valeur par défaut) signifie que le segment de données est accessible en lecture et en écriture.
- PRELOAD signifie que le segment de données est chargé lors du premier chargement d'un module utilisant ce segment. LOADONCALL (valeur par défaut) signifie que le segment de données est chargé lorsqu'il est sollicité pour la première fois (ceci n'est pas vrai pour les applications 32 bits).
- SHARED (valeurs par défaut des fichiers .DLL 16 bits) signifie qu'un exemplaire du segment de données est partagé par tous les processus. NONSHARED (valeur par défaut pour les programmes et fichiers .DLL 32 bits) signifie qu'un exemplaire du segment de données est chargé pour chaque processus qui a besoin d'utiliser le segment.

**Instruction
DESCRIPTION**

DESCRIPTION (facultatif) insère du texte dans le module d'application ; cette instruction est en principe utilisée pour inclure le nom de l'auteur, la date ou des informations de copyright. La syntaxe est la suivante :

```
DESCRIPTION 'Texte'
```

Texte est une chaîne ASCII délimitée par des apostrophes.

**Instruction
EXETYPE**

EXETYPE définit le type d'en-têtes par défaut du fichier exécutable (.EXE) pour les applications 16 bits. Vous pouvez laisser cette section pour les applications 32 bits, pour des besoins de compatibilité amont, mais si vous êtes amené à changer EXETYPE, reportez-vous à l'instruction NAME, plus loin dans ce chapitre. La syntaxe d'EXETYPE est la suivante :

```
EXETYPE WINDOWS
```

**Instruction
EXPORTS**

EXPORTS définit le nom et les attributs des fonctions à exporter. Le mot réservé EXPORTS indique le début des définitions. Il peut être suivi par n'importe quel nombre de définitions d'exportation, chacune sur une ligne distincte. La syntaxe est la suivante :

```
EXPORTS  
    NomExport [Ordinal] [RESIDENTNAME] [Paramètre]
```

- *NomExport* est une chaîne ASCII définissant le symbole à exporter :

```
    NomEntrée [=NomInterne]
```

NomInterne est le nom utilisé à l'intérieur de l'application pour faire référence à cette entrée. *NomEntrée* est le nom qui figure dans la table des entrées du fichier exécutable ; il est visible à l'extérieur.

- *Ordinal* définit la valeur ordinale de la fonction, comme suit :

```
    @@ordinal
```

où *Ordinal* est un nombre entier indiquant le nombre ordinal de la fonction.

Quand une application ou un module DLL appelle une fonction exportée d'une DLL, le module appelant peut désigner la fonction par son nom ou par son numéro ordinal. Il est plus rapide d'appeler la fonction par son numéro ordinal car les comparaisons de chaînes ne sont pas nécessaires pour localiser la fonction. Pour économiser la mémoire, exportez les fonctions par leur nombre ordinal (du point de vue de la DLL de la fonction) et importez/appelez-les par leur numéro ordinal (du point de vue du module sollicitateur).

Quand une fonction est exportée par son numéro ordinal, son nom se trouve dans la table des nom non-résidents. Quand elle est exportée par son nom, le nom se trouve dans le table des noms résidents. La table des noms résidents d'un module donné est en mémoire chaque fois que ce module est chargé ; la table des noms non-résidents ne l'est pas.

- **RESIDENTNAME** indique que le nom de la fonction doit être en permanence résident. Ceci n'est utile que pour exporter par le numéro ordinal (si le nom n'était pas résident par défaut).
- *Paramètre* est un entier facultatif qui précise le nombre de mots que la fonction veut transmettre comme paramètres.

**Instruction
IMPORTS**

IMPORTS définit le nom et les attributs des fonctions à importer de DDL. Au lieu d'énumérer les fonctions à importer dans l'instruction **IMPORTS**, vous pouvez :

- Indiquer une bibliothèque d'importation DLL sur la ligne de commande **TLINK**,
ou
- Inclure la bibliothèque d'importation DDL dans le gestionnaire de projet de l'EDI.

Si vous programmez une application 32 bits, utilisez **__import** pour importer toute fonction, classe ou donnée recherchées ; pour les applications 16 bits, vous devez utiliser **__import** avec des classes. Reportez-vous au *Guide du programmeur* pour de plus amples informations sur l'utilisation de **__import**.

Le mot réservé IMPORTS signale le début des définitions ; il est suivi de n'importe quel nombre de définitions d'importation, chacune sur une ligne distincte. La syntaxe est la suivante :

```
IMPORTS
    [NomInterne=] NomModule.Entrée
```

- *NomInterne* est une chaîne ASCII qui précise le nom unique utilisé par l'application pour appeler la fonction.
- *NomModule* indique un ou plusieurs caractères ASCII majuscules qui définissent le nom du module exécutable contenant la fonction. Ce nom doit être le même que celui du fichier exécutable. Par exemple, le fichier SAMPLE.DLL a pour nom de module SAMPLE.
- *Entrée* désigne la fonction à importer, soit une chaîne ASCII désignant la fonction, soit un entier correspondant au nombre ordinal de la fonction.

**Instruction
LIBRARY**

LIBRARY définit le nom d'un module DLL. Un fichier de définition de modules peut contenir soit une instruction LIBRARY pour désigner une DLL, soit une instruction NAME pour désigner un programme.

Tout nom de module de bibliothèque doit correspondre au nom du fichier exécutable. Par exemple, la bibliothèque MABIB.DLL a pour nom de module MABIB. La syntaxe est la suivante :

```
LIBRARY NomBiblio [INITGLOBAL | INITINSTANCE]
```

- *NomBiblio* (facultatif) est une chaîne ASCII qui définit le nom du module de la bibliothèque. Si vous n'indiquez pas de *NomBiblio*, TLINK utilise le nom du fichier en supprimant l'extension. Si le fichier de définition de modules ne comporte ni instruction NAME ni instruction LIBRARY, TLINK utilise par défaut une instruction NAME sans paramètre *NomModule*.
- INITGLOBAL signifie que la routine d'initialisation de la bibliothèque n'est appelée que lorsque le module de bibliothèque est chargé pour la première fois dans la mémoire. INITINSTANCE signifie que la routine d'initialisation est appelée chaque fois qu'un nouveau processus utilise la bibliothèque.

Instruction NAME

NAME est le nom du module exécutable de l'application. Le nom du module permet d'identifier le module quand vous exportez des fonctions. Pour les applications 32 bits, NAME doit figurer avant EXETYPE. Si NAME et EXETYPE n'indiquent pas le même type d'application, c'est le type indiqué par NAME qui est utilisé. La syntaxe est la suivante :

```
NAME NomModule [WINDOWSAPI] | [WINDOWCOMPAT]
```

NomModule (facultatif) correspond à un ou plusieurs caractères ASCII majuscules désignant le module exécutable. Ce nom doit correspondre au nom du fichier exécutable. Par exemple, une application avec le fichier exécutable SAMPLE.EXE aura pour nom de module SAMPLE.

Si *NomModule* n'est pas indiqué, TLINK considère que le nom du module correspond au nom du fichier exécutable. Par exemple, si vous n'indiquez pas de nom de module et si le fichier exécutable s'appelle MYAPP.EXE, TLINK considère que le nom du module est MYAPP.

Si le fichier de définition de modules ne comporte ni instruction NAME ni instruction LIBRARY, TLINK utilise par défaut une instruction NAME sans paramètre *NomModule*.

WINDOWAPI est un fichier Windows exécutable équivalant à l'option TLINK32 /**aa**.

WINDOWCOMPAT est un fichier exécutable en mode caractères compatible Windows, équivalant à l'option TLINK32 /**ap**.

Instruction SEGMENTS

SEGMENTS définit les attributs des segments supplémentaires de code et de données. La syntaxe est la suivante :

```
SEGMENTS  
    NomSegment [CLASS 'NomClasse'] [MinAlloc]  
    [SHARED | NONSHARED]  
    [PRELOAD | LOADONCALL]
```

- *NomSegment* est une chaîne de caractères qui désigne le nouveau segment. Cela peut être n'importe quel nom, y compris les noms de segments standard `_TEXT` et `_DATA` qui correspondent aux segments de code et de données standard.
- *NomClasse* (facultatif) est le nom de classe du segment concerné. Si le nom de classe n'est pas indiqué, TLINK utilise le nom de classe `CODE`.

- *MinAlloc* (facultatif) est un entier désignant le minimum de mémoire requis pour le segment. TLINK et TLINK32 ne tiennent pas compte de cette valeur.
- SHARED (valeur par défaut pour les fichiers .DLL 16 bits) signifie qu'un exemplaire du segment est partagé par plusieurs processus. NONSHARED (valeur par défaut des programmes et fichiers.DLL 32 bits) signifie qu'un exemplaire du segment est chargé pour chaque processus qui a besoin du segment de données.
- PRELOAD signifie que le segment est chargé immédiatement ; LOADONCALL que le segment est chargé quand il est sollicité ou appelé (non pris en compte par TLINK32). Le compilateur de ressources peut annuler l'effet de l'option LOADONCALL et précharger des segments.

**Instruction
STACKSIZE**

STACKSIZE définit le nombre d'octets dont l'application a besoin pour sa pile locale. L'application utilise la pile locale chaque fois qu'elle appelle des fonctions. N'utilisez pas l'instruction STACK-SIZE pour les bibliothèques à liaisons dynamiques. La syntaxe est la suivante :

```
STACKSIZE octets
```

octets est la taille de la pile en nombre d'octets.

Instruction STUB

STUB ajoute un fichier DOS exécutable désigné par *NomFichier* au début du module. Le fichier de remplacement temporaire exécutable affiche un message de mise en garde et s'interrompt si l'utilisateur essaie de l'exécuter dans le mauvais environnement (application Windows exécutée sous DOS, par exemple).

Borland C++ ajoute un fichier de remplacement temporaire intégré au début de toute application Windows, sauf si un autre fichier de remplacement a été indiqué avec l'instruction STUB. En principe vous ne devriez pas utiliser l'instruction STUB pour intégrer WINSTUB.EXE, dans la mesure où l'éditeur de liens le fait automatiquement. La syntaxe est la suivante :

```
STUB "NomFichier"
```

NomFichier est le nom du fichier DOS exécutable qui doit être ajouté au module. Le nom doit avoir la forme d'un nom de fichier DOS.

Si le fichier désigné par *NomFichier* n'est pas dans le répertoire en cours, TLINK le recherche dans les répertoires indiqués par la variable d'environnement PATH.

**Valeurs par défaut
des fichiers de
définition de
modules**

Le fichier de définition de module n'est pas absolument indispensable pour produire un fichier Windows exécutable en Borland C++.

Si aucun fichier de définition de module n'est indiqué, les valeurs utilisées par défaut sont les suivantes :

```
CODE          PRELOAD MOVEABLE DISCARDABLE
DATA          PRELOAD MOVEABLE MULTIPLE (for applications)
              PRELOAD MOVEABLE SINGLE (for DLLs)
HEAPSIZE      4096
STACKSIZE     5120 (1048576 for TLINK32)
```

A la place de l'instruction EXETYPE, l'éditeur de liens de Borland C++ peut découvrir le type de fichier exécutable que vous voulez produire en vérifiant les valeurs définies dans l'EDI ou les options de la ligne de commande.

Vous pouvez inclure une bibliothèque d'importation à la place de la section IMPORTS du fichier de définition de modules.

Vous pouvez utiliser le mot réservé **_export** pour définir les fonctions d'exportation de votre code source C et C++, si vous voulez vous passer d'une section EXPORTS. Notez cependant que si vous utilisez **_export** pour exporter une fonction, cette fonction sera exportée par son nom et non par son numéro ordinal (ce dernier étant généralement plus efficace).

Si vous voulez modifier les valeurs par défaut de différents attributs, vous aurez besoin d'un fichier de définition de modules.

Les outils de compilation de ressources

Ce chapitre décrit les outils de compilation de ressources de Borland.

- BRCC.EXE et BRCC32.EXE sont les compilateurs de ressources de Borland. Ils compilent les fichiers .RC et génèrent des fichiers binaires .RES.
- RLINK.EXE et RLINK32.DLL (par le biais de TLINK32.EXE) sont les éditeurs de liens de ressources qui traduisent les fichiers .RES en fichiers .EXE et marquent ces derniers comme exécutable Windows.
- BRC.EXE et BRC32.EXE sont les shells par lesquels BRCC, BRCC32, RLINK et RLINK32 peuvent être lancés tous ensemble.

Sauf indication du contraire dans les pages qui suivent, tous les outils de compilation de ressources 32 bits fonctionnent comme leur équivalent 16 bits.

Les applications Windows offrent une interface standard. Les composantes de cette interface, les ressources, comprennent :

- Les menus
- Les boîtes de dialogue
- Les pointeurs
- Les icônes
- Les graphiques Bitmap
- Les chaînes de caractères
- Les raccourcis-clavier
- Les polices

Les ressources sont définies séparément de votre code, puis lui sont associées lors de l'édition des liens. L'application ne charge les ressources en mémoire que lorsqu'elle en a besoin, ce qui minimise la consommation de mémoire vive.

Les scripts de ressources (les fichiers .RC) sont des fichiers texte qui décrivent les ressources qu'utilise une application. Les outils BRCC et RC se servent de ces fichiers pour compiler les ressources en un fichier de ressources binaires (.RES). RLINK associe ensuite les fichiers .RES, qui contiennent vos ressources, à votre exécutable : cette phase s'appelle l'édition de liens des ressources ("resource linking").

BRCC : le compilateur de ressources

BRCC est la version en ligne de commande du compilateur de ressources de Resource Workshop. Il traduit les fichiers .RC en fichiers objet de ressources (.RES). La syntaxe de cet outil est la suivante :

```
BRCC [options] <fichier>.RC
```

Le tableau 10.1 décrit les options de BRCC. Notez que celles-ci peuvent être entrées en minuscules comme en majuscules.

Tableau 10.1
BRCC (le
compilateur de
ressources de
Borland)

Option	Description
@Fichier de recherche	Prend ce fichier en entrée.
-d <nom>[=<chaîne>]	Définit un symbole de pré-processeur
-fo <fichier>	Renomme le fichier de sortie .RES (par défaut, il s'agit du nom du fichier .RC d'entrée).
-i <chemin>	Ajoute un ou plusieurs répertoires (séparés par un point-virgule) au chemin include.
-r	Sans effet. Ne sert qu'à assurer la compatibilité avec d'autres compilateurs.
-v	Affiche des messages ("verbeux").
-x	Supprime le chemin include actuel.
-? ou -h	Affiche l'aide.
-30	Crée des fichiers .RES pour Windows 3.0.
-31	Crée des fichiers .RES pour Windows 3.1.
-w32	Crée des fichiers .RES pour Win32.

Comme le compilateur de RW, BRCC prédéfinit des constantes Windows liées aux ressources telles que WS_VISIBLE ET BS_PUSHBUTTON, ainsi que deux symboles spéciaux liés au compilateur, RC_INVOKED et WORKSHOP_INVOKED. Ces symboles peuvent être utilisés dans le source en plus des instructions conditionnelles de pré-processeur pour contrôler la compilation. Le code ci-dessous, par exemple, accélère grandement la compilation :

```
# ifndef WORKSHOP_INVOKED
# include "windows.h"
# endif
```

L'exemple ci-dessous ajoute deux répertoires au chemin include et génère un fichier .RES au même nom que le fichier .RC d'entrée :

```
brcc -i <répertoire1>;<répertoire2> <fichier>.RC
```

L'exemple ci-dessous génère un fichier .RES avec un nom différent du fichier .RC d'entrée :

```
brcc -fo <fichier>.RES <fichier>.RC
```

RLINK : l'éditeur de liens pour ressources

RLINK combine un fichier .RES et un fichier .EXE pour générer un exécutable Windows. RLINK accepte plusieurs fichiers .RES mais un seul exécutable en entrée. RLINK édite les liens des ressources en fixant les stringtables (tables de chaînes) et les messagetables (tables de messages) et en liant ces ressources à l'exécutable.

La syntaxe de cet outil est la suivante :

```
RLINK [options] <fichier>.RES <fichier>.EXE
```

Tableau 10.2
Les options de
RLINK

Option	Description
@<Fichier de recherche>	Prend ce fichier en entrée.
-d	Supprime les ressources du fichier .EXE (aucun fichier .RES n'est indiqué).
-fe <fichier>	Renomme le fichier de sortie .EXE.
-fi <fichier>	Renomme le fichier d'entrée .RES.
-k	Inhibe la réorganisation des segments pour un chargement rapide.

Tableau 10.2
Les options de
RLINK
(suite)

Option	Description
-v	Affiche des messages.
-vx	Liste les ressources mais ne les lie pas à l'EXE.
-? ou -h	Affiche l'aide.
-30	Crée des fichiers .RES pour Windows 3.0.
-31	Crée des fichiers .RES pour Windows 3.1.
-w32	Crée des fichiers .RES pour Win32.

L'exemple ci-dessous lie les ressources du fichier .RES et génère un fichier .EXE :

```
rlink <fichier>.RES <fichier>.EXE
```

L'exemple ci-dessous lie les ressources de deux fichiers .RES et génère un fichier .EXE :

```
rlink -fi <fichier>.RES <fichier>.RES <fichier>.EXE
```

L'exemple ci-dessous combine le code du programme .EXE en entrée avec les ressources du fichier d'entrée .RES et crée un fichier .EXE sous un nom différent :

```
rlink -fe<fichier>.EXE <fichier>.RES <fichier>.EXE
```

L'exemple ci-dessous prend en entrée un fichier de commande .RLK, puis lie les ressources de trois fichiers .RES et génère un fichier .EXE :

```
rlink <@fichier>.RLK
```

Le fichier .RLK d'entrée contient les lignes suivantes :

```
-fi <fichier>.RES  
-fi <fichier>.RES  
<fichier>.RES  
<fichier>.EXE
```

BRC : le shell de ressources

BRC (Borland Resource Compiler) est un shell de compilateur de ressources. Il lance BRCC ou RLINK (ou les deux) en fonction de la syntaxe de la ligne de commande.

La syntaxe de cet outil est la suivante :

```
BRC [options] <fichier>.RC [<fichier>.EXE]
```

Les options utilisables sont les suivantes :

Tableau 10.3
Les options de BRC

Option	Description
-d <nom>=chaîne	Définit un symbole testable par la directive #IFDEF.
-fe <fichier>	Renomme le fichier .EXE.
-fi <fichier>	Indique d'autres fichiers .RES.
-fo <fichier>	Renomme le fichier .RES.
-ihemin	Ajoute un ou plusieurs répertoires (séparés par un point-virgule) au chemin include.
-khemin	Ne pas réorganiser les segments pour un chargement rapide.
-r	Crée un fichier .RES sans l'ajouter au fichier .EXE.
-v	Affiche des messages.
-x	Force le compilateur à Ignorer la variable d'environnement INCLUDE lors de la recherche des fichiers include ou de ressources.
-31	Crée des fichiers .RES pour Windows 3.1.
-w32	Crée des fichiers .RES pour Win32.

La syntaxe de BRC peut varier en fonction du type de tâche à effectuer.

- L'exemple ci-dessous compile le fichier .RC, crée un fichier .RES et l'ajoute à l'EXE :

```
brc <fichier>.RC [<fichier>.EXE]
```

BRC recherche un fichier .EXE de même nom que le fichier .RC. Ne spécifiez le nom de l'EXE que s'il diffère de celui du fichier .RC.

- L'exemple ci-dessous crée un fichier .RES mais ne génère pas le fichier .EXE, même si vous spécifiez son nom dans la ligne de commande :

```
brc -r <fichier>.RC
```

- L'exemple ci-dessous ajoute un fichier .RES existant à un fichier exécutable. Ne spécifiez le nom de l'EXE que s'il diffère de celui du fichier .RC.

```
brc <fichier>.RES [<fichier>.EXE]
```

Les bibliothèques

Ce chapitre décrit les outils qui vous permettent de manipuler les fichiers bibliothèques à partir de l'EDI ou de la ligne de commande.

- IMPLIB crée les bibliothèques d'import et IMPDEF crée les fichiers de définition de module (fichiers .DEF). Ces deux types de fichiers fournissent à l'éditeur de liens des informations sur les fonctions importées depuis les DLL.
- TLIB est un utilitaire qui gère les bibliothèques de fichiers .OBJ (module objet). Une bibliothèque est un outil pratique pour gérer plusieurs modules objet comme s'il s'agissait d'un seul et unique élément.

IMPLIB : le bibliothécaire d'import

IMPLIB crée des bibliothèques d'import à partir de fichiers .DLL, de fichiers de définition de module, ou des deux. L'EDI utilise IMPLIB comme traducteur pour une cible DLL (voir le chapitre 2 pour de plus amples informations sur le gestionnaire de projets et les cibles). Lorsque vous ajoutez une DLL à une cible, le gestionnaire de projets compile et lie les fichiers dépendant de la DLL pour créer le fichier .DLL, puis lance IMPLIB pour créer un fichier .DLL. Il est possible de lancer IMPLIB depuis l'EDI (voir le chapitre 6).

Les bibliothèques d'import contiennent des enregistrements. Chacun contient le nom d'une DLL et indique à quel emplacement, dans cette DLL, se trouvent les fonctions à importer. Ces enregistrements sont liés à l'application par TLINK ou l'éditeur de liens de l'EDI, et fournissent à Windows les informations requises pour résoudre les appels aux fonctions DLL. Une bibliothèque d'import peut être substituée à une partie (ou à la totalité) de la section IMPORTS d'un fichier de définition de module.

Si vous avez créé une application Windows, vous avez déjà utilisé la bibliothèque d'import `IMPORT.LIB` : il s'agit de la bibliothèque d'import pour les DLL Windows standards. `IMPORT.LIB` est liée automatiquement lorsque vous créez votre application dans l'EDI et lorsque vous utilisez `BCC` pour éditer les liens. Vous ne devez spécifier que vous souhaitez éditer les liens avec `IMPORT.LIB` que si vous utilisez `TLINK` séparément.

Voir dans les pages qui suivent les indications sur la personnalisation d'une bibliothèque d'import à l'aide d'`IMPDEF` et d'`IMPLIB`

Une bibliothèque d'import liste certaines (ou toutes) fonctions exportées pour une ou plusieurs DLL. `IMPLIB` crée une bibliothèque d'import à partir de DLL, à partir de fichiers de définition de module, ou à partir des deux.

Pour créer une bibliothèque d'import pour une DLL, tapez :

```
IMPLIB Options Bibliothèque [fichiers de définition...] DLL...
```

Une DLL peut aussi avoir l'extension `.EXE` ou `.DRV`

où `Options` est une suite facultative d'options `IMPLIB` (voir le tableau 11.1), `Bibliothèque` est le nom de la nouvelle bibliothèque d'import, `fichiers de définition` est une liste de fichiers de définition de module existants pour une ou plusieurs DLL, et `DLL` est une liste de DLL existantes. Vous devez spécifier au moins une DLL et un fichier de définition de module.

Tableau 11.1
Options `IMPLIB`

Les options doivent être en minuscules et précédées d'un tiret ou d'une barre oblique ("/").

Option	Description
-c	Fait la différence entre majuscules et minuscules dans les noms de symboles (<code>MASYM</code> et <code>MaSym</code> seraient ainsi considérés comme différents).
-i	<code>IMPLIB</code> doit ignorer <code>WEP</code> , la Windows Exit Procedure requise pour terminer une DLL. Utilisez cette option si vous spécifiez plusieurs DLL dans la ligne de commande.
-w	Supprime les messages d'avertissement.

IMPDEF : le gestionnaire de fichiers de définition de module

Les bibliothèques d'import permettent d'accéder aux fonctions d'une DLL Windows.

Voir la page précédente pour de plus amples informations.

IMPDEF prend en entrée une DLL et génère un fichier de définition de module contenant une section export définissant les fonctions exportées par la DLL. La syntaxe est la suivante :

```
IMPDEF Cible.DEF Source.DLL
```

Ceci crée le fichier de définition de module *Cible.DEF* à partir de *Source.DLL*. Le fichier de définition de module résultant pourrait avoir le contenu suivant :

```
LIBRARY      Fichier
DESCRIPTION  'Description'
EXPORTS
             FonctionExport                @ordinal
             .
             .
             FonctionExport                @ordinal
```

où *Fichier* est le nom du fichier racine de la DLL, *Description* est la valeur de l'instruction du même nom si la DLL a été liée avec un fichier de définition de module contenant cette instruction, *FonctionExport* désigne une fonction exportée, et *@ordinal* est la valeur entière ordinaire de cette fonction.

Les classes d'une DLL

IMPDEF est utile pour une DLL contenant des classes C++. Si vous utilisez le mot clé **_export** lorsque vous définissez une classe, toutes les fonctions membre non-inline et tous les membres de données statiques de cette classe sont exportés. Il est plus simple de laisser IMPDEF créer pour vous le fichier de définition de module car il liste toutes les fonctions exportées et inclut automatiquement les fonctions membres et les données membres statiques.

Les noms de fonctions étant non significatifs, il serait fastidieux de les lister toutes dans la section EXPORTS d'un fichier de définition de module pour créer une bibliothèque d'import à partir du fichier de définition de module. Si vous utilisez IMPDEF pour créer le fichier de définition de module, il inclut la valeur ordinale de chaque fonction exportée. Si le nom d'une fonction est non significatif, IMPDEF inclut aussi son nom original sous forme de commentaire placé après l'entrée de cette fonction. En voici un exemple :

```
LIBRARY      Fichier
DESCRIPTION  'Description'
EXPORTS
              NomFonctionNonSignif @ordinal ; NomOriginal
              .
              .
              .
              NomFonctionNonSignif @ordinal ; NomOriginal
```

où *Fichier* est le nom du fichier racine de la DLL, *Description* est la valeur de l'instruction du même nom si la DLL a été liée avec un fichier de définition de module contenant cette instruction, *NomFonctionNonSignif* désigne un nom de fonction non significatif, *@ordinal* est la valeur entière ordinale de cette fonction, et *NomOriginal* est le nom original de la fonction.

Les fonctions d'une DLL

IMPDEF crée un fichier source éditable listant les fonctions exportées dans la DLL. Vous pouvez éditer ce fichier .DEF pour qu'il ne contienne que les fonctions que vous désirez passer à une application, puis lancer IMPLIB sur le fichier modifié. Vous obtiendrez une bibliothèque d'import contenant des informations d'import pour un sous-ensemble spécifique des fonctions d'export d'une DLL.

Supposons que vous distribuiez une DLL contenant des fonctions utilisables par plusieurs applications ; chaque fonction d'export de cette DLL est définie par **_export**. Si toutes les applications utilisaient toutes ces fonctions, vous pourriez générer, à l'aide de IMPLIB, une seule bibliothèque d'import pour la DLL, que vous livreriez avec la DLL ; cette DLL fournirait des informations d'import pour toutes les fonctions d'export de la DLL. La bibliothèque d'import pourrait être liée à n'importe quelle application, ce qui supprimerait le besoin, pour l'application cible, de lister toutes les fonctions de DLL qu'elle utilise dans la section IMPORTS de son fichier de définition de module.

Supposons maintenant que vous ne souhaitiez passer qu'une partie des exports de la DLL à une application. L'idéal serait d'avoir une bibliothèque d'import personnalisée liée à cette application - qui ne passerait que les informations d'import concernant les fonctions utilisées par application.

Pour obtenir ce résultat, lancez IMPDEF sur la DLL liée et compilée. IMPDEF génère un fichier de définition de module contenant une section EXPORT listant toutes les fonctions d'export. Supprimez alors de ce fichier les fonctions que vous ne voulez pas voir dans la bibliothèque d'import personnalisée, puis lancez IMPLIB sur le fichier modifié. Vous obtenez ainsi une bibliothèque d'import ne contenant que les informations d'import concernant les fonctions d'export listées dans la section EXPORTS du fichier de définition de module.

TLIB : le Turbo bibliothécaire

Lorsqu'il modifie une bibliothèque existante, TLIB crée une sauvegarde avec l'extension .BAK.

Les bibliothèques de Borland C++ ont été générées avec TLIB. Vous pouvez utiliser cet outil pour créer ou modifier vos propres bibliothèques ou celles de Borland C++. Vous pouvez aussi utiliser TLIB pour :

- Créer une bibliothèque à partir d'un ensemble de modules objet.
- Ajouter des modules objet ou d'autres bibliothèques à une bibliothèque existante.
- Supprimer des modules objet d'une bibliothèque.
- Remplacer des modules objet dans une bibliothèque.
- Extraire des modules objet d'une bibliothèque.
- Lister le contenu d'une bibliothèque.

Voir plus loin la section traitant de l'option /E pour de plus amples informations

TLIB peut aussi créer (et inclure dans le fichier bibliothèque) un dictionnaire étendu utilisable pour accélérer le processus d'édition des liens.

Bien que TLIB ne soit pas indispensable pour créer des exécutables avec Borland C++, il permet de gagner du temps dans les gros projets de développement.

Pourquoi utiliser des bibliothèque de modules objet ?

Quand vous programmez en C ou en C++, vous vous créez un ensemble de fonctions et de classes réutilisables. Grâce à la modularité de ces langages, vous pouvez éclater ces fonctions en plusieurs fichiers sources compilés. Ainsi, une application peut n'utiliser qu'un sous-ensemble de ces fonctions.

Une bibliothèque de modules objet gère un ensemble de fonctions et de classes. Lorsque vous linkez votre application avec une bibliothèque, l'éditeur de liens analyse la bibliothèque et ne sélectionne que les modules requis pour application.

Ligne de commande de TLIB

La syntaxe de la ligne de commande de TLIB est la suivant (les paramètres entre crochets sont optionnels) :

```
TLIB [@F_rech] [option] bibliothèque [opérations] [, fichierListe]
```

Tapez `TLIB` puis
Entrée pour afficher les
options depuis la ligne
de commande.

Dans l'EDI, vous pouvez créer une bibliothèque comme cible dans un fichier de projet. Depuis TargetExpert, choisissez Static Library comme type de cible (voir le chapitre 2 pour de plus amples informations sur le gestionnaire de projets). TLIB est le traducteur par défaut des fichiers bibliothèque ; il utilise les options définies dans la section Librarian de la boîte de dialogue Project Options (choisissez Options | Project dans le menu principal).

Tableau 11.2 Options de TLIB

Option	Librarian (option EDI)	Description
<code>@f_rech</code>		Le chemin d'accès complet du fichier de recherche à inclure. Vous pouvez en spécifier plusieurs.
<code>bibliothèque</code>		Chemin d'accès complet de la bibliothèque à créer ou à manipuler. Ce paramètre est obligatoire. Les caractères jokers ne sont pas autorisés. Par défaut, TLIB prend .LIB comme extension si vous n'en spécifiez aucune. N'utilisez que cette extension car les compilateurs en ligne de commande BCC et BCC32, ainsi que l'EDI, requièrent cette extension pour reconnaître les fichiers bibliothèques. Remarque : Si la bibliothèque indiquée n'existe pas et si vous spécifiez l'opération ADD, la bibliothèque est créée.
<code>/C</code>	Différences entre majuscules et minuscules dans la bibliothèque.	Voir page suivante pour de plus amples informations.
<code>/E</code>	Créer un dictionnaire étendu	Voir page suivante pour de plus amples informations.
<code>/Ptaille</code>	Taille de la page de la bibliothèque.	Voir page suivante pour de plus amples informations.
<code>/O</code>	Effacer les commentaires.	Supprime les enregistrements de commentaire de la bibliothèque.
<code>opérations</code>		La liste des opérations effectuées par TLIB. Aucun ordre particulier n'est requis. Si vous ne souhaitez que voir le contenu d'une bibliothèque, n'indiquez aucune opération.

Tableau 11.2 Options de TLIB (suite)

<i>fichierListe</i>	Générer un fichier liste
---------------------	--------------------------

Le nom du fichier listant le contenu de la bibliothèque, précédé d'une virgule. Si vous omettez cette option, aucun fichier liste n'est généré. Le listage contient les noms de modules, classés par ordre alphabétique. L'extension par défaut de ce fichier est .LST. Les informations qu'il contient peuvent être envoyées à l'écran en utilisant CON comme *fichierListe*, et à l'imprimante en utilisant PRN.

Les fichiers de recherche

Lorsque vous utilisez de nombreuses opérations, ou si vous spécifiez régulièrement les mêmes opérations, nous vous recommandons de vous servir des fichiers de recherche. Il s'agit de fichiers ASCII, pouvant être créés avec l'éditeur de Borland C++, contenant des commandes TLIB. Vous pouvez ainsi générer des commandes qui ne tiendraient pas sur une seule ligne de commande. Les fichiers de recherche peuvent :

- Contenir plusieurs lignes de texte ; utilisez le caractère **&** à la fin d'une ligne suivie par une autre.
- Inclure une liste de commandes partielle. Il est possible de combiner les options de la ligne de commande avec les options d'un fichier de recherche.
- Être utilisés avec d'autres fichiers de recherche dans une ligne de commande TLIB.

Distinction des majuscules et des minuscules dans les symboles : l'option /C

N'utilisez pas l'option /C si vous avez l'intention d'utiliser une bibliothèque avec d'autres éditeurs de liens, ou si vous voulez que d'autres puissent utiliser la bibliothèque.

TLIB utilise un dictionnaire des symboles publics définis dans les modules de la bibliothèque. Si vous ajoutez un module à une bibliothèque, son symbole doit être unique. Dans le cas contraire, TLIB affiche un message d'erreur et n'ajoute pas le module en question.

Certains éditeurs de liens n'étant pas sensibles à la différence entre majuscules et minuscules, TLIB refuse les noms de symboles dont la seule différence réside dans la casse ; ainsi, *LOOKUP* et *LookUp* sont considérés comme identiques. Cependant, TLINK est capable de faire la distinction entre les deux, grâce à l'option TLIB /C.

Création d'un dictionnaire étendu : l'option /E

Pour accroître les capacités de l'éditeur de liens en ce qui concerne les liens de taille importante, vous pouvez utiliser TLIB pour créer un dictionnaire étendu et pour l'ajouter au fichier bibliothèque. Ce dictionnaire contient, dans un format compacté, des informations qui n'apparaissent pas dans le dictionnaire standard de la bibliothèque et qui permettent à TLINK de pré-traiter les fichiers bibliothèques pour que les modules non requis ne soient pas traités.

Pour créer un dictionnaire étendu pour une bibliothèque que vous modifiez, utilisez l'option **/E** au lancement de TLIB pour ajouter, effacer ou remplacer des modules dans la bibliothèque. Cette option permet aussi de créer un dictionnaire étendu pour une bibliothèque que vous ne souhaitez pas modifier. Par exemple, si vous tapez TLIB /E bibliothèque, l'éditeur de liens ajoute un dictionnaire étendu à la bibliothèque spécifiée.

Si vous utilisez **/E** pour ajouter un module bibliothèque contenant une classe C++ avec une fonction virtuelle, vous verrez le message d'erreur `Library contains COMDEF records--extended dictionary not created.`

Choix du format de la page : l'option /P

Toute bibliothèque DOS contient un dictionnaire qui apparaît à la fin du fichier .LIB, après les modules objet. Pour chaque module, le dictionnaire contient l'adresse (sur 16 bits) du module dans le fichier .LIB. Cette adresse est donnée par rapport à la taille de la page de bibliothèque (16 octets par défauts).

La taille de la page de bibliothèque détermine la taille combinée maximale de tous les modules objet dans la bibliothèque ; celle-ci ne peut excéder 65 536 pages. La valeur par défaut minimale de 16 octets autorise une bibliothèque d'environ 1 Mo. Pour créer une bibliothèque plus grande, utilisez l'option **/P** pour augmenter la taille des pages. La taille de page doit être une puissance de 2 et comprise entre 16 et 32 768.

Tous les modules de la bibliothèque doivent commencer sur une limite de page. Par exemple, si la taille de page est de 32, 16 octets en moyenne sont perdus, pour des besoins de remplissage ("padding"), par module. Si vous essayez de créer une bibliothèque trop grande pour la taille de page spécifiée, TLIB affiche un message d'erreur et vous suggère d'utiliser l'option **/P** avec la première taille de page supérieure et disponible.

Suppression des enregistrements de commentaire :
l'option /O

Utilisez cette option pour supprimer des enregistrements de commentaire (débogage ou browsing) afin de réduire la taille de vos bibliothèques.

Liste d'opérations

La liste d'opérations passe des ordres d'exécution à TLIB. Elle est composée d'une suite d'opérations placées les unes après les autres. Chacune consiste en un symbole d'une ou de deux lettres suivies d'un nom de fichier ou de module. Vous pouvez insérer des espaces entre les symboles d'action ou entre les noms de fichiers ou de modules, mais pas entre les caractères d'un symbole ou d'un nom de fichier.

La seule limite au nombre d'opérations est la limite de 127 caractères dans une ligne de commande, limite imposée par DOS. L'ordre dans lequel TLIB effectue les opérations demandées est le suivant :

1. Les extractions en premier.
2. Les suppressions en second.
3. Les ajouts en dernier.

Pour remplacer un module, supprimez-le puis ajoutez le nouveau module.

Les caractères jokers ne sont pas autorisés dans les noms de fichiers ou de modules.

TLIB trouve le nom d'un module en enlevant les données concernant l'unité, le chemin et l'extension du fichier spécifié. TLIB fonctionne avec des valeurs par défauts raisonnables : ainsi, pour ajouter un module ayant l'extension .OBJ et situé dans le répertoire courant, vous ne devez spécifier que le nom du module ; le chemin et l'extension .OBJ sont superflus.

TLIB reconnaît trois symboles d'action (-, + et *) que vous pouvez utiliser séparément ou en paire pour obtenir un total de cinq opérations. L'ordre des caractères n'est pas important pour les opérations qui utilisent une paire de caractères. Les symboles et leur descriptions sont répertoriés dans le tableau 11.3

Tableau 11.3
Symboles d'action
de TLIB

Pour créer une bibliothèque, ajoutez des modules à une bibliothèque qui n'a encore jamais été créée.

Symbole	Nom	Description
+	Ajouter	TLIB ajoute le nom à la bibliothèque. Si aucune extension n'est spécifiée, TLIB prend .OBJ par défaut. Si le fichier est lui-même une bibliothèque (extension .LIB), l'opération ajoute tous ses modules à la bibliothèque cible. Si l'un des modules à ajouter existe déjà, TLIB vous le signale et ne l'ajoute pas.
-	Supprimer	TLIB supprime de la bibliothèque les modules spécifiés. S'ils ne s'y trouvent pas, il vous le signale. Cette opération ne requiert qu'un nom de module, bien que le chemin, l'unité et l'extension soient autorisés.
*	Extraire	TLIB crée le fichier spécifié en extrayant le module indiqué et en le copiant dans le nouveau fichier. Si le module n'existe pas, TLIB affiche un message et ne crée pas le fichier. Si le fichier à créer existe déjà, il est écrasé.
-* ou *-	Extraire et supprimer	TLIB copie le module spécifié dans le fichier indiqué, puis le supprime de la bibliothèque source
-+ ou +-	Remplacer	TLIB remplace le module spécifié dans le fichier indiqué.

Exemples

Ces exemples montrent l'étendue des possibilités de TLIB.

- Pour créer la bibliothèque BIBLIO.LIB avec les modules X.OBJ, Y.OBJ et Z.OBJ, tapez `TLIB BIBLIOTHEQUE +X +Y +Z`
- Pour créer la bibliothèque BIBLIO.LIB et obtenir également une liste dans MALISTE.LST, tapez `TLIB BIBLIO +X +Y +Z, MALISTE.LST`
- Pour remplacer le module X.OBJ par une version plus récente, ajouter Y.OBJ et supprimer Z.OBJ de BIBLIO.LIB, tapez `TLIB BIBLIO -+X +Y -Z`
- Pour créer la bibliothèque ALPHA.LIB avec les modules A.OBJ, B.OBJ... G.OBJ à l'aide d'un fichier de recherche :
 - Créez un fichier texte (appelé ici ALPHA.RSP) contenant les instructions suivantes :


```
+A.OBJ +B.OBJ + C.OBJ &
+D.OBJ +E.OBJ +F.OBJ &
+G.OBJ +
```
 - Entrez ensuite la commande suivante :


```
TLIB ALPHA @ALPHA.RSP, ALPHA.LST
```


Utilisation de MAKE

MAKE.EXE est un gestionnaire de projet en ligne de commande qui vous permet de compiler uniquement les fichiers d'un projet qui ont été modifiés depuis la dernière compilation (MAKER est une version en mode réel de MAKE). Si vous utilisez l'EDI, nous vous conseillons d'utiliser le gestionnaire de projet de l'EDI (voir le chapitre 2).

Ce chapitre couvre les sujets suivants :

- Généralités sur MAKE
- Contenu d'un fichier MAKE
- Utilisation des règles explicites et implicites
- Les macros de MAKE
- Les directives de MAKE

Généralités sur MAKE

MAKE utilise les règles d'un fichier texte (MAKEFILE ou MAKEFILE.MAK par défaut) pour déterminer quels fichiers doivent être créés et comment les créer. Vous pouvez ainsi, par exemple, compiler un fichier .EXE si les fichiers .CPP contenant le source correspondant sont plus anciens que le fichier .EXE résultant. MAKE est très utile lorsque vous générez un programme à partir de plusieurs fichiers, car il ne recompile que ceux que vous avez modifiés.

Deux types de règles (explicites et implicites) indiquent à MAKE quels fichiers dépendent les uns des autres. MAKE compare alors les dates des fichiers d'une règle et décide de l'exécution d'une commande (les commandes portent en général sur la compilation ou l'édition des liens, mais il peut s'agir de pratiquement toutes les commandes du système d'exploitation).

MAKE accepte les caractères jokers ? et *

Pour obtenir de l'aide, tapez `MAKE -?` ou `MAKE -h`

Pour placer les instructions de MAKE ailleurs que dans un fichier MAKE, voyez la section "Options de MAKE".

La syntaxe générique de MAKE est la suivante :

```
MAKE [options...] [cibles [s]]
```

où `options` sont les options de MAKE contrôlant son fonctionnement, et `cibles` sont les noms des fichiers du fichier MAKE à créer. Les options sont séparées de MAKE par un seul espace. Les options et les cibles sont séparées entre elles par un espace.

Si vous tapez MAKE sur la ligne de commande, MAKE effectue les tâches par défaut suivantes :

1. MAKE recherche dans le répertoire courant le fichier BUILTINS.MAK, qui contient les règles que MAKE respecte si vous n'utilisez pas l'option **-r**. S'il ne le trouve pas, il le cherche dans le répertoire où MAKE.EXE est placé. Après avoir chargé BUILTINS.MAK, MAKE recherche un fichier appelé MAKEFILE ou MAKEFILE.MAK. S'il n'en trouve aucun, il affiche un message d'erreur.
2. Quand MAKE trouve le fichier MAKE, il tente de ne créer que le premier fichier cible dans ce fichier (mais la première cible peut forcer la création d'autres cibles). MAKE vérifie la date et l'heure des fichiers dépendant de la première cible. S'ils sont plus récents que le fichier cible, MAKE exécute les commandes cibles, ce qui met la cible à jour. Voir la section "Fichiers MAKE" pour de plus amples informations.
3. Si un fichier dépendant de la première cible est présent sous forme de cible dans le fichier MAKE, MAKE vérifie ses dépendances et le crée avant de créer la première cible. Cette réaction en chaîne s'appelle "dépendance liée".
4. Si MAKE ne peut créer les fichiers demandés, il efface le fichier cible qu'il a commencé à créer. Pour éviter cela, utilisez la directive **-precious** (voir plus loin dans le chapitre).

Pour arrêter MAKE, appuyez sur *Ctrl+C* ou sur *Ctrl+Pause*.

BUILTINS.MAK

BUILTINS.MAK contient les règles et macros standards utilisées par MAKE pour avant qu'il ne lise le fichier MAKE (pour que BUILTINS.MAK soit ignoré, utilisez l'option **-r**). Servez-vous du fichier BUILTINS.MAK pour indiquer quelles instructions ou macros doivent être exécutées à chaque lancement de MAKE. Voici le contenu par défaut de ce fichier :

```
#
# Borland C++ - (C) Copyright 1992 by Borland International
#
CC= BCC
AS = TASM
RC = RC
.asm.obj:
    $(AS) $(AFLAGS) $&.asm
.c.exe:
    $(CC) $(CFLAGS) $&.c
.c.obj:
    $(CC) $(AFLAGS) /c $&.c
.cpp.obj:
    $(CC) $(CPPFLAGS) /c $&.cpp
.rc.res:
    $(RC) $(RFLAGS) /r $&
.SUFFIXES: .exe .obj .asm .c .res .rc
```

Utilisation de TOUCH.EXE

Il est parfois nécessaire de recompiler un fichier cible même si vous ne l'avez pas modifié. L'un des moyens pour obtenir ce résultat est d'utiliser l'utilitaire TOUCH. Cet outil change la date et l'heure des fichiers spécifiés (leur donnant par défaut la date et l'heure actuelles), ce qui les rend plus récents que les fichiers dont ils dépendent.

Vous pouvez obliger MAKE à reconstruire un fichier cible en appliquant TOUCH à l'un des fichiers dont dépend la cible. La syntaxe est la suivante :

```
TOUCH fichier [fichier...]
```

Les jokers * et ?
sont autorisés

TOUCH met à jour la date et l'heure des fichiers indiqués.

Important !

Avant d'utiliser TOUCH, vérifiez les valeurs indiquées par l'horloge interne de votre ordinateur. En cas d'informations erronées, TOUCH et MAKE ne fonctionneront pas correctement.

Options de MAKE

MAKE accepte plusieurs options en ligne de commande. Celles-ci, sensibles à la différence entre majuscules et minuscules, doivent être précédées d'un "-" ou d'un "/". Ainsi, pour utiliser le fichier PROJETA.MAK comme fichier MAKE, tapez : `MAKE -f PROJETA.MAK` (l'espace après **-f** est facultatif). La plupart des options en ligne de commande ont une directive correspondante dans le fichier MAKE (voir les pages qui suivent pour de plus amples informations sur les directives).

Tableau 12.1 : Options MAKE

Option	Description
-h ou -?	Affiche les options de MAKE (les options par défaut sont suivies d'un +).
-B	Crée toutes les cibles, quelle que soit la date des fichiers.
-Dmacro	Définit macro comme un seul caractère, ce qui force toute expression lifdef macro dans le fichier MAKE à retourner VRAI.
[-D]macro=[chaîne]	Définit la <i>chaîne</i> en tant que <i>macro</i> . Si cette chaîne contient des espaces ou des tabulations, entourez-la d'apostrophes ('). Le -D est facultatif.
-Irépertoire	Recherche les fichiers include dans le répertoire spécifié.
-K	Garde les fichiers temporaires créés par MAKE. Voir aussi KEEP dans les pages qui suivent.
-N	Exécute MAKE comme NMAKE de Microsoft (voir les pages qui suivent pour de plus amples informations).
-Umacro	Annule les définitions précédentes de macro.
-W	Ecrit les options non-chaînes courantes dans MAKE.EXE pour en faire des options par défaut.
-ffichier	Utilise fichier ou fichier.MAK au lieu de MAKEFILE (l'espace après -f est facultatif).
-a	Vérifie les dépendances des fichiers include et les fichiers include imbriqués associés aux fichiers .OBJ et met à jour le fichier .OBJ si le fichier .H a été modifié. Voir aussi -c .
-c	Place les informations d'auto-dépendance en mémoire cache pour de meilleures performances. A utiliser avec -a. Ne pas utiliser si MAKE modifie les fichiers include (avec TOUCH depuis un fichier MAKE, en créant un en-tête ou des fichiers include, par exemple).
-drépertoire	A utiliser avec -S pour spécifier l'unité et le répertoire utilisés par MAKE lorsqu'il quitte la mémoire. Cette option est sans effet avec MAKER.EXE.
-e	Ignore une macro si son nom est le même qu'une variable d'environnement (MAKE utilise alors cette dernière).
-i	Ignore l'état de sortie de tous les programmes lancés depuis MAKE et continue le processus de création.
-m	Affiche la date et l'heure de chaque fichier traité.
-n	Affiche les commandes mais ne les exécute pas (utile pour le débogage).

Tableau 12.1 : Options MAKE (suite)

Option	Description
-p	Affiche les définitions de macros et les règles implicites avant d'exécuter le fichier MAKE.
-q	Retourne 0 si la cible est à jour (à utiliser dans un fichier batch).
-r	Ignore les règles définies dans BUILTINS.MAK.
-s	Supprime l'affichage des commandes à l'écran.
-S	Retire MAKE de la mémoire lors de l'exécution des commandes, pour de meilleures performances. Cette option est sans effet sur MAKER.

Définir des options par défaut

L'option **-W** permet de définir certaines options de MAKE par défaut pour qu'elles soient activées à chaque lancement de MAKE. La syntaxe est la suivante :

```
MAKE -option[-] [-option] [-]. . . -W
```

Vous pouvez par exemple taper `MAKE -m -w` pour que la date des fichiers traités s'affiche toujours. Pour annuler cette commande, tapez `MAKE -m -w`. Lorsque MAKE vous demande s'il doit enregistrer les modifications dans MAKE.EXE, tapez Y (oui).

Attention ! L'option **-W** ne fonctionne pas si le programme SHARE de DOS est lancé : le message `Fatal: unable to open file MAKE.EXE` s'affiche. L'option **-W** ne fonctionne pas avec les options de MAKE suivantes :

- **-Dmacro**
- **-Dmacro=[chaîne]**
- **-drépertoire**
- **-Usymbole**
- **-ffichier**
- **-? ou -h**
- **-lrépertoire**

Compatibilité avec NMAKE de Microsoft

Utilisez l'option **-N** si vous souhaitez traiter des fichiers MAKE créés pour NMAKE de Microsoft. Les changements occasionnés sont les suivants :

- MAKE interprète l'opérateur **<** comme l'opérateur **&&** : les fichiers temporaires sont utilisés comme fichiers de recherche, puis effacés. Pour conserver un fichier, utilisez l'option **-K** dans la ligne de commande ou **KEEP** dans le fichier MAKE.

En général, MAKE efface ses fichiers temporaires.

```
<FICHTEMP.TXT!  
texte  
.  
.  
.  
!KEEP
```

Si vous ne voulez pas conserver un fichier temporaire, tapez NOKEEP ou uniquement le nom du fichier en question. Si vous utilisez NOKEEP avec un fichier temporaire, puis l'option **-K** avec MAKE, le fichier temporaire est effacé.

- La macro **\$d** est traitée différemment. Utilisez plutôt **lifdef** ou **lifndef**.
- Les macros qui retournent un chemin d'accès ne retourneront pas le dernier "\". Par exemple, si **\$()** retourne normalement **C:\CPP**, elle ne retournera, avec **-N**, que **C:\CPP**.
- A moins qu'il n'existe la directive **.suffixes** correspondante, MAKE lit les règles en commençant par le bas du fichier MAKE.
- La macro **\$*** est toujours convertie en nom de cible au lieu de celui de la dépendance dans une règle implicite.

Utilisation des fichiers MAKE

Un fichier MAKE est un fichier ASCII d'instructions pour MAKE.EXE. MAKE suppose que votre fichier MAKE s'appelle MAKEFILE ou MAKEFILE.MAK, sauf si vous utilisez l'option **-f** (voir plus haut).

MAKE crée les cibles spécifiées dans la ligne de commande ou ne crée que la première cible trouvée dans le fichier MAKE (voir la section Cibles symboliques pour de plus amples informations). Un fichier MAKE peut contenir :

- Des commentaires.
- Des règles explicites.
- Des règles implicites.
- Des macros.
- Des directives.

Cibles symboliques

Une cible symbolique force MAKE à créer des cibles multiples dans un fichier MAKE (il n'est pas nécessaire d'utiliser les dépendances liées). La ligne de dépendance liste toutes les cibles que vous désirez créer. Vous n'entrez aucune commande pour une cible symbolique.

Dans le fichier MAKE ci-dessous, la cible symbolique ToutesCibles crée à la fois FICHER1.EXE et FICHER2.EXE :

```
ToutesCibles: FICHER1.EXE FICHER2.EXE # Cette cible n'a
                                                # pas de commande

FICHER1.EXE: FICHER1.OBJ
             BCC FICHER1.OBJ
FICHER2.EXE: FICHER2.OBJ
             BCC FICHER2.OBJ
```

Règles pour cibles symboliques

Respectez les règles suivantes lorsque vous utilisez les cibles symboliques :

- Les cibles symboliques ne requièrent pas de ligne de commande.
- Donnez à vos cibles symboliques un nom unique, différent des noms de fichiers dans le répertoire courant.
- Nommez vos cibles symboliques en respectant les conventions de noms de fichiers de votre système d'exploitation.

Règles explicites et implicites

Les règles explicites et implicites qui gèrent MAKE sont généralement définies comme suit :

- Les règles explicites concernent des fichiers spécifiques.
- Les règles implicites sont suivies par MAKE lorsqu'il ne trouve pas de règle explicite.

Le format d'une règle est le suivant :

```
Ligne de dépendance
  Commandes

.
.
.
```

La ligne de dépendance est différente pour les règles explicites et implicites, mais les commandes sont les mêmes (pour de plus amples informations sur les dépendances liées, voir les pages précédentes).

MAKE supporte des règles multiples pour une seule cible. Vous pouvez ajouter des fichiers dépendants après la première règle explicite, mais un seul doit contenir une ligne de commande. Par exemple :

```
Cible1: dépendance1 dép2 dép3 dép4 dép4
Cible1: dép6 dép7
      bcc -c $**
```

Syntaxe des règles explicite

Les règles explicites sont des instructions indiquant à MAKE les noms de fichiers exacts. Elles nomment une ou plusieurs cibles, suivies d'un ou de deux caractères ":". Un seul signifie qu'une règle est écrite pour la cible, et deux signifient que plusieurs règles le sont.

La syntaxe des règles explicites est la suivante :

Utilisez les accolades si vous incluez le paramètre *chemin*

```
Cible [cible...] : [{chemin}] [dépendance(s)...]
                        [commandes]
```

.
.
.

- *Cible* Nom et extension du fichier à mettre à jour. *Cible* doit être en début de ligne (les espaces et les tabulations ne sont pas autorisés). Si vous avez plusieurs cibles, séparez-les par des espaces ou par des tabulations. N'utilisez pas une cible plusieurs fois à la position de la cible d'une règle explicite dans un fichier MAKE.
- *chemin* Liste des répertoires, séparés par un point-virgule et placés entre accolades, pointant sur les fichiers dépendants.
- *dépendance* Fichier(s) dont la date et l'heure est vérifiée par MAKE. Ce paramètre doit être précédé par un espace. Si un fichier dépendant apparaît aussi dans le fichier MAKE comme cible, MAKE met à jour ou crée le fichier cible avant de l'utiliser comme dépendance pour une autre cible.

- *commandes* Toute commande du système d'exploitation. Les commandes multiples sont autorisées dans une règle. Les commandes doivent être indentées d'au moins un espace ou d'une tabulation (voir la section traitant des commandes dans les pages qui suivent).

Si la dépendance ou les commandes continuent sur la ligne suivante, utilisez la barre oblique inverse ("`\`") à la fin de la ligne après un nom de cible ou de fichier dépendant :

```
SOURCE.EXE: FILE1.OBJ\  
             FILE2.OBJ\  
             FILE3.OBJ  
             bcc file1.obj file2.obj file3.obj
```

Cibles uniques avec règles multiples

Une cible unique peut avoir plusieurs règles explicite. Vous devez utiliser le caractère ":" après le nom de la cible pour que MAKE sache qu'il va traiter plusieurs règles explicites :

```
.cpp.obj:  
    bcc -c -ncobj $  
  
.asm.obj:  
    tasm /mx $, asmobj \  
  
mabiblio.lib :: f1.obj f2.obj  
             echo Ajout des fichiers C  
  
tlib mabiblio -+cobj\f1 -+cobj\f2  
mabiblio.lib :: f3.obj f4.obj  
             echo Ajout des fichiers ASM  
             tlib mabiblio -+asmobj\f3 -+asmobj\f4
```

Syntaxe des règles implicites

Une règle implicite commence par un chemin ou par un point et sous-entend une relation de fichiers dépendant de la cible. Ses composants principaux sont des extensions de fichiers séparés par des points. La première extension appartient à la dépendance, la seconde à la cible.

Si des dépendances implicites ne sont pas à jour, MAKE exécute les commandes associées à la règle. MAKE met à jour les dépendances explicites avant les dépendances implicites.

La syntaxe des règles implicites est la suivante :

```
[{rép_source}] . ext_source [{rép_cible}] . ext_cible  
[commandes]
```

.
:
.

- *rép_source* Le répertoire des fichiers dépendants. Séparez les différents répertoires par un point-virgule.
- *ext_source* L'extension du fichier dépendant.
- *rép_cible* Le répertoire des fichiers cibles (exécutables). Séparez les différents répertoires par un point-virgule.
- *ext_cible* L'extension du fichier cible.
- **:** Marque la fin de la ligne de dépendance.
- *commandes* Toute commande du système d'exploitation. Les commandes multiples sont autorisées dans une règle. Les commandes doivent être indentées d'au moins un espace ou d'une tabulation (voir la section traitant des commandes dans les pages qui suivent).

Si deux règles implicites correspondent à une extension cible mais qu'aucune dépendance n'existe, MAKE utilise la règle implicite dont l'extension de la dépendance apparaît en premier dans la liste .SUFFIXES. Voir les pages qui suivent pour de plus amples informations sur les extensions.

**Règles explicites
utilisées avec des
commandes
implicites**

Une cible dans une règle explicite peut prendre sa ligne de commande dans une règle implicite. L'exemple ci-dessous montre une règle implicite et une règle explicite sans ligne de commande :

```
.c.obj:  
    bcc -c $ #Cette commande utilise la macro $  
            #décrite plus loin.  
  
.projet.obj: #Cette règle explicite utilise la commande  
            #bcc -c .projet.obj:
```

Voir plus loin pour de plus amples informations sur les macros par défaut

La commande de la règle implicite dit à MAKE de compiler PROJET.C (la macro `$<` remplace le nom `projet.obj` par `PROJET.C`).

Syntaxe des commandes

Les commandes peuvent être toute commande du système d'exploitation, mais elles peuvent aussi inclure des macros et directives MAKE, ainsi que des opérateurs spéciaux non reconnus par les systèmes d'exploitation (notez que vous ne pouvez pas utiliser le caractère "|" dans les commandes). Voici quelques exemples de commandes :

```
cd..

bcc -c mysource.c

COPY *.OBJ C:\PROJETA

bcc -c $(SOURCE)      # Les macros sont décrites dans les
                       # pages qui suivent.
```

La syntaxe des commandes est la suivante :

```
[préfixe...] commandes
```

Préfixes des commandes

Les commandes dans les règles implicites et explicites peuvent avoir un préfixe modifiant leur traitement par MAKE. Le tableau 12.2 liste les préfixes que vous pouvez utiliser dans les fichiers MAKE. Ceux-ci sont décrits plus en détail dans les pages qui suivent.

Tableau 12.2
Préfixes des
commandes

Option	Description
@	Ne pas afficher la commande en cours d'exécution.
- <i>valeur</i>	Arrêter le traitement des commandes du fichier MAKE si le code de sortie retourné par la commande est supérieur à <i>valeur</i> . Par défaut, MAKE arrête le traitement si le code de sortie est différent de zéro. Vous ne pouvez pas insérer d'espace entre "-" et " <i>valeur</i> ".
-	Continuer le traitement des commandes du fichier MAKE, quel que soit leur code de sortie.
&	Interpréter soit la macro <code>\$**</code> , qui représente tous les fichiers dépendants, soit la macro <code>\$?</code> , qui représente tous les fichiers dépendants modifiés après le fichier cible. Exécuter la commande une fois pour chaque fichier dépendant dans la macro interprétée.

Utilisation de @

La commande ci-dessous utilise le modificateur @ qui empêche l'affichage de la commande exécutée par MAKE :

```
diff.exe : diff.obj
          @bcc diff.obj
```

**Utilisation de
-valeur et de -**

Les modificateurs **-valeur** et **-** contrôlent le traitement de MAKE en cas d'erreur. Ils vous permettent de spécifier si MAKE doit s'arrêter si une erreur se produit ou si les erreurs dépassent une certaine valeur.

Dans l'exemple ci-dessous, MAKE poursuit son traitement si BCC ne s'exécute pas normalement :

```
cible.exe : cible.obj
cible.obj : cible.cpp
          bcc -c cible.cpp
```

Utilisation de &

Le modificateur **&** lance une commande une fois par fichier dépendant. Il est très utile pour les commandes qui n'acceptent pas de liste de fichiers comme paramètres. Par exemple,

```
copitout : fichier1.cpp fichier2.cpp
          &copy $** c:\temp
```

lance deux fois la commande COPY comme suit :

```
copy fichier1.cpp c:\temp
copy fichier2.cpp c:\temp
```

Sans le modificateur **&**, la commande COPY ne serait lancée qu'une fois.

**Opérateurs de
commandes**

Vous pouvez utiliser toute commande du système d'exploitation dans la section de commandes de MAKE. MAKE utilise les opérateurs standards (+, -, etc.), mais il compte aussi les opérateurs suivants :

Tableau 12.3
Opérateurs MAKE

Opérateur	Description
<	Prendre les entrées de commande dans le fichier spécifié plutôt que dans l'entrée standard.
>	Envoyer la sortie de la commande dans le fichier spécifié.
>>	Ajouter la sortie de la commande au fichier.

Tableau 12.3
Opérateurs MAKE
(suite)

Opérateur	Description
<<	Créer un fichier inline temporaire et l'utiliser comme entrée de la commande spécifiée.
&& <i>délimiteur</i>	Créer un fichier temporaire et insérer son nom dans le fichier MAKE. Tout caractère autre que # et \ utilisé avec < et && comme délimiteur de début et de fin d'un fichier temporaire. Tout caractère sur la même ligne et suivant immédiatement le délimiteur de début est ignoré. Le délimiteur de fin doit être placé seul sur une ligne.

Débogage avec fichiers temporaires

Les fichiers temporaires peuvent vous aider à déboguer une suite de commandes en plaçant la commande exécutée par MAKE dans le fichier temporaire. Le nom des fichiers temporels commence à MAKE0000.@@@, où le 0000 s'incrémente avec chaque fichier temporaire créé. Vous devez placer des délimiteurs après && et à la fin de ce que vous envoyez au fichier temporaire ("!" est un bon choix).

L'exemple ci-dessous montre comment les caractères && demandent à MAKE de créer un fichier avec l'entrée de TLINK :

```
program.exe: A.obj B.obj
    TLINK /c &&!
    cOs.obj $**
    program.exe
    program.map
    maths.lib cd.lib
    !
```

Le fichier de recherche créé par && contient les instructions suivantes :

```
cOs.obj A.obj B.obj
program.exe
program.map
maths.lib cd.lib
```

Les macros de MAKE

Attention à la casse des macros : MACRO1 est différent de macro1.

Une macro MAKE est une chaîne qui est utilisée dès que la macro est appelée dans un fichier MAKE. Les macros permettent de créer des fichiers-types que vous pouvez modifier au fil des projets. Par exemple, pour définir la macro NOMDELIB représentant la chaîne "mabiblio.lib", tapez `NOMDELIB = mabiblio.lib`.

Lorsque MAKE rencontre la macro `$(NOMDELIB)`, il utilise la chaîne `mabiblio.lib`.

Si MAKE rencontre une macro non définie, il cherche une variable d'environnement du système d'exploitation (définie par SET) portant le même nom et utilise sa définition comme valeur. Par exemple, si vous avez entré `$(path)` dans un fichier MAKE mais n'avez jamais défini `path`, MAKE utilisera la valeur de `PATH` dans votre AUTOEXEC.

Définition des macros

La syntaxe générale pour les macros des fichiers MAKE est la suivante :

NomdeMacro = texte_à_utiliser

- *NomdeMacro* est limité à 512 caractères. Les minuscules sont différenciées des majuscules.
- *texte_à_utiliser* est limité à 4 096 caractères, y compris les signes de ponctuation et les espaces.

Chaque macro doit être sur une ligne distincte dans le fichier MAKE, placée au début du fichier en règle générale. Si MAKE rencontre plusieurs définitions pour la même macro, c'est la dernière qui prévaut.

Vous pouvez définir une macro en utilisant l'option en ligne de commande **-D** (voir le début du chapitre). Pour en définir plusieurs, séparez-les par un espace :

```
make -D RépSource = c:\projetA
```

```
make command = "bcc -c"
```

```
make command = bcc option=-c
```

Vous trouverez ci-dessous les différences de syntaxe qui existent entre les macros entrées depuis la ligne de commande et les macros placées dans un fichier MAKE :

Tableau 12.4
Différences entre
les deux types de
macros

Syntaxe	Fichier MAKE	Ligne de commande
Espaces permis avant et après "="	Oui	Non
Espace permis avant NomdeMacro	Non	Oui

Utilisation des macros

Pour utiliser une macro dans un fichier MAKE, tapez $\$(NomDeLaMacro)$. Vous pouvez utiliser des accolades ou des parenthèses autour du nom de la macro.

MAKE étend les macros à différents moments, selon leur position dans le fichier MAKE :

- Les macros imbriquées sont étendues lorsque la macro extérieure est lancée.
- Les macros dans les règles et les directives sont étendues lorsque MAKE ouvre le fichier MAKE.
- Les macros dans les commandes sont étendues lorsque la commande est exécutée.

Substitution de chaînes dans les macros

MAKE permet de temporairement substituer des caractères dans une macro déjà définie. Par exemple, si vous avez défini la macro SOURCE ainsi :
SOURCE = f1.cpp f2.cpp f3.cpp, vous pouvez remplacer ".CPP" par ".OBJ" en tapant $\$(SOURCE:.CPP=.OBJ)$. Ceci ne redéfinit pas la macro.

Règles concernant la substitution dans les macros :

- Syntaxe :
 $\$(NomDeMacro:texte_original=texte_de_remplacement)$.
- Ne placez pas d'espaces avant ou après les deux points.
- Les caractères de *texte_original* doivent être absolument identiques, en terme de majuscules et de minuscules, à la définition de la macro.

MAKE vous permet maintenant d'utiliser les macros au sein de macros de substitution :

```
MONEXT=.C
SOURCE = f1.cpp f2.cpp f3.cpp
$(SOURCE:.cpp=$(MONEXT)) # Change f1.cpp en f1.c, etc.
```

Macros par défaut de MAKE

MAKE contient plusieurs macros par défaut que vous pouvez utiliser dans vos fichiers MAKE. Le tableau 12.5 donne leur définition et leur interprétation dans les règles implicites et explicites.

Tableau 12.5 : Macros par défaut

Macro	Interprétation implicite	Interprétation explicite	Exemple
\$*	chemin\ fichier dépendant	chemin\cible	C:\projeta\macible
\$	chemin\dépendant+ext	chemin\cible+ext	C:\projeta\macible.obj
\$(chemin des dépendants	chemin de la cible	C:\projeta
\$.	fichier dépendant+ext	cible + ext	MYSOURCE.c
\$&	fichier dépendant	cible MYSOURCE	
\$@	chemin\cible +ext	chemin\cible+ext	C:\projeta\mYsource.c
**	chemin\dépendant+ext	tous les dépendants+ext	fichier1.cpp fichier2.cpp fichier3.cpp
\$?	chemin\dépendant+ext	anciens dépendants	fichier1.cpp

Tableau 12.6
Autres macros par défaut

Macro	Interprétation	Commentaire
__MSDOS__	1	Si DOS actif.
__MAKE__	0x0370	Version de MAKE (en hexadécimal).
MAKE	make	Nom de l'exécutable MAKE.
MAKEFLAGS	options	Options entrées sur la ligne de commande.
MAKEDIR	répertoire	Répertoire de MAKE.

Modification des macros par défaut

Si les macros du tableau 12.5 ne vous donnent pas le résultat attendu, vous pouvez les modifier pour extraire la partie de la chaîne qui vous intéresse.

Voici la syntaxe à utiliser :

```
$(NomDeMacro [modificateur])
```

Le tableau 12.7 liste les modificateurs de macros et donne un exemple d'utilisation.

Tableau 12.7
Modificateurs de
noms de fichiers

Modificateur	Partie du nom de fichier traitée	Exemple	Résultat
D	Unité et répertoire	\$(<D)	C:\PROJETA\
F	Base et extension	\$(<F)	MYSOURCE.C
B	Base uniquement	\$(<B)	MYSOURCE
R	Unité, répertoire et base	\$(<R)	C:\PROJETA\MYSOURCE

Les directives de MAKE

Les directives de MAKE ressemblent aux directives des langages tels que Pascal ou C ; elles effectuent diverses opérations de contrôle, telles que l'affichage des commandes avant leur exécution. Les directives de MAKE commencent par un point ou par un point d'exclamation. Le tableau 12.8 ci-dessous décrit les directives et leur option en ligne de commande correspondante (les directives ont priorité sur les options en ligne de commande). Vous trouverez après ce tableau une description détaillée de chaque directive.

Tableau 12.8
Les directives de
MAKE

Directive	Option	Description
.autodepend	-a	Active le contrôle de l'auto-dépendance.
!elif		Equivalent du else if de C
!else		Equivalent du else de C
!endif		Termine une boucle !if, !ifdef ou !ifndef.
!error		Arrête MAKE et affiche un message d'erreur.
!if		Commence une instruction conditionnelle.
!ifdef		Equivalent du ifdef de C, mais se rapporte aux macros et non aux directives #define.
!ifndef		"If not defined" (si non défini)
.ignore	-i	MAKE ignore la valeur de retour d'une commande.
!include		Inclut un fichier dans le fichier MAKE.
!message		Affiche un message depuis le fichier MAKE.
.noautodepend	-a-	Désactive le contrôle de l'auto-dépendance.
.noignore	-i-	Désactive .ignore
.nosilent	-s-	Affiche les commandes avant de les exécuter.
.noswap	-S-	MAKE ne sort pas de la mémoire avant d'exécuter une commande.

Tableau 12.8
Les directives de
MAKE (suite)

Directive	Option	Description
.path.ext		MAKE recherche les fichiers avec l'extension spécifiée dans les répertoires indiqués.
.precious		Conserve les cibles même si le build échoue.
.silent	-s	N'affiche pas les commandes exécutées.
.suffixes		Définit la règle implicite pour les dépendances ambiguës.
.swap	-S	MAKE sort de la mémoire avant d'exécuter une commande.
lundef		Efface une définition de macro.

.autodepend

Les auto-dépendances se produisent dans les fichiers .OBJ ayant des fichiers .CPP, .C ou .ASM correspondant. Si cette option est activée, MAKE compare la date et l'heure des fichiers servant à créer le .OBJ avec celles du fichier .OBJ. Si les premières sont antérieures, le fichier .OBJ est recompilé. Vous pouvez utiliser **.autodepend** ou **-a** à la place des dépendances liées (voir le début du chapitre pour de plus amples informations à ce sujet).

!error

La syntaxe de cette directive est la suivante :

```
!error message
```

MAKE s'interrompt et affiche la chaîne "message", comme dans l'exemple ci-dessous :

```
Fatal makefile exit code: Error directive: message affiché
```

Imbriguez **!error** dans les instructions conditionnelles pour arrêter le traitement et afficher un message d'erreur :

```
!if !$d(MAMACRO)  
# Si MAMACRO n'est pas définie  
!error MAMACRO n'est pas définie...  
!endif
```

Si MAMACRO n'est pas définie, MAKE affiche le message suivant :

```
Fatal makefile exit code: Error directive: MAMACRO n'est  
pas définie...
```

Résumé des contrôles d'erreurs

Il existe quatre façons de désactiver les contrôles d'erreurs :

- La directive **.ignore** désactive le contrôle d'erreur pour la partie sélectionnée du fichier MAKE.

- L'option en ligne de commande **-i** désactive le contrôle d'erreur pour tout le fichier MAKE.
- L'opérateur **-valeur**, saisi comme partie intégrante d'une règle, désactive le contrôle d'erreur pour la commande associée si le code de sortie dépasse la valeur spécifiée.
- L'opérateur **-** désactive le contrôle d'erreur pour la commande associée quel que soit le code de sortie.

!if et autres directives conditionnelles

La directive **!if** a le même effet que l'instruction **if** du C. Comme illustré ci-dessous, la syntaxe de **!if** et des autres directives conditionnelles ressemble à celle des instructions conditionnelles du compilateur :

```

!if condition      !if condition  !if condition
!ifdef macro
                    .
                    .
                    .
!endif             !else           :elif condition  !endif
                    .
                    .
                    .
                    !endif         !endif

```

Les expressions ci-dessous sont équivalentes :

```

!ifdef macro et !if $d(macro)
!ifndef macro et !if !$d(macro)

```

Ces règles s'appliquent aux directives conditionnelles :

- Une directive **!else** est autorisée entre les directives **!if**, **!ifdef** ou **!ifndef** et **!endif**.
- Plusieurs directives **!elif** sont autorisées entre les directives **!if**, **!ifdef** ou **!ifndef** et **!else** et **!endif**.
- Vous ne pouvez pas répartir de règles entre plusieurs directives conditionnelles.
- Vous pouvez imbriquer les directives conditionnelles.
- A chaque **!if**, **!ifdef** et **!ifndef** doit correspondre une instruction **!endif** dans le fichier source.

Les informations ci-dessous peuvent être insérées entre les directives **!if** et **!endif** :

- Définition de macro
- Règle explicite
- Règle implicite
- Directive **!include**
- Directive **!error**
- Directive **!undef**

La *condition* dans une instruction **if** représente une expression conditionnelle consistant en une constante décimale, octale ou hexadécimale suivi des opérateurs du tableau 12.9.

Tableau 12.9
Opérateurs
conditionnels

Opérateur	Description	Opérateur	Description
-	Négation	?:	Expression conditionnelle
~	Complément de bit	!	NON logique
+	Addition	>>	Décalage à droite
-	Soustraction	<<	Décalage à gauche
*	Multiplication	&	ET de niveau bit
/	Division		OU de niveau bit
%	Reste	^	XOR de niveau bit
&&	ET logique	>=	Supérieur ou égal *
	OU logique	<=	Inférieur ou égal *
>	Supérieur	==	Egalité *
<	Inférieur	!=	Inégalité *

* Cet opérateur fonctionne aussi avec les expressions de type chaîne.

MAKE évalue une expression conditionnelle comme un entier simple 32 bits non signé ou comme une chaîne littérale.

!include

Cette directive est semblable à la directive de pré-traitement **#include** de C et de C++ : elle vous permet d'inclure le contenu d'un autre fichier texte dans le fichier MAKE.

```
!include fichier
```

Vous pouvez placer le nom du fichier entre guillemets ou entre les signes `&` et imbriquer les directives sans limite de profondeur, mais vous ne pouvez pas avoir plusieurs directives **!include** dans votre fichier MAKE. Le cas échéant, le message d'erreur *cycle in the include file* s'affiche.

Les règles, les commandes et les directives doivent être complètes au sein d'un fichier source : il n'est pas permis de commencer une commande dans un fichier **!include** et de la finir dans le fichier MAKE.

MAKE recherche les fichiers **!include** dans le répertoire courant si vous n'utilisez pas l'option **-I** pour spécifier un répertoire différent.

!message

La directive **!message** permet d'afficher des messages à l'écran depuis un fichier MAKE. Ces messages vous servent à déboguer un fichier MAKE qui ne fonctionne pas comme vous le souhaiteriez. Ainsi, si une définition de macro vous pose des problèmes, insérez cette ligne dans le fichier MAKE :

```
!message La macro est définie ici comme $(NomDeMacro)
```

Lorsque MAKE interprète cette ligne, il affiche ce message, ainsi que le nom de la macro.

.path.ext

La directive **.path.ext** indique à MAKE où rechercher les fichiers ayant une extension précise. L'exemple ci-dessous indique à MAKE que les fichiers **.C** sont dans **C:\SOURCES** ou dans **C:\FICHIERS** et que les fichiers **.OBJ** sont dans **C:\OBJS**.

```
.path.c = C:\SOURCES;C:\FICHIERS  
.path.obj = C:\OBJS
```

.precious

Si un build de MAKE échoue, le fichier cible est effacé. Cette directive interdit la suppression des fichiers cibles dans ce cas.

La syntaxe est la suivante :

```
.precious: cible [cible] . . . [cible]
```

.suffixes

Cette directive indique à MAKE l'ordre (par extension de fichiers) de création des règles implicites.

La syntaxe est la suivante :

```
.suffixes: .ext [.ext] [.ext] . . . [.ext]
```

où **.ext** représente l'extension du fichier dépendant dans les règles implicites. Par exemple, vous pouvez inclure la ligne `.suffixes: .asm .c .cpp` pour que MAKE interprète les règles implicites en commençant par celles qui dépendent de fichiers .ASM, puis .C, puis .CPP quel que soit leur ordre dans le fichier MAKE.

Les exemples ci-dessous illustrent un fichier MAKE contenant une directive **.suffixes** qui force MAKE à rechercher un fichier source (MONPROG.EXE), d'abord avec l'extension .ASM, puis .C, puis .CPP. S'il trouve le premier, il crée MONPROG.OBJ en chargeant TASM, puis TLINK. S'il ne trouve pas MONPROG.ASM, il passe aux instructions suivantes.

```
.suffixes .asm .c .cpp

monprog.exe: monprog.obj
tlink monprog.obj

.cpp.obj:
    bcc -P $<
.asm.obj:

    tasm /mx $<
.c.obj:
    bcc -P- $<
```

!undef

La syntaxe de cette directive est la suivante :

```
!undef NomDeMacro
```

!undef ("undefine", supprimer la définition) réinitialise la macro spécifiée (*NomDeMacro*). Le test **!ifdef** *NomDeMacro* retourne ensuite la valeur FAUX.

Utilisation de macros dans les directives

La macro **\$d** est utilisée avec la directive **!if** pour effectuer certains traitements si la macro spécifiée est définie. L'instruction **\$d** est suivie d'un nom de macro entre parenthèses ou entre accolades, comme dans l'exemple ci-dessous :

```
!if $d(debug)                #Si DEBUG est définie,  
bcc -v f1.cpp f2.cpp        #compiler avec les informations  
                             #de débogage.  
!else                        #dans le cas contraire,  
bcc -v- f1.cpp f2.cpp      #compiler sans les informations  
                             #de débogage.  
!endif
```

Attention ! N'utilisez pas la macro **\$d** lorsque MAKE est lancé avec l'option **-N**.

Macros vides

Si une macro n'est pas définie, **!ifdef NomDeMacro** retourne la valeur FAUX. Si elle est vide, la valeur VRAI est retournée. Une macro vide ne contient que des espaces (placés à droite du signe égal) ou aucun caractère :

```
MACROVIDE =
```

Les instructions ci-dessous définissent une macro vide dans la ligne de commande MAKE :

```
MACROVIDE = " "
```

ou

```
-DMACROVIDE
```


Utilisation de Resource Workshop

Cette section du *Guide de l'utilisateur* décrit l'emploi de Resource Workshop. Cet outil intègre l'ensemble du processus de conception et de compilation des ressources destinées à des applications sous Microsoft Windows version 3.0 ou plus.

Resource Workshop possède les caractéristiques suivantes =

- Travail sur les ressources en format texte ou binaire. Des éditeurs graphiques vous permettent d'intervenir sur des fichiers binaires, et un éditeur de texte autorise l'édition des ressources sous forme de scripts ressource.
- Gestion facile de centaines de ressources dans des dizaines de fichiers.
- Fonctions d'annulation et de ré-application des modifications sur plusieurs niveaux.
- Décompilation des fichiers ressource binaires de façon à pouvoir modifier les ressources d'un programme même si vous n'avez pas accès au code source.

Quatre fichiers en ligne sont fournis pour Resource Workshop. Ils fournissent des informations très détaillées, qui n'intéresseront que les utilisateurs chevronnés.

- BWCCAPI.TXT Décrit l'interface de programmation (API) BWCC
- CUSTCNTL.TXT Explique comment créer des classes de contrôle personnalisées. Des exemples de code C++ et Pascal sont proposés. Vous devez bien posséder ces notions pour créer vos propres classes de contrôles.

- BWCCSTYL.TXT Expose les critères pris en considération pour la conception des boîtes de dialogue de style Borland avec les contrôles BWCC.
- RWINI.TXT Décrit les options qui ne peuvent être définies que via le fichier WORKSHOP.INI.

Présentation de Resource Workshop

Ce chapitre présente Resource Workshop et les ressources Windows. Il décrit également les différents types de fichiers de ressource de Resource Workshop, et leur rôle dans le cadre d'un projet.

Ressources Windows

Les *ressources* correspondent aux éléments visibles (graphiques) d'une application Windows. Exemple : lorsque vous ouvrez une boîte de dialogue et que vous cliquez sur un bouton pour exécuter une tâche, vous dialoguez avec les ressources du programme. Les ressources Windows sont décrites ci-dessous en détail.

En général, les ressources des applications Windows sont distinctes du code du programme, ce qui permet d'apporter des modifications à l'interface utilisateur sans avoir à intervenir sur le fichier contenant ce code.

Les ressources existantes et les ressources que vous créez ou personnalisez peuvent être réutilisées dans différentes applications.

Resource Workshop supporte les ressources Windows suivantes :

- *Boîtes de dialogue*. Fenêtre (généralement de type "surgissante") qui permet les interactions entre l'application et l'utilisateur : l'application fournit (affiche) les informations destinées à l'utilisateur, et l'utilisateur sélectionne parmi les options présentées pour ouvrir un fichier, modifier des couleurs, rechercher du texte, etc.
- *Menus*. Les applications Windows disposent d'une "barre menu", qui affiche le nom des menus disponibles. Chaque menu contient plusieurs éléments de menu (commandes). Exemple : le menu Fichier, présent dans la plupart des applications Windows, comprend des commandes de création, ouverture, enregistrement et impression de fichiers.

- *Raccourcis*. Touche ou combinaison de touches permettant de lancer une tâche rapidement, sans passer par les menus. Exemple : certaines applications Windows disposent du raccourci *Maj+Insérer* pour la commande Coller, qui permet de transférer le contenu du Presse-papiers Windows dans le fichier actif. En général, les raccourcis apparaissent dans le menu, à droite de la commande à laquelle ils sont associés.
- *Tables de chaînes*. Table contenant les chaînes de texte (messages système, messages d'erreur, descriptions diverses, etc.) nécessaires à une application Windows. Ces chaînes étant des ressources Windows distinctes du programme, vous pouvez les modifier sans avoir à intervenir sur le code source du programme.
- *Graphiques par point*. Représentation binaire d'une image graphique, par exemple les flèches des barres de défilement, la case du menu Système ou les triangles de Réduction/Agrandissement/Restauration. Chaque bit ou groupe de bits du graphique correspond à un pixel de l'image.
- *Icônes*. Petit graphique par points de dimensions variables : 64×64 pixels (sous Windows 3.1 et moniteurs à très haute résolution), 32×32 pixels ou 16×32 pixels qui représente une fenêtre en réduction. Vous pouvez créer des icônes à l'aide de l'éditeur Bitmap (décrit au chapitre 19) .
- *Curseurs*. Petit graphique par points de 32×32 pixels qui matérialise sur l'écran la position de la souris. Dans les applications Windows, le curseur change de forme en fonction de la nature de la tâche en cours. Vous pouvez créer des curseurs à l'aide de l'éditeur Bitmap (décrit au chapitre 19).

Pour voir un exemple de curseurs personnalisés, lancez l'éditeur Bitmap pour éditer une ressource graphique (graphique par points, icône, curseur ou police). Chaque fois que vous sélectionnez un nouvel outil de dessin et que vous déplacez le curseur sur l'image en cours de modification, le curseur prend une forme correspondant à sa fonction.

- *Polices*. En principe, le terme "police" désigne un ensemble de caractères texte ; dans Resource Workshop, une police est plus généralement un ensemble de graphiques par points stockés et utilisés ensemble.

Les applications Windows disposent de nombreuses polices dont les caractéristiques conditionnent l'aspect des caractères : type (Helvetica, Times Roman, etc.), taille (exprimée en points) et style (gras, italiques, etc.). Par exemple, vous pouvez afficher ou imprimer un texte en police **Times Roman 10 points gras**.

Resource Workshop dispose des fonctions permettant de modifier les polices existantes et de créer des polices personnalisées.

- *Ressources utilisateur et ressources RC*. Ces ressources, qui sont sensiblement identiques dans Resource Workshop, contiennent des données que vous pouvez ajouter dans un fichier exécutable. Exemple : si vous utilisez un gros bloc de données initialisées et accessibles en lecture seule (par exemple, un fichier texte), vous pouvez l'insérer dans un fichier exécutable en tant que ressource définie par l'utilisateur.

L'ajout de ressources utilisateur dans une application permet d'optimiser la gestion de la mémoire. En effet, de nombreuses applications Windows utilisent le modèle de mémoire intermédiaire, qui comprend un seul segment de données. Si vous disposez d'une grande quantité de données et que vous ne souhaitez pas que ces données résident en mémoire de façon permanente, il vous suffit de les enregistrer comme ressource définie par l'utilisateur ; elles sont alors chargées uniquement sur ordre du programme.

- *VERSIONINFO*. Resource Workshop supporte la ressource VERSIONINFO (gestionnaire de version des fichiers .EXE de Windows 3.1). Pour plus de détails sur VERSIONINFO, consultez l'aide en ligne de Resource Workshop ou la documentation de programmation de Windows 3.1.

Pour l'édition des ressources, Resource Workshop dispose de deux éditeurs : un éditeur graphique convivial et performant, et un éditeur de texte destiné à la création des scripts ressources.

En outre, vous pouvez spécifier un éditeur externe pour travailler avec les fichiers projet qui ne subissent pas de compilation à l'ouverture.

Types de fichiers ressources

Les fichiers créés ou édités sous Resource Workshop peuvent être au format binaire ou au format texte. Resource Workshop supporte également les formats Windows standard ; vous pouvez donc utiliser des fichiers Resource Workshop avec des programmes qui génèrent du code binaire à partir de fichiers scripts de ressources.

- *Fichier de compilation (.RC)*. Un fichier de compilation (extension .RC) est un fichier script (texte) qui contient les définitions d'une ou plusieurs ressources, définies sous forme de script ou de références pointant sur d'autres fichiers contenant des ressources.

En général, un projet Resource Workshop doit comporter au moins un fichier .RC

- *Fichier compilé*. Un fichier compilé (extension .RES) contient une ou plusieurs ressources compilées.
En général, lors de la création d'une application Windows, toutes les ressources sont compilées dans un seul fichier .RES, ce dernier étant ensuite lié au fichier exécutable. Avec Resource Workshop, il n'est pas nécessaire de générer de fichier .RES, car Resource Workshop peut compiler les fichiers ressources et les lier directement à un fichier exécutable.
- *Fichier exécutable et fichier DLL*. Le fichier exécutable (extension .EXE) ou DLL est la destination finale des ressources définies sous Resource Workshop. En général, vous compilez un fichier .RC pour obtenir un fichier .RES, puis vous lancez un compilateur pour lier le fichier .RES au fichier exécutable ou DLL. Resource Workshop peut lier directement les ressources aux fichiers exécutables ou DLL, ce qui évite l'emploi du compilateur de ressources Microsoft.
- *Fichier .DRV*. Un fichier .DRV est un pilote de périphérique Windows, c'est-à-dire un cas particulier de fichier DLL. Les ressources de ces fichiers peuvent être éditées comme celles d'un fichier DLL.
Si vous devez modifier les ressources d'un fichier binaire compilé (fichier exécutable, DLL ou .RES), vous pouvez demander à Resource Workshop de décompiler le fichier, apporter les modifications requises et réenregistrer les ressources modifiées dans leur fichier binaire d'origine.
- *Fichier dialogue (.DLG)*. Un fichier dialogue (.DLG) est un fichier script (texte) de ressources qui contient généralement la description d'une ou plusieurs boîtes de dialogue, mais qui peut également contenir toute autre ressource de fichier RC.

Fichiers ressources bitmap

Il existe quatre types de fichiers ressources bitmap :

- Le fichier bitmap (extension .BMP) contient une ressource bitmap au format binaire.
- Le fichier icône (extension .ICO) contient une icône au format binaire.
- Le fichier curseur (extension .CUR) contient un curseur personnalisé au format binaire.

- Les fichiers polices existent sous forme binaire et sous forme de bibliothèques :
 - Le fichier polices binaire (extension .FNT) contient la définition d'une police personnalisée au format binaire. Vous pouvez utiliser l'éditeur Paint de Resource Workshop pour créer une police et la mémoriser dans un fichier .FNT.
 - Le fichier bibliothèque de polices (extension .FON) est une bibliothèque à liaisons dynamiques (DLL) contenant un répertoire de polices et une ou plusieurs polices. Les fichiers .FON doivent être créés à l'extérieur de Resource Workshop ; une fois le fichier .FON créé, vous pouvez utiliser Resource Workshop pour le modifier.

Projets, ressources et identificateurs

Ce chapitre décrit la fenêtre projet. Il explique comment contrôler, dans cette fenêtre, l'affichage des ressources, mais aussi leur choix en vue d'une édition. Il couvre aussi certains sujets généraux concernant les ressources : comment les ajouter ou les effacer des projets, comment les sauvegarder, et comment manipuler les identificateurs.

Comme indiqué au chapitre 2, un *projet* est une collection d'une ou plusieurs ressources. Un fichier de projet (en général, .RC) contient une ou plusieurs ressources, fait référence à des fichiers contenant des ressources, ou encore les deux.

Création d'un projet

Pour créer un projet :

Si vous avez un projet Ouvert, Resource Workshop le ferme d'abord (s'il a été modifié, un message vous demande si vous souhaitez enregistrer les modifications avant de le fermer).

1. Choisissez File | New Project.

Décidez du type de fichier sur lequel vous voulez baser votre projet. Un projet caractéristique est basé sur un fichier .RC, car ce type de fichier vous permet de travailler avec les différents types de ressources.

Cependant, vous pouvez aussi choisir :

- .RC pour créer un fichier ressources script
- .RES pour travailler avec un fichier ressources binaire
- .CUR pour créer un projet contenant un curseur
- .ICO pour créer un projet contenant une icône
- .BMP pour créer un projet contenant un graphique par points
- .FNT pour créer un projet ne contenant que des polices

2. Cliquez sur le type de fichier projet souhaité, puis sur OK.
Resource Workshop affiche votre nouveau projet, sans titre, dans la fenêtre projet. Pour les projets .RC, vous devez spécifier un fichier en-tête par défaut dans lequel Resource Workshop enregistrera les identificateurs.

Ouverture d'un projet existant

Un projet existant peut être un projet créé avec Resource Workshop, ou un fichier .RC créé avec un autre logiciel de développement de ressources. Vous pouvez aussi travailler avec les ressources d'une application Windows 3.0 ou ultérieure, même si vous n'avez pas accès au code source. Si vous ne pouvez accéder qu'à un fichier exécutable, Resource Workshop décompile les ressources liées à ce fichier pour que vous puissiez les modifier.

Pour ouvrir un projet existant :

1. Choisissez File | Open Project. Resource Workshop affiche la boîte de dialogue Open Project.
2. Spécifiez le fichier contenant le projet à ouvrir par l'une des procédures suivantes :
 - Tapez le chemin d'accès et le nom du fichier et appuyez sur *Entrée*.
 - Choisissez un type de fichier, puis un nom de fichier dans la liste Files et cliquez sur OK.

Resource Workshop réagit différemment selon que le projet est un fichier binaire ou un fichier contenant des données ressource.

- Si le projet est un fichier binaire (fichier exécutable, fichier .RES ou fichier DLL), Resource Workshop décompile les ressources en indiquant la progression de l'opération, sur le côté gauche de la ligne d'état.
- Si le projet est un fichier RC ou un autre type de fichier contenant des données ressource (tel est généralement le cas), Resource Workshop lit le fichier directement (puisque aucune décompilation n'est nécessaire), puis compile le fichier et chaque ressource en vous indiquant la progression de l'opération dans la boîte de dialogue Compile Status.

Si le compilateur rencontre une erreur, Resource Workshop affiche la boîte de dialogue Compiler Error, qui contient la ligne (en surbrillance) où se trouve l'erreur.

Pour éditer le fichier avec l'éditeur de texte externe spécifié dans File | Preferences, appuyez sur le bouton Run Editor (boîte de dialogue Compiler Error). Lorsque l'éditeur apparaît, apportez les modifications requises, enregistrez-les et quittez l'éditeur. Vous devez ensuite recharger le projet.

3. Une fois le projet compilé ou décompilé, Resource Workshop affiche la fenêtre projet énumérant toutes les ressources.

Comment utiliser la fenêtre projet

Lorsque vous créez un projet ou que vous ouvrez un projet existant, Resource Workshop affiche la fenêtre projet.

La fenêtre projet agit comme un outil de gestion des fichiers, ce qui facilite la vue générale d'un projet. Même si le projet contient de nombreuses ressources, vous pouvez rapidement les parcourir en faisant défiler le contenu de la fenêtre projet.

Pour un nouveau projet, la fenêtre est vide, et vous devez y introduire des ressources en les créant ou en les ajoutant en tant que fichiers (ces sujets seront approfondis ultérieurement).

Pour un projet existant, vous pouvez voir :

- La liste complète des fichiers du projet
- Les types de ressources de chaque fichier
- Les identificateurs (**#define** ou constantes) associés aux ressources, si le fichier contient des données ressource (ce n'est pas un fichier .EXE, .RES ou .DLL). Les identificateurs sont traités dans la section "Comment utiliser les identificateurs" de ce chapitre.

Ressources intégrées ou liées

Les ressources de votre fichier projet peuvent être *intégrées* dans le fichier ou *liées* à celui-ci.

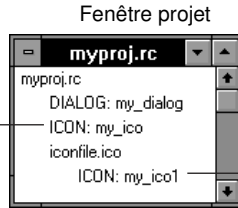
- Une ressource intégrée est stockée sous la forme d'un script ressource dans le fichier projet. Elle existe uniquement en tant que partie du projet dans lequel elle est stockée, et ne peut pas être utilisée dans d'autres projets.
- Une ressource liée est un fichier distinct référencé dans le fichier projet. Les ressources liées de type .RC, .DLG ou de format binaire peuvent être utilisées dans d'autres projets.

Affichage d'informations dans la fenêtre projet

Utilisez le menu View pour déterminer le mode d'affichage des informations dans la fenêtre projet. Par défaut, la fenêtre projet affiche la ressource sélectionnée dans une fenêtre de prévisualisation. Utilisez View | Hide preview pour désactiver la prévisualisation (dans la figure 14.1, la prévisualisation est désactivée). Vous pouvez également décider de l'emplacement de la prévisualisation par rapport au cadre du projet en choisissant View | Show vertical preview ou View | Show horizontal preview.

Figure 14.1
Fenêtre projet montrant les ressources par fichier

Cette ressource ICON est dans le fichier projet où elle est stockée en format script ressource.



Cette ressource est un fichier distinct ICONFILE.ICO, en format binaire (comme toutes les ressources graphiques stockées dans des fichiers externes)

Show Identifiers

Affiche les identificateurs (**#define** ou constantes) du projet. Les identificateurs sont traités dans la section "Comment utiliser les identificateurs" de ce chapitre.

Show Resources

Enumère les types et les noms des ressources, tel que BITMAP: airplane. Dans la plupart des cas, cette option devra être activée.

Désactivez l'option Show Resources si vous ne désirez voir que les noms de fichiers, sans la liste des ressources qu'ils contiennent ou si seuls les identificateurs vous intéressent. Si Show Resources est désactivé, aucune des ressources définies dans le fichier projet ne peut être sélectionnée pour édition.

Show Items

Affiche un autre niveau de détail dans la fenêtre projet. Lorsque Show Items est activé, Resource Workshop affiche les éléments de chaque ressource (par exemple, les POPUP et MENUITEM définis dans une ressource menu).

Show Unused Types

Répertorie tous les types de ressources, qu'ils soient ou non utilisés dans le projet.

Show Unused Types est uniquement disponible lorsque View | By Type est activé.

Sélection d'une ressource

Pour sélectionner une ressource, utilisez la souris ou les touches fléchées pour la mettre en surbrillance dans la fenêtre projet.

- Si vous avez choisi View | By Type, recherchez d'abord le type de ressource. La ressource est listée par nom, sous le type de ressource.
- Si vous avez choisi View | By File, recherchez la ressource sous son nom de fichier. Le nom de ressource est précédé par le type de ressource et le signe deux points (:).

Par exemple, dans la fenêtre projet myproj.rc de la figure 14.1, la ressource Icône my_ico1 apparaît sous le fichier iconfile.ico, sous le nom ICON:my_ico1.

Comment travailler avec les ressources

Cette section décrit les tâches exécutées avec les ressources.

Chargement d'une ressource

Pour charger une ressource en vue de l'attribution d'un nom ou d'une édition, exécutez l'une des procédures suivantes :

- Double-cliquez le nom de ressource (et non le type de fichier) dans la fenêtre projet. Resource Workshop lance automatiquement l'éditeur graphique approprié. Si aucun éditeur de ressources n'est disponible (par exemple, pour une ressource définie par l'utilisateur), Resource Workshop lance l'éditeur de texte interne.
- Sélectionnez le nom de ressource dans la fenêtre projet, puis choisissez soit Resource | Edit (pour ouvrir un éditeur graphique), soit Resource | Edit as Text (pour ouvrir l'éditeur de texte).

Éditeurs de ressources

Chaque éditeur de ressources est décrit dans le chapitre consacré à la ressource correspondante

Lorsque vous double-cliquez sur l'un des types de ressource suivants dans la fenêtre projet (ou que vous sélectionnez un type et choisissez *Ressource | Edit*), Resource Workshop charge l'éditeur de ressources approprié :

- Si la ressource est une boîte de dialogue, un menu, un raccourci ou une table de chaînes, Resource Workshop charge un éditeur spécialement conçu pour ce type de ressource. Pour les descriptions de ces éditeurs, reportez-vous aux chapitres 15 à 18.
- Si la ressource est une ressource bitmap (icône, graphique par points, curseur ou police), Resource Workshop l'affiche dans l'éditeur Bitmap.

L'éditeur de texte interne

Pour une description du langage script de ressource, reportez-vous à l'index de l'aide en ligne.

Les ressources définies par l'utilisateur et deux types de ressources prédéfinis (RCDATA et VERSIONINFO) peuvent uniquement être édités au moyen de l'éditeur de texte interne.

Si la ressource est au format binaire (comme une boîte de dialogue ou un menu), Resource Workshop la décompile pour vous permettre de travailler avec le script ressource.

L'éditeur de texte interne est similaire à l'éditeur Notepad de Windows. Il a recours aux touches *Suppr*, *Début*, *Fin*, *PgPréc*, *PgSuiv*, *Tab* et *Retour arrière*. Il est toujours en mode insertion.

Lorsque vous entrez du texte, ne perdez pas de temps à le formater, Resource Workshop réorganise le texte à votre place lors de la compilation de la ressource.

Si Resource Workshop identifie une erreur, il vous indique sa nature et vous ramène dans l'éditeur de texte pour la corriger.

Si vous désirez directement éditer le script ressource sans l'assistance de Resource Workshop, ouvrez le fichier source avec l'éditeur de votre choix, puis éditez-le. Cependant, notez le point suivant :



Lorsque Resource Workshop charge la ressource, il recompile le fichier et reformate le script ressource. S'il rencontre des commentaires dans le script à l'ouverture de la ressource pour édition, il affiche une boîte de dialogue précisant que ces commentaires seront supprimés.

Ajout d'une ressource

Vous pouvez ajouter directement une ressource à un projet (ressource intégrée), ou l'ajouter en tant que référence à un fichier (ressource liée).

**Ajout d'une
ressource intégrée**

Pour ajouter une ressource à votre projet , ouvrez celui-ci et suivez les étapes ci-dessous :

1. Choisissez Resource | New pour afficher la boîte de dialogue New Resource.
2. Choisissez les fichiers en-tête dans lesquels vous souhaitez faire apparaître la ressource et son identificateur. Vous pouvez choisir uniquement parmi les fichiers en-tête associés à votre projet ; pour associer un nouveau fichier en-tête à votre projet, utilisez File | Add to project.
3. Double-cliquez sur le type de ressource à créer.
 - Si vous créez un raccourci, un menu, une boîte de dialogue ou une table de chaînes, Resource Workshop place une entrée pour la ressource dans la fenêtre projet et ouvre l'éditeur de ressources approprié.
 - Si vous créez une ressource de données RC (rcdata) ou une ressource VERSIONINFO, Resource Workshop place une entrée pour la ressource dans la fenêtre projet, et ouvre l'éditeur de texte interne.
 - Si vous créez une ressource bitmap (icône, curseur, graphique par points ou police), Resource Workshop vous demande si vous voulez enregistrer la ressource comme texte source ou au format binaire. Pour intégrer une ressource bitmap, sélectionnez Source text.
 - Si le type de ressource à créer n'est pas répertorié, appuyez sur le bouton New Type afin de créer un type défini par l'utilisateur. Pour plus d'informations sur les ressources définies par l'utilisateur, reportez-vous au chapitre 23.

**Ajout d'une
ressource liée**

Vous pouvez ajouter une ressource en la liant, c'est-à-dire en créant une référence :

1. Choisissez File | Add to Project. Resource Workshop affiche la boîte de dialogue Add File to Project.

Vous pouvez aussi entrer des critères de recherche de fichier en sélectionnant le type du fichier dans la boîte liste File Type.

2. Tapez le nom du fichier contenant la ressource dans la boîte File Name, ou cliquez deux fois le nom du fichier s'il est énuméré dans la boîte Files. Si le fichier n'est pas dans le répertoire actif ou si son type ne correspond pas au type en cours, vous pouvez le sélectionner de deux manières :
 - Vous pouvez taper le nom et le chemin d'accès complet du fichier dans la boîte File Name.
 - Vous pouvez changer de répertoire en utilisant la boîte liste Directories. Entrez ensuite vos critères de recherche dans la boîte File Name et appuyez sur *Entrée* (par exemple, utilisez *.CUR pour trouver tous les fichiers curseur), ou sélectionnez un type dans la boîte liste File Type. Lorsque le nom de fichier désiré apparaît dans la boîte liste Files, effectuez un double-clic pour sélectionner le fichier.
3. Dans la seconde liste déroulante (sous la liste déroulante File Type), vous voyez le fichier projet en cours, où vous placerez très probablement la référence au nouveau fichier. Si votre projet contient plusieurs fichiers .RC et que vous souhaitez placer la référence ailleurs, faites défiler la liste et trouvez le nom du fichier dans lequel vous voulez la placer.
4. Appuyez sur *Entrée* ou cliquez sur OK pour ajouter le fichier au projet. Dans la fenêtre projet, Resource Workshop place une entrée qui pointe sur ce fichier.

Si vous choisissez View | By File, vous verrez le nom de fichier énuméré et, au-dessous, le nom de la ressource. Toute modification effectuée sur cette ressource dans le projet est répercutée dans le fichier ressources d'origine.

Pour ajouter une nouvelle ressource selon la même méthode,

1. Dans la boîte File Name de la boîte de dialogue Add File to Project, entrez le nom du fichier à créer. Si vous n'utilisez pas l'une des extensions standard des fichiers de ressources (.CUR, par exemple), servez-vous de la boîte File Type pour identifier le type de ressource voulu.
2. Cliquez sur OK. Une bote de dialogue vous indique que le fichier n'existe pas et demande confirmation pour le créer.
3. Cliquez sur OK pour créer le fichier. Resource Workshop insère une référence à ce fichier dans la fenêtre projet.

**Comment déplacer
une ressource**

Vous pouvez déplacer des ressources d'un fichier de projet à un autre (du même projet), à l'aide de **Ressource | Move**.

1. Sélectionnez la ressource à déplacer en l'éditant ou en affichant son nom en surbrillance dans la liste des ressources du projet.
2. Choisissez **Ressource | Move**. La boîte de dialogue qui s'affiche contient le nom de cette ressource et du fichier qui la contient (signalé par la mention "Old").
3. Sélectionnez un fichier New (le fichier dans lequel vous souhaitez faire apparaître la ressource) et cliquez sur OK

**Comment copier
une ressource
entre projets**

Pour copier une ressource du projet actif (le projet "source") à un autre projet (le projet "cible"), vous disposez de deux solutions :

- Enregistrer la ressource sous forme de fichier, fermer le projet actif, ouvrir le projet cible et y ajouter la ressource. Si la ressource comporte des identificateurs, vérifiez que ceux-ci sont conservés lors de cette opération.
- Ouvrir deux copies de **Resource Workshop** (une par projet), et utiliser le **Presse-papiers** de Windows pour copier la ressource dans le projet source et la coller dans le projet cible. Cette méthode est plus rapide et garantit la conservation des identificateurs

Pour copier une ressource par la deuxième méthode :

1. Ouvrez deux exemplaires de **Resource Workshop** : l'un avec le projet contenant la ressource à copier (projet source), l'autre avec le projet vers lequel vous souhaitez la copier (projet cible).
2. Vérifiez que le projet cible contient une référence pointant sur un fichier d'identificateurs qui pourra recevoir les identificateurs de la nouvelle ressource (si nécessaire, choisissez **File | Add to Project** et ajoutez ce fichier).
3. Dans la fenêtre **Projet**, sélectionnez le projet source, puis la ressource à copier. Choisissez **Edit | Copy** pour copier la ressource dans le **Presse-papiers**.
4. Sélectionnez le projet cible, vérifiez que la fenêtre **Project** est active et choisissez **Edit | Paste** pour coller la ressource dans ce projet. **Resource Workshop** affiche la boîte de dialogue **Paste Resource**.

N'utilisez pas le raccourci **Ctrl+Ins**, car la copie ne s'effectuerait pas de manière correcte.

5. La boîte liste Paste Resource Into doit contenir le nom du projet cible. Vérifiez que la boîte liste Paste Identifiers Into affiche le nom du fichier qui doit recevoir les identificateurs de la ressource (si nécessaire, déroulez la liste et sélectionnez le nom de fichier approprié). Appuyez ensuite sur *Entrée* ou cliquez sur OK pour coller la nouvelle ressource dans le projet cible.

Comment supprimer une ressource

Pour supprimer une ressource d'un projet, sélectionnez-la dans la fenêtre projet et choisissez Edit | Delete ou Edit | Cut (dans ce dernier cas, vous avez la possibilité de coller la ressource à un autre emplacement).

Comment renommer une ressource

Pour renommer une ressource :

1. Choisissez Resource | Rename. Resource Workshop affiche la boîte de dialogue Rename Resource.
2. Dans la boîte New Name, tapez le nouveau nom de la ressource et appuyez sur *Entrée*.
3. Resource Workshop vous demande si vous souhaitez créer un nouvel identificateur portant ce nom.
 - Si vous cliquez sur Yes, Resource Workshop renomme la ressource et lui affecte la valeur d'identificateur spécifiée.
 - Si vous cliquez sur No, Resource Workshop renomme la ressource sans créer d'identificateur. Pour plus de détails sur les avantages de la création d'identificateurs, reportez-vous à la section "Comment utiliser les identificateurs".

Reportez-vous à la section "Ajout d'identificateurs" plus loin dans ce chapitre.

Spécification des options de mémoire ressource

Resource Workshop vous permet de spécifier la façon de gérer en mémoire chaque ressource de votre projet. Cependant, si vous n'êtes pas un programmeur Windows expérimenté, il est sans doute préférable de conserver les valeurs par défaut.

Pour spécifier les options de mémoire, sélectionnez la ressource dans la fenêtre projet, puis choisissez Resource | Memory Options. La boîte de dialogue Resource Memory Options apparaît.

Désactivez toute option de mémoire non désirée (voir tableau 14.1).



Si vous définissez les options de mémoire pour une ressource Icône, ces options s'appliquent à toutes les images de cette ressource.

Voici les effets de chaque option de la boîte de dialogue Resource Memory Options :

Tableau 14.1
Options mémoire
des ressources

Option	Description
Load on Call	<p>Cette option ne charge la ressource en mémoire que lorsque cette ressource est nécessaire. Le choix de Load On Call peut réduire le temps nécessaire pour charger votre programme.</p> <p>Si vous désactivez cette option, vous activez Preload, ce qui signifie que Windows charge la ressource en mémoire lors du chargement initial du programme. Vous ne devez précharger une ressource que si vous savez que Windows l'utilise dès le début de l'exécution de l'application.</p>
Moveable	<p>Cette option permet à Windows de déplacer le segment ressource en mémoire pour ménager de la place en vue d'autres affectations de la mémoire.</p> <p>Si vous désactivez cette option, le segment ressource occupe un bloc fixe en mémoire.</p>
Discardable	<p>Permet à Windows d'éliminer le segment ressource de la mémoire lorsqu'il n'est plus nécessaire. Windows peut recharger la ressource en mémoire lorsqu'elle redevient utile.</p> <p>Si vous désactivez cette option, vous activez Nondiscardable. Windows ne pourra pas retirer le segment de la mémoire pendant l'exécution de l'application, et si l'option Pure n'est pas activée, vous pourrez modifier la ressource depuis votre application. Notez que Nondiscardable n'est pas compatible avec RC.EXE.</p>
Pure	<p>Empêche la modification du segment ressource en mémoire.</p> <p>Généralement, vous laissez cette option activée. Pour plus d'informations, reportez-vous à la documentation Windows.</p>

Comment utiliser les identificateurs

Windows requiert que chaque ressource soit associée à un nom unique ou à un entier unique (appelé *identificateur de ressource*). Par défaut, Resource Workshop affecte un nom à chaque nouvelle ressource (par exemple, DIALOG_1 pour une boîte de dialogue ou MENU_1 pour une ressource menu).

Le nom par défaut n'est pas très descriptif, et la référence à une ressource seulement par son nom diminue le rendement d'une application lors de son exécution. Pour faire face à ces limitations, vous pouvez renommer la ressource et lui affecter un *identificateur* (constante Pascal ou **#define** en C). L'emploi d'identificateurs permet d'obtenir à la fois des noms significatifs et un meilleur rendement à l'exécution.

- Vous pouvez renommer la ressource en une valeur entière. L'emploi d'une valeur entière a le même effet sur le rendement de l'application à l'exécution que l'affectation d'un identificateur, avec le désavantage que l'identificateur de ressource n'est toujours pas très descriptif. Par exemple, DIALOG:225 ne précise pas de quelle boîte de dialogue il s'agit.
- Vous pouvez renommer la ressource sans créer d'identificateur. Vous obtenez alors une ressource avec un nom descriptif et un code plus facile à gérer qu'un identificateur (les paramètres Windows s'attendent à un pointeur **char** de type far), mais le rendement lors de l'exécution de l'application sera inférieur.

Composants des identificateurs de ressource

Un identificateur est composé de deux parties : un texte littéral (le nom d'identificateur) et une valeur (généralement un entier). Par exemple, l'instruction suivante déclare un identificateur portant le nom dlg_OpenFile et ayant la valeur 100 :

```
#define dlg_OpenFile 100
```

Resource Workshop affiche cette ressource dans la fenêtre projet sous le nom DIALOG: dlg_OpenFile, qui l'identifie comme la boîte de dialogue Open File.

Les identificateurs doivent être uniques à l'intérieur d'un type de ressource. Seuls les 31 premiers caractères sont significatifs (Resource Workshop ignore tout caractère au-delà des 31 premiers). Vous pouvez modifier la valeur des identificateurs dans les boîtes de saisie. Exemple : dans l'éditeur Menu, vous pouvez taper l'identificateur `CM_FILENEW=125` ; lorsque vous quittez la boîte de saisie, Resource Workshop fait passer la valeur de l'identificateur `CM_FILENEW` à 125.

L'affectation d'une valeur entière à un identificateur accélère les appels à la ressource pendant l'exécution, mais vous ne pourrez pas utiliser directement la valeur entière courte (short) en tant que paramètre. Vous devez soit transtyper l'entier en un pointeur **char** de type far, soit utiliser une macro qui réalisera le transtypage à votre place.

Si vous écrivez votre programme en C ou en C++, vous pouvez utiliser la macro `MAKEINTRESOURCE`. Si vous écrivez votre programme en Pascal, vous pouvez utiliser le type `MakeIntResource` (un pointeur sur *char*).

La macro `MAKEINTRESOURCE` ressemble à un appel de fonction mais effectue en fait un transtypage sur l'identificateur. Par exemple, pour utiliser `dlg_OpenFile` en tant que paramètre dans un programme C ou C++, entrez-le sous la forme suivante :

```
MAKEINTRESOURCE (dlg_OpenFile)
```



Si vous travaillez avec un fichier .RES, un fichier exécutable ou un fichier DLL, Resource Workshop décompile tous les identificateurs de ressource de ce fichier en valeurs entières. Vous ne pouvez pas ajouter des identificateurs à ce type de fichier, mais vous pouvez enregistrer le fichier en tant que fichier .RC, puis affecter des identificateurs à ses ressources. Reportez-vous à la section "Comment travailler avec des fichiers binaires" de ce chapitre.

Fichiers identificateurs

Lorsque vous ouvrez un nouveau projet, la première chose à faire est de spécifier un fichier pour y stocker vos identificateurs.

Stockez vos identificateurs dans un ou plusieurs fichiers en-tête (.H) qui utilisent des instructions **#define** pour affecter des valeurs aux noms d'identificateurs.

Dans ce manuel, le terme *fichiers identificateurs* correspond aux fichiers en-tête, aux unités et aux fichiers inclus.

Vous pouvez utiliser un éditeur de texte ou un traitement de texte pour créer vos fichiers identificateurs, mais il est plus facile de laisser Resource Workshop le faire à votre place, comme illustré dans les paragraphes qui suivent et dans la section "Ajout d'identificateurs" de ce chapitre.

Création des fichiers identificateurs

Après l'ouverture d'un nouveau projet (File | New Project) et l'affectation d'un nom à ce projet (File | Save Project), ajoutez le fichier identificateur en suivant les étapes ci-dessous :

1. Choisissez File | Add to Project. Resource Workshop affiche la boîte de dialogue Add File to Project.
2. Cliquez sur le bouton *flèche vers le bas* de la boîte liste File Type pour afficher la liste des types de fichiers que vous pouvez ajouter à votre projet.
3. Si vous écrivez votre application en C, choisissez :

```
H c header
```
4. Dans la boîte texte File Name, tapez le nom du fichier identificateur.
5. Cliquez sur OK pour quitter la boîte de dialogue Add File to Project. Resource Workshop crée le fichier identificateur.

Fichiers en-tête C

Si votre programme est écrit en langage C, stockez vos identificateurs dans un fichier en-tête .H. Les instructions **#define** de ce fichier affectent des valeurs entières aux noms d'identificateurs.

Exemple extrait d'un fichier en-tête :

```
/*
 * #define sélectionnés depuis RWCDEMO.H
 */

#define bmp_StatusBar
#define cm_About_CUA
#define id_ClearWindow
#define dlg_About
#define dlg_FileNew
#define sth_Edit
#define men_Main
#define acc_Main
#define ico_RWCDemo
#define sth_EditClear
#define ScribbleWindow
#define FileWindow
#define GraphWindow
```

En plus des instructions **#define**, vous pouvez aussi stocker des définitions de structure et de type, du code de programme et des commentaires dans un fichier en-tête. Resource Workshop ignore toutes les données dans le fichier en-tête, à l'exception des instructions **#define** et des directives de préprocesseur.

Gestion automatique des identificateurs

Resource Workshop peut créer et supprimer automatiquement les identificateurs. Pour activer la fonction de gestion automatique des identificateurs, choisissez File | Preferences et cochez Generate identifiers automatically. Remarque : si vous utilisez AppExpert à partir de l'environnement EDI de Borland C++, cette case est cochée par défaut et indisponible (en effet, AppExpert applique systématiquement la gestion automatique des identificateurs).

Lorsque la gestion automatique des identificateurs est activée, chaque fois que vous créez une ressource qui utilise un identificateur (par exemple, un élément de menu), Resource Workshop définit un identificateur unique pour cette ressource et le place dans le fichier en-tête de cette ressource (.RH ou .RC). Par ailleurs, si vous supprimez une ressource, son identificateur est automatiquement supprimé.

Resource Workshop affecte chaque identificateur d'un préfixe, que vous pouvez modifier. Le tableau ci-dessous fournit la liste des préfixes par défaut et indique les commandes de menu à utiliser pour les modifier (pour que le menu approprié soit affiché, vous devez éditer le type de ressource correct) :

Tableau 14.2
Préfixes des
identificateurs

Ressource	Préfixe	Commande de menu
Table de chaînes	IDS_	String Table Change identifier prefix
Menu	CM	Menu Change identifier prefix
Accélérateur	CM	Accelerator Change identifier prefix
Boîte de dialogue	IDC	Options Change identifier prefix

Comment travailler sans fichier d'identificateurs

Si vous n'ajoutez pas de fichier d'identificateurs à votre projet, vous pouvez néanmoins créer des identificateurs pour vos ressources. Resource Workshop stocke ces identificateurs dans le fichier projet actif, sous forme d'instructions **#define**.

Cette méthode n'est pas conseillée pour les raisons suivantes :

- Si vous décidez par la suite d'utiliser un fichier d'identificateurs distinct, vous devrez faire appel à un éditeur de texte pour couper les instructions **#define** dans les scripts de ressources et les coller dans ce fichier d'identificateurs.
- Le fichier script de ressources acceptant uniquement des instructions **#define**, vous devrez prévoir une opération d'édition supplémentaire pour créer une unité Pascal ou un fichier de constantes.

Ajout d'identificateurs

Il existe deux méthodes pour ajouter des identificateurs à votre projet :

- En renommant votre ressource
- A l'aide de la boîte de dialogue Identifiers

En renommant les ressources

Comme spécifié à la section "Comment renommer une ressource" plus haut dans ce chapitre, vous pouvez renommer vos ressources à mesure que vous les créez. Lorsque Resource Workshop rencontre un nouveau nom de ressource, il vous demande automatiquement si vous désirez créer un identificateur pour ce nom.

Exemple : vous venez d'ajouter une nouvelle ressource menu à votre projet. Par défaut, Resource Workshop lui affecte le nom MENU_1 (ce menu est le premier dans le projet). Puisqu'il s'agit d'un menu fichier, vous décidez de le renommer en Menu_File : choisissez Resource | Rename pour afficher la boîte de dialogue Rename Resource.

Lorsque vous entrez un nouveau nom dans la boîte de dialogue Rename Resource et que vous cliquez sur OK, Resource Workshop vous demande si vous souhaitez créer un nouvel identificateur pour ce nom. Si vous cliquez sur OK, Resource Workshop affiche la boîte de dialogue New Identifieur.

Assurez-vous d'abord que la boîte liste File contient le nom de fichier identificateur correct, puis tapez une valeur entière appropriée dans la boîte texte Value et cliquez sur OK. Lors du prochain enregistrement de votre projet, la valeur et le nom d'identificateur seront enregistrés dans votre fichier .H, .PAS ou .INC.

A partir de la boîte de dialogue Identifiers

A partir de la boîte de dialogue Identifiers, vous pouvez ajouter un identificateur à votre fichier d'identificateurs avant de créer la ressource qui lui est associée :

1. Choisissez Resource | Identifiers pour afficher la boîte de dialogue Identifiers.
2. Cliquez sur le bouton New. Resource Workshop affiche la boîte de dialogue New Identifieur.
3. Utilisez la boîte liste File pour spécifier le fichier dans lequel l'identificateur sera stocké.
4. Tapez le nom de la ressource dans la boîte Name.
5. Tapez la valeur d'identificateur dans la boîte Value.
6. Cliquez sur OK.

Le nouveau nom de ressource apparaît maintenant dans la boîte liste Identifiers, et sa valeur est initialisée à "(unused)".

En plus de l'ajout d'identificateurs, vous pouvez utiliser la boîte de dialogue Identifiers pour éditer, supprimer ou lister les identificateurs, et pour lancer un éditeur de ressources. Choisissez Resource | Identifiers pour afficher la boîte de dialogue Identifiers.

**Edition
d'identificateurs**

Vous pouvez modifier la valeur d'un identificateur :

1. Choisissez Resource | Identifiers pour afficher la boîte de dialogue Identifiers.
2. Sélectionnez l'identificateur dont vous voulez modifier la valeur.
3. Cliquez sur le bouton Change. Resource Workshop affiche la boîte de dialogue Change Identifier Value.
4. Tapez la nouvelle valeur dans la boîte New Value et cliquez sur OK.

La nouvelle valeur d'identificateur sera écrite dans votre fichier .H, .PAS ou .INC au prochain choix de File | Save Project.

Vous pouvez également déplacer et renommer des ressources à partir de la boîte de dialogue Identifiers.

**Suppression
d'identificateurs**

Si un identificateur n'est pas utilisé dans votre projet, il est conseillé de le supprimer du fichier .H, .PAS ou .INC. Un identificateur peut devenir superflu dans l'un des cas suivants :

- Vous avez affecté un identificateur à une ressource, puis vous avez supprimé cette ressource.
- Vous avez ajouté un identificateur au projet et ne l'avez jamais utilisé.
- Vous avez renommé une ressource qui disposait déjà d'une valeur entière d'identificateur.

Pour supprimer un identificateur :

1. Choisissez Resource | Identifiers pour afficher la boîte de dialogue Identifiers.
2. Sélectionnez l'identificateur à supprimer.

Si l'identificateur sélectionné n'est pas associé à une ressource (si la ressource a été supprimée ou si l'identificateur n'a jamais été utilisé), la boîte Usage indique "(unused)".

Il est possible de supprimer un identificateur qui est toujours utilisé.

Cependant, si l'identificateur est toujours associé à une ressource, la boîte Usage met automatiquement en surbrillance le type et le nom de cette ressource.

3. Cliquez sur le bouton Delete.

Si l'identificateur est inutilisé, il est automatiquement supprimé. Aucune boîte de dialogue d'avertissement ne s'affiche.

Si l'identificateur est toujours utilisé, Resource Workshop affiche une boîte de dialogue d'avertissement contenant le message "#define [or Constant] is used. Delete anyway?". Pour supprimer l'identificateur, cliquez sur le bouton Yes. Si vous ne voulez pas supprimer l'identificateur, cliquez sur le bouton No.

4. Au prochain choix de File | Save Project, ResourceWorkshop met à jour le fichier identificateur, en retirant l'identificateur supprimé.

Liste des identificateurs

Pour obtenir la liste des identificateurs de votre projet :

1. Choisissez Resource | Identifiers pour afficher la boîte de dialogue Identifiers.
2. Effectuez votre choix entre All et Single File dans le groupe View. Si vous choisissez Single File, sélectionnez le fichier dont vous voulez voir les identificateurs à partir de la boîte liste File name du groupe View.
3. Faites défiler le contenu de la boîte liste Identifiers jusqu'aux identificateurs à visualiser. Lorsque vous mettez en surbrillance un identificateur dans la boîte liste, son nom et sa valeur entière apparaissent dans les boîtes Name et Value situées au-dessus de la boîte liste.

Lancement d'un éditeur de ressources

Vous pouvez utiliser la boîte de dialogue Identifiers pour activer un éditeur de ressources avec une ressource présélectionnée déjà chargée :

1. Faites défiler le contenu de la boîte liste Identifiers jusqu'à ce que la ressource désirée soit en surbrillance. Le nom et le type de la ressource apparaissent dans la boîte liste Usage.
2. Double-cliquez sur le nom et le type en surbrillance dans la boîte liste Usage. Resource Workshop lance l'éditeur approprié, avec cette ressource déjà chargée.

Préférences de configuration

Pour définir les préférences de configuration, choisissez File | Preferences. Pour plus d'information sur les sujets non abordés dans les paragraphes qui suivent, affichez l'aide en ligne en cliquant sur le bouton Help de la boîte de dialogue Preferences.

Undo Levels

Resource Workshop comporte une fonction Undo (défaire) et Redo (refaire) multi-niveaux, pour corriger les actions dans les éditeurs de ressources. Selon la quantité de mémoire disponible sur votre ordinateur, vous pouvez défaire ou refaire jusqu'à 99 actions. Le nombre de niveaux par défaut est 10.

Pour chaque annulation (défaire), appuyez sur *Alt+Retour arrière* ou choisissez Edit | Undo. Pour chaque rétablissement (refaire), appuyez sur *Maj+Alt+Retour arrière* ou choisissez Edit | Redo. Vous pouvez avancer et revenir pendant la session d'édition en fonction du nombre de niveaux défini dans l'option Undo Levels.



Le nombre de niveaux d'annulation peut être limité par la mémoire système disponible. Une ressource nécessitant beaucoup de mémoire, par exemple un gros graphique par points avec plusieurs remplissages denses, réduit le nombre de niveaux d'annulation disponibles.

Text Editor

Lorsque Resource Workshop charge un fichier projet, il compile toutes les ressources de ce fichier. Si le compilateur découvre une erreur, il interrompt la compilation, vous signale l'erreur et vous demande si vous souhaitez corriger le fichier avec l'éditeur de texte externe spécifié dans cette option.

L'éditeur de texte externe par défaut est l'éditeur Notepad de Windows ; si vous spécifiez un autre éditeur, il doit tourner sous Windows (tout éditeur DOS doté d'un fichier .PIF convient).

Multi-Save

Les préférences .RES et Executable contrôlent la façon dont un projet sera enregistré lorsque vous sélectionnez File | Save Project. Ces préférences ne sont activées que si un projet du compilateur de ressource (.RC ou .DLG) est ouvert, car elles ne s'appliquent qu'à un projet spécifique. Quels que soient les réglages de cette option, le projet est également enregistré dans son format d'origine. Par exemple, si le projet est un fichier .RC, outre les options de Multi-Save, les ressources du fichier sont toujours enregistrées en tant que scripts ressource en complément des éventuelles valeurs Multi-Save.

- *.RES*. Cette option compile les ressources du projet en cours et les enregistre au format *.RES* (format binaire).
- *Executable*. Cette option compile les ressources du projet en cours et les lie au fichier exécutable spécifié dans cette option (fichier *.EXE* ou *.DLL*).
- *Make Backups When Saving Files*. Si vous activez l'option Backups, Resource Workshop crée un jeu complémentaire de fichiers de sauvegarde chaque fois que vous enregistrez un projet. Les fichiers de sauvegarde comportent un tilde (~) comme premier caractère de l'extension de fichier. Par exemple, si vous enregistrez *MYPROJ.BMP*, le fichier de sauvegarde est appelé *MYPROJ.~BM*.

Target Windows Version

Ces boutons radio vous permettent d'adapter votre ressource à des versions spécifiques de Windows. Cependant, les fichiers *.RES* et *.EXE* adaptés pour Windows 3.1 ne sont pas compatibles avec Windows 3.0.

Si le fichier *.RES* est adapté à Windows 3.1, vous ne pouvez pas utiliser la version Windows 3.0 de *RC.EXE* pour lier vos ressources à l'application. Vous devez utiliser les outils Borland ou la version 3.1 du compilateur de ressources Microsoft.



Cette option est disponible uniquement si aucun projet n'est ouvert.

Comment travailler avec des fichiers binaires

Resource Workshop vous permet d'ouvrir des fichiers exécutables, des fichiers *.RES*, et des fichiers *DLL* en tant que projets, afin de personnaliser leurs interfaces utilisateur. Par exemple, vous pouvez traduire l'interface de votre application dans une autre langue.

Lorsque vous chargez l'un de ces fichiers, Resource Workshop décompile les ressources du fichier et vous les montre comme faisant partie d'un fichier *.RC* normal. Vos modifications achevées, Resource Workshop compile de nouveau les ressources en code binaire, et les stocke dans le fichier d'origine.

Tous les identificateurs de ressource des fichiers binaires doivent être des entiers.

Les ressources d'un fichier binaire décompilé n'étant pas stockées au format script ressource, vous ne pouvez pas affecter des noms aux identificateurs de ressource.

Cependant, vous pouvez enregistrer le projet en tant que fichier *.RC*. Les ressources peuvent alors être enregistrées en tant que scripts ressource, et donc recevoir des identificateurs.

Si vous personnalisez l'interface utilisateur d'un programme, et que vous avez accès uniquement au fichier exécutable ou au fichier DLL, vous pouvez enregistrer vos modifications dans un fichier .RC distinct, afin de les appliquer à la version suivante du programme. Les ressources de ce fichier .RC doivent avoir les mêmes identificateurs que leurs homologues de la nouvelle version, et doivent être compatibles avec la nouvelle version.

Lorsque vous enregistrez le projet en tant que fichier .RC, Resource Workshop n'enregistre pas automatiquement les ressources dans le fichier d'origine, sauf si vous spécifiez le nom de celui-ci sous forme d'option Multi-Save dans la boîte de dialogue Preferences.

Pour enregistrer un fichier binaire comme un projet et ajouter des identificateurs, procédez comme suit :

1. Choisissez File | Open Project et sélectionnez le fichier exécutable, .RES ou DLL dans la boîte de dialogue Open Project.
2. Choisissez File | Save File As. Dans la boîte de dialogue Save File As, sélectionnez RC Resource Script dans la boîte liste File Type. Entrez le nom du nouveau fichier .RC.

Lorsque vous cliquez sur OK pour enregistrer le fichier, Resource Workshop vous place automatiquement dans le fichier .RC.

3. Choisissez File | Preferences et entrez le nom du fichier binaire d'origine, en tant qu'option Multi-Save.
 - Si le fichier binaire d'origine était un fichier .RES, activez .RES et entrez le nom dans cette boîte texte.
 - Si le fichier binaire d'origine était un fichier exécutable, ou .DLL, activez Executable et entrez le nom dans cette boîte texte.
4. Choisissez File | Add to Project et spécifiez un fichier identificateur pour contenir les nouveaux identificateurs. Si le fichier que vous spécifiez n'existe pas, Resource Workshop le crée.
5. Modifiez les ressources et spécifiez des identificateurs aux endroits souhaités. Pour chaque nouvel identificateur, Resource Workshop vous demande si vous voulez l'enregistrer dans le fichier identificateur.
6. Lorsque vous quittez et enregistrez le fichier, Resource Workshop enregistre le fichier .RC et le fichier binaire. Si ce dernier est un fichier exécutable, ou DLL, les ressources modifiées lui sont liées (et sont immédiatement disponibles lorsque vous exécutez ce programme).

Vous devez conserver les valeurs entières en cours des identificateurs de ressource.

Comment créer des ressources 32 bits

Par défaut, Resource Workshop crée des ressources 16 bits si vous travaillez sous Windows en mode 16 bits et des ressources 32 bits si vous utilisez Windows NT, mais vous pouvez créer des ressources 32 bits dans le premier cas et des ressources 16 bits dans le deuxième cas.

Pour changer le type des ressources créées, fermez tous les projets ouverts et procédez comme suit :

1. Choisissez File | Preferences.
2. Cochez l'option Target Windows Version requise et cliquez sur OK.
3. Choisissez File | New project. Lorsque vous avez commencé à créer un projet, vous ne pouvez modifier le type de Target (par exemple, vous ne pouvez pas passer de Win32 à Windows 3.1).

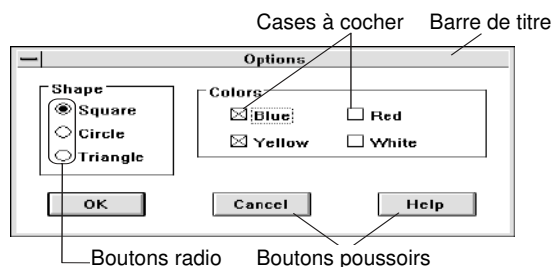
Si vous choisissez le type de Target Win32, vous pouvez également choisir une langue majeure et une langue mineure dans la boîte de dialogue File | Preferences (dans ce cas, la langue choisie doit être la langue qui sera utilisée sous NT ; sinon, l'affichage des ressources – et plus particulièrement des menus – risque d'être incorrect).

Création de boîtes de dialogue

Les *boîtes de dialogue* sont des fenêtres surgissantes qui permettent de choisir diverses options d'un programme (fichiers à ouvrir, couleurs à afficher, texte à rechercher, etc.).

Une boîte de dialogue contient généralement un certain nombre de *contrôles* (boutons, boîtes texte et barres de défilement) permettant de spécifier les informations requises, ou d'afficher un texte et des graphiques statiques dans la boîte de dialogue.

Figure 15.1
Exemple de boîte de dialogue



Lancement de l'éditeur Dialog

Avec l'éditeur Dialog, vous pouvez créer des boîtes de dialogue, ou éditer des boîtes de dialogue existantes. Vous lancerez l'éditeur Dialog différemment selon la tâche à réaliser.

Création d'une boîte de dialogue

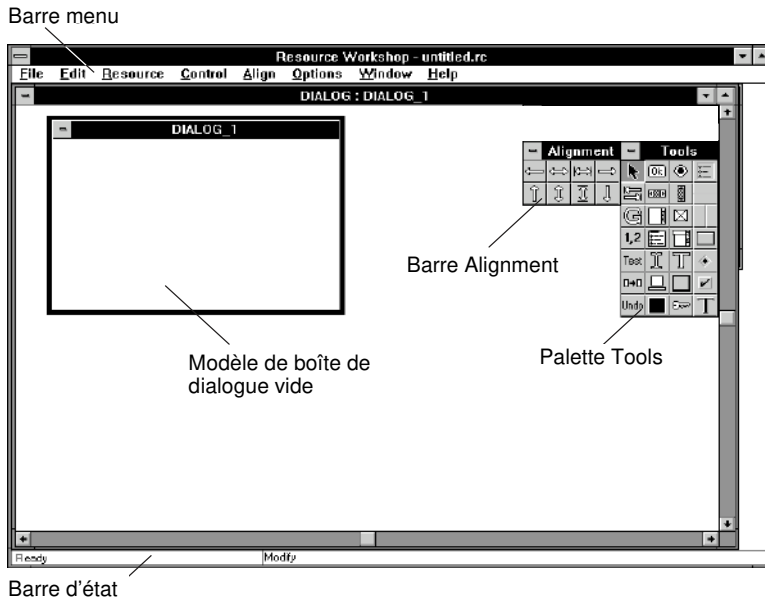
Le chapitre 14 décrit la procédure d'ouverture d'un projet.

Pour lancer l'éditeur Dialog et créer une boîte de dialogue :

1. Choisissez File | New Project pour lancer un nouveau projet, ou File | Open Project pour charger un projet existant.
2. Choisissez Resource | New. La boîte de dialogue New Resource s'affiche.
3. Dans la boîte liste Resource Type, double-cliquez sur DIALOG.

4. Une autre boîte de dialogue s'affiche pour vous permettre de spécifier un modèle (type) de boîte de dialogue ; choisissez un type et cliquez sur OK.
5. Vous êtes maintenant dans l'éditeur Dialog, qui affiche le gabarit de boîte de dialogue choisi. La figure 15.2 illustre les composants de l'éditeur Dialog.

Figure 15.2
Editeur Dialog avec
boîte de dialogue
vide



Édition d'une boîte de dialogue existante

Pour éditer une boîte de dialogue qui existe déjà dans un fichier projet :

1. Ouvrez le projet qui contient la boîte de dialogue à éditer.
2. Retrouvez la boîte de dialogue à éditer dans la fenêtre projet et double-cliquez sur son nom.

L'éditeur Dialog s'ouvre. Il ressemble à la figure 15-2, à ceci près que la boîte de dialogue sélectionnée apparaît à la place de la boîte de dialogue vide.

Utilisation de l'éditeur Dialog

Comme illustré à la figure 15.2, l'éditeur Dialog vous présente un modèle vide lorsque vous créez une boîte de dialogue. Vous pouvez choisir le type de fenêtre de votre boîte de dialogue, un libellé, les polices à afficher, etc. Vous pouvez aussi ajouter, modifier, grouper, réordonner, déplacer, redimensionner ou supprimer des contrôles dialogue, afin que votre boîte de dialogue fonctionne comme vous le souhaitez.

Sélection d'une boîte de dialogue

Pour modifier la fenêtre de la boîte de dialogue, ou pour définir les attributs d'une boîte de dialogue, vous devez d'abord la sélectionner. Pour cela, cliquez sur la barre de titre ou sur le bord externe de la boîte de dialogue requise.

Lorsqu'une boîte de dialogue est sélectionnée, Resource Workshop l'entoure d'un *cadre de sélection*. Vous pouvez alors :

- Déplacer la boîte de dialogue à l'aide de la souris ou des touches fléchées.
- Redimensionner la boîte de dialogue selon l'une des procédures suivantes :
 - Faites glisser le bord ou le coin approprié.
 - Sélectionnez la boîte de dialogue en cliquant sur sa barre de titre, puis en choisissant Align | Size pour afficher la boîte de dialogue Size Dialog. Entrez les valeurs de largeur (CX) et de hauteur (CY) en unités dialogue. Dans une unité dialogue, Y correspond à 1/8 de la hauteur de police, et X correspond à 1/4 de la largeur de police. (Ces valeurs s'appliquent aux dimensions extérieures de la boîte de dialogue.)

Définition des attributs d'une boîte de dialogue

Au moyen de la boîte de dialogue Window Style, vous pouvez définir les attributs suivants de votre nouvelle boîte de dialogue :

- Libellé, classe et menu
- Type de fenêtre
- Style de cadre
- Style de dialogue
- Police

Pour afficher la boîte de dialogue Window Style, sélectionnez la boîte de dialogue dont vous souhaitez définir les attributs et appuyez sur *Entrée*, ou double-cliquez avec l'outil de sélection (le pointeur de la souris se change en flèche) sur la barre de titre de la barre de dialogue.

Ajout d'un libellé

Pour modifier un libellé, vous pouvez également utiliser la boîte Propriétés (choisissez Options | Show Properties)

Pour ajouter un libellé :

1. Double-cliquez sur la barre de titre de la boîte de dialogue pour ouvrir la boîte de dialogue Window Style.
2. Tapez le nouveau nom de votre boîte de dialogue dans la boîte de saisie Caption.
3. Au-dessous de Frame Style, veillez à ce que Caption (la valeur par défaut) soit sélectionnée.
4. Cliquez sur OK.

La boîte de dialogue Window Style permet de choisir l'un des types suivants pour votre boîte de dialogue :

- *Popup*. Fenêtre surgissante (valeur par défaut, dans la mesure où la plupart des boîtes de dialogue sont de ce type).
- *Child*. Enfant de la fenêtre active.
- *Overlapped*. Fenêtre surgissante qui peut être recouverte (intégralement ou partiellement) par une autre. Pour être de ce type, une boîte de dialogue doit être la fenêtre principale de l'application.

Sélection d'un style de cadre

Le style du cadre de la boîte de dialogue détermine l'aspect du cadre et de la barre titre en haut de la boîte de dialogue.

Choisissez un style de boîte de dialogue à partir de la boîte de dialogue Window Style. Les options disponibles sont les suivantes :

- *Dialog Frame*. Une double bordure, sans barre de titre.
- *Border*. Une seule bordure fine, sans barre de titre.
- *Caption*. Une seule bordure fine et une barre de titre où un libellé pourra s'afficher (par défaut).
- *No Border*. Pas de bordure ni de barre de titre.

Spécification du style de dialogue

Le style de dialogue détermine l'aspect de la boîte de dialogue et son type d'utilisation. Les options disponibles sont les suivantes :

- *System Menu*. Intègre une case de menu Système à gauche de la barre de titre. Également appelé menu Control, le menu Système s'affiche uniquement si vous avez choisi Caption comme style de cadre pour votre boîte de dialogue (Caption et System Menu sont deux valeurs par défaut). Si la boîte de dialogue est définie comme une fenêtre enfant, vous obtiendrez une case Fermeture au lieu d'un menu Control.

- *Thick Frame*. Traçage d'un cadre épais autour de la boîte de dialogue. Cette option définit ce que l'utilisateur verra lorsque la boîte de dialogue va s'afficher dans une application. Si vous souhaitez une boîte de dialogue redimensionnable, utilisez cette option.
Ne confondez pas cette option avec l'option *Thick Frame* de la commande *Preferences* de l'éditeur *Dialog*, qui définit l'aspect de la boîte de dialogue si vous la sélectionnez dans l'éditeur *Dialog*.
- *Vertical Scroll*. Ajout d'une barre de défilement verticale dans le cadre de la boîte de dialogue.
- *Horizontal Scroll*. Ajout d'une barre de défilement horizontale dans le cadre de la boîte de dialogue.
- *Minimize Box*. Ajout d'un bouton Réduction à droite de la barre de titre. Ce bouton s'affiche uniquement si vous avez sélectionné *Caption* comme style de cadre pour votre boîte de dialogue.
- *Maximize Box*. Ajout d'un bouton *Maximize* à droite de la barre de titre. Ce bouton s'affiche uniquement si vous avez sélectionné l'option *Caption* comme style de cadre pour votre boîte de dialogue.
- *Absolute Align*. Les coordonnées de la boîte de dialogue deviennent proportionnelles à l'écran plutôt qu'à la fenêtre parent.
- *System Modal*. La boîte de dialogue devient modale par rapport au système, ce qui signifie que l'utilisateur ne peut passer à une autre action tant que la boîte de dialogue est affichée.
- *Local Edit*. Alloue tous les contrôles d'édition de texte inclus dans cette boîte de dialogue au segment de mémoire local de l'application.
Choisissez *Local Edit* si votre application doit utiliser les messages `EM_SETHANDLE` et `EM_GETHANDLE`.
- *Modal Frame*. Encadre la fenêtre avec une combinaison de styles d'encadrement et de libellé (valeur par défaut). Choisissez *Modal Frame* si vous souhaitez que les utilisateurs puissent déplacer la boîte de dialogue.
- *No Idle Messages*. Supprime la transmission des messages `WM_ENTERIDLE` à la fenêtre principale de l'application. Cette option est possible uniquement avec les boîtes de dialogue modales.
- *Clip Children*. Protège la zone client des fenêtres enfant.
- *Clip Siblings*. Protège les soeurs de la fenêtre. L'affichage est limité à cette fenêtre. Cette option n'est pas nécessaire pour les fenêtres surgissantes, mais elle peut être utile pour les fenêtres de dialogue enfant.

- *Visible*. Rend visible une boîte de dialogue non modale avant le retour de `CreateDialog`. Cette option n'a aucun effet sur les boîtes de dialogue modales (le type de boîte de dialogue le plus courant). Par défaut, cette option est désactivée (`NOT_WS_VISIBLE`).

Sélection de police

Par défaut, les boîtes de dialogue utilisent la police MS Sans Serif 8 points pour le texte, mais vous pouvez choisir une police et une taille différentes :

1. Ouvrez la boîte de dialogue `Window Style`.
2. Cliquez sur le bouton `Fonts` pour ouvrir la boîte de dialogue `Select Font`.
3. Utilisez la boîte de dialogue `Select Font` pour choisir une police et une taille de caractère pour le texte de votre boîte de dialogue. Les caractères de la boîte `Text` située en bas de la boîte de dialogue illustrent la police et la taille de caractère actuellement sélectionnée. (Notez que Windows accepte uniquement du texte en gras dans les boîtes de dialogue.)

Comment inclure un menu

Une boîte de dialogue peut inclure un menu puisqu'elle est en fait une fenêtre. Par exemple, certaines applications utilisent une boîte de dialogue pour leur fenêtre principale. Dans ce cas, la boîte de dialogue nécessite un menu.

Pour inclure un menu dans votre boîte de dialogue :

1. Créez le menu comme une ressource séparée et vérifiez qu'il fait partie du projet (voir le chapitre 16). Notez le nom de la ressource ou l'identificateur numérique qui désigne le menu.
2. Ouvrez l'éditeur `Dialog` et chargez la boîte de dialogue dans laquelle vous voulez ajouter le menu.
3. Ouvrez la boîte de dialogue `Window Style`.
4. Dans la boîte de saisie `Menu`, tapez le nom de ressource ou l'identificateur numérique du menu.



Dans les modes d'affichage standard, l'éditeur `Dialog` n'affiche pas le menu. Lorsque vous testez la boîte de dialogue (voir la section "Comment tester une boîte de dialogue" de ce chapitre), `Resource Workshop` affiche le menu générique "Menu" qui ne contient qu'un seul élément de menu : "Item".

Comment affecter une classe personnalisée à une boîte de dialogue.

Si vous êtes un programmeur Windows confirmé, vous souhaitez peut-être affecter une classe personnalisée à une boîte de dialogue, afin de gérer les messages de la boîte de dialogue avec vos propres procédures de fenêtres (au lieu des procédures Windows). Vous pouvez aussi affecter une classe personnalisée à une boîte de dialogue pour la transformer en boîte de dialogue de style Borland.

Pour affecter une classe personnalisée à une boîte de dialogue :

1. Ouvrez la boîte de dialogue Window Style.
2. Tapez le nom de la classe dans la boîte de saisie Class.

Définition de la position de la boîte de dialogue

Si votre boîte de dialogue utilise le style `WS_OVERLAPPED`, vous pouvez laisser Windows la positionner à l'écran. Cette option est généralement utilisée pour les boîtes de dialogue qui jouent le rôle de fenêtres principale. Pour donner le contrôle de la position de la boîte de dialogue à Windows :

1. Sélectionnez le cadre de la boîte de dialogue en cliquant sur son bord ou sur la barre de titre.
2. Choisissez Align | Size.
3. Dans la boîte de dialogue Size Dialog, cliquez sur le bouton radio Set By Windows.
Resource Workshop noircit la valeur de coordonnée X dans la boîte de dialogue, spécifiant ainsi que Windows a le contrôle de la boîte de dialogue.
4. Cliquez sur OK pour quitter la boîte de dialogue Size Controls.

Comment travailler avec les contrôles

Les contrôles sont des composants individuels des boîtes de dialogue. Ils permettent l'échange d'informations entre l'utilisateur et l'application, et appartiennent aux catégories suivantes :

- Boutons
- Barres de défilement
- Boîtes liste
- Boîtes de saisie
- Contrôles statiques
- Boîtes à options combinées
- Contrôles personnalisés

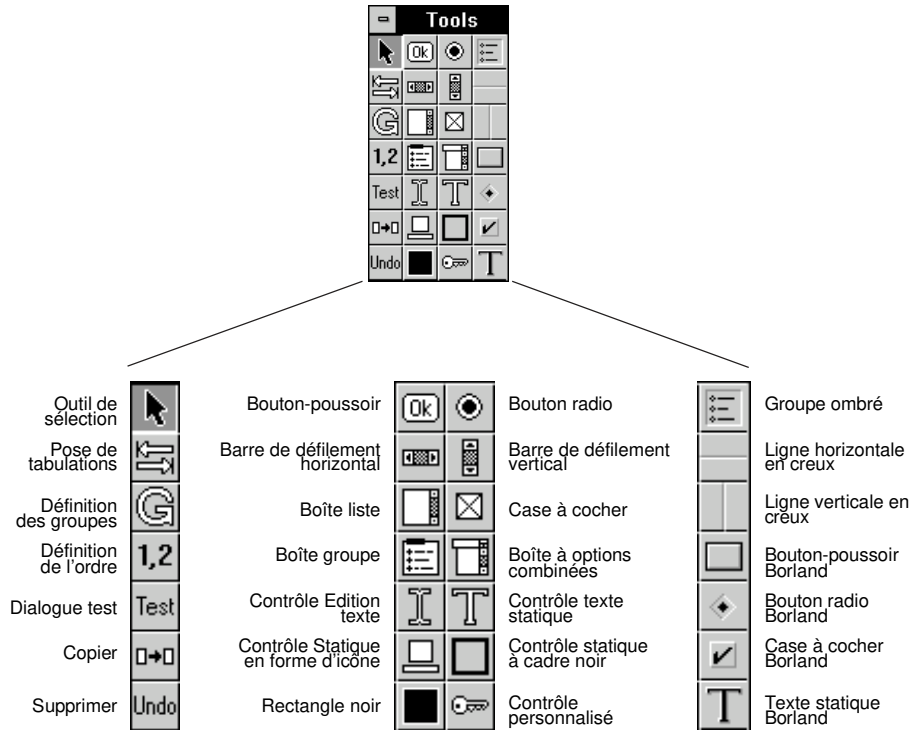
Chaque contrôle utilisé provient d'un des cinq groupes de famille suivants :

- Contrôles Windows standard tels que les boutons-poussoirs, les cases à cocher, les boîtes liste et les boutons radio. Les icônes des contrôles Windows standard apparaissent dans les deuxième et troisième colonnes de la palette Tools. Ces contrôles sont toujours disponibles.
- Borland Windows Custom Controls (BWCC), toujours disponibles. Les Contrôles Personnalisés Windows de Borland (y compris les boutons radio, les cases à cocher et les boutons-poussoirs) offrent des améliorations fonctionnelles et visuelles par rapport aux contrôles Windows standard. Les icônes de ces contrôles apparaissent dans la quatrième colonne de la palette Tools et sont décrits dans l'annexe B.
- Contrôles personnalisés dont la classe est reconnue par Resource Workshop. Ces contrôles sont stockés dans un fichier DLL qui inclut la fonction ListClasses. Lorsque le fichier DLL est installé, les icônes de ces contrôles apparaissent dans une ou plusieurs colonnes supplémentaires de la palette Tools, en commençant à droite des icônes BWCC. Ce type de contrôle est décrit à la section "Contrôles personnalisés" de ce chapitre, et dans le fichier CUSTCNTL.RW.
- Contrôles personnalisés dont la classe est reconnue par l'éditeur de dialogues du SDK de Windows. Ces contrôles sont stockés dans un fichier DLL qui inclut leurs bitmaps mais non la fonction ListClasses. Ils ne sont pas représentés dans la palette Tools, mais leur nom apparaît dans la liste déroulante de la boîte de dialogue New Custom Control (voir la section "Contrôles personnalisés" de ce chapitre). Lorsque vous ajoutez un de ces contrôles dans une boîte de dialogue, son bitmap apparaît à l'écran si Resource Workshop est en mode d'affichage WYSIWYG.
- Contrôles personnalisés dont la classe n'est pas reconnue par Resource Workshop ni par l'éditeur de dialogues du SDK de Windows. Resource Workshop ajoute leurs noms dans la liste déroulante de la boîte de dialogue New Custom Control, mais ils apparaissent à l'écran (en mode WYSIWYG) sous forme de rectangles gris.

La colonne gauche de la palette Tools contient des outils qui définissent le mode d'exploitation en cours de l'éditeur Dialog.

Les autres colonnes contiennent des icônes pour les contrôles Windows standard, pour les contrôles BWCC et pour tous les contrôles personnalisés chargés et reconnus par Resource Workshop.

Figure 15.3
PaLETTE Tools



Voici une brève description des contrôles Windows standard dont les icônes apparaissent dans les deuxième et troisième colonnes de la palette Tools.



Bouton-poussoir Bouton rectangulaire sur lequel l'utilisateur "appuie" pour sélectionner une action. Puisque les boutons-poussoirs contiennent toujours un texte, vous devez indiquer un titre pour chacun d'entre eux.



Bouton radio Bouton circulaire avec un texte à sa gauche ou à sa droite. Lorsque le bouton est sélectionné, un point apparaît dans le cercle. Les boutons radio s'utilisent en groupe pour représenter une série d'options qui s'excluent mutuellement.



Barre de défilement horizontal Rectangle horizontal avec flèche directionnelle à chaque extrémité.



Barre de défilement vertical

Rectangle vertical avec flèche directionnelle à chaque extrémité.



Boîte liste

Rectangle contenant généralement une liste de chaînes de texte. Si vous utilisez le style Owner-draw, la boîte liste peut ainsi contenir une représentation visuelle d'une liste de données. En général, l'utilisateur peut faire défiler le contenu de la liste et choisir un ou plusieurs éléments. Les boîtes listes se rencontrent fréquemment dans les boîtes de dialogue Ouvrir fichier.



Case à cocher

Bouton rectangulaire avec un texte à sa gauche ou à sa droite. Si vous sélectionnez une case à cocher, elle est marquée d'un X. Lorsqu'elle n'est pas sélectionnée, le carré est vide. Les cases à cocher s'utilisent en général pour représenter l'état booléen (activé/désactivé) d'une option.



Boîte groupe

Boîte rectangulaire utilisée pour regrouper visuellement plusieurs contrôles. Vous pouvez inclure un libellé à afficher dans l'angle supérieur gauche de la boîte groupe.



Boîte à options combinées

Combinaison d'une boîte liste et d'un contrôle édition texte, ou d'une boîte liste et d'un contrôle statique.



Contrôle édition texte

Rectangle dans lequel l'utilisateur peut entrer du texte à partir du clavier.



Contrôle statique texte

Texte qui s'affiche dans la boîte de dialogue.



Contrôle statique en forme d'icône

Icône.



Contrôle statique à cadre noir

Cadre vide rectangulaire de la couleur du cadre de la fenêtre active.



Rectangle noir Icône contrôle statique qui apparaît sous la forme d'un rectangle de la couleur du cadre de la fenêtre active.



Contrôle personnalisé Contrôle dont la classe est différente des types BWCC ou Windows standard, non reconnu par Resource Workshop.

Sélection de contrôles

Un certain nombre d'options d'édition Resource Workshop nécessitent qu'un ou plusieurs contrôles soient *sélectionnés*. Par exemple, vous devez sélectionner un contrôle avant de pouvoir modifier sa taille, et vous devez sélectionner au moins deux contrôles avant de pouvoir les aligner l'un par rapport à l'autre. Tout contrôle sélectionné est entouré par un cadre de sélection.



Pour basculer en *mode sélection* afin de pouvoir sélectionner les contrôles, vous devez d'abord cliquer sur l'outil Sélectionneur (le pointeur de la souris se change en flèche).

Pour sélectionner un seul contrôle, placez le Sélectionneur à l'intérieur du contrôle ou sur son bord, selon l'état courant de la case à cocher Select Near Border dans la boîte de dialogue Options | Preferences et cliquez.

Pour sélectionner plusieurs contrôles, vous disposez de deux options :

- Faites glisser un *rectangle de sélection* autour des contrôles à sélectionner, en suivant les étapes ci-dessous :
 1. Choisissez l'outil Sélectionneur.
 2. Cliquez sur un des contrôles pour garantir que le contrôle (et non la boîte de dialogue entière) est sélectionné.
 3. Faites glisser un rectangle de sélection. Selon l'état courant de la case à cocher Selection Rectangle Surrounds de la boîte de dialogue Options | Preferences, le rectangle de sélection doit entourer complètement les contrôles, ou simplement les toucher.

Resource Workshop place un cadre de sélection autour des contrôles sélectionnés. Pour déplacer les contrôles sélectionnés, faites glisser le cadre de sélection.

- Vous pouvez effectuer un *Maj-clic* pour sélectionner certains contrôles, et non l'ensemble des contrôles comme le ferait la méthode précédente. Pour effectuer un *Maj-clic*, cliquez sur le premier contrôle, puis maintenez enfoncée la touche *Maj* pendant que vous cliquez sur les autres contrôles à sélectionner.

Pour ajouter un ou plusieurs contrôles à un groupe déjà sélectionné, ou supprimer un ou plusieurs contrôles de ce groupe, effectuez un *Maj*-clic sur les contrôles à ajouter ou à supprimer.

Pour sélectionner tous les contrôles d'une boîte de dialogue, choisissez *Edit | Select All*. Resource Workshop place un cadre autour de chaque contrôle, et un cadre de sélection autour du groupe ainsi constitué. Notez que *Edit | Select All* ne sélectionne pas le cadre de la fenêtre contenant la boîte de dialogue.

Vous pouvez aussi sélectionner les contrôles au moyen de la touche *Tab* :

Pour afficher la séquence, cliquez sur l'outil *Set Order*.

- Si un seul contrôle est sélectionné, l'appui sur *Tab* déplace la sélection sur le prochain contrôle dans la séquence (ordre dans lequel les contrôles ont été ajoutés). L'appui sur *Maj-Tab* ramène la sélection sur le contrôle précédent dans la séquence.

Dans tous les cas, si le contrôle sélectionné est le dernier de la séquence, *Tab* déplace la sélection sur le cadre de la boîte de dialogue.

- Si vous avez sélectionné un groupe de contrôles à l'aide du rectangle de sélection, *Tab* déplace la sélection sur le prochain contrôle dans la séquence, après le contrôle le plus élevé du groupe.

Par exemple, si le rectangle de sélection contient les contrôles 4, 5, 9 et 10, *Tab* déplace la sélection sur le contrôle 11 (s'il existe).

- Si vous avez sélectionné un groupe de contrôles à l'aide de *Maj*-clic, *Tab* déplace la sélection sur le premier contrôle de la séquence qui suit le dernier contrôle sélectionné.

Par exemple, si les contrôles sélectionnés dans un groupe sont 9, 10, 4 et 5 dans cet ordre, *Tab* déplace la sélection sur le contrôle 6.

Ajout de contrôles

La manière la plus simple d'ajouter un nouveau contrôle à votre boîte de dialogue est la suivante :

1. Cliquez sur le contrôle souhaité dans la palette *Tools*. Le pointeur se transforme pour indiquer le type de contrôle que vous allez installer.
2. Cliquez à l'endroit où vous souhaitez positionner le contrôle dans la boîte de dialogue.

Si vous sélectionnez un contrôle dans la palette *Tools* mais que vous souhaitez modifier son emplacement, choisissez l'outil *Sélecteur*. Le pointeur redevient une flèche, ce qui vous permet de sélectionner les contrôles dans votre boîte de dialogue. L'appui sur *Echap* avant de placer un contrôle vous renvoie au mode de sélection.

Vous pouvez aussi utiliser le menu Control de l'éditeur Dialog (dans la barre menu, et non la bordure de fenêtre) pour ajouter des contrôles à votre boîte de dialogue :

1. Utilisez la souris ou appuyez sur *Alt+C* pour ouvrir le menu Control.
2. Choisissez le type de contrôle à ajouter dans votre boîte de dialogue.
3. Cliquez sur la boîte de dialogue à l'endroit où vous souhaitez placer le contrôle.

Ajout de plusieurs copies d'un contrôle

Vous pouvez également placer plusieurs copies identiques d'un contrôle en rangées ou colonnes. Vous souhaitez, par exemple, placer deux colonnes de quatre cases à cocher dans votre boîte de dialogue. Vous pourriez positionner chaque case à cocher séparément, mais Resource Workshop vous propose une solution plus simple grâce à la commande EditDuplicate.

Pour placer plusieurs copies identiques d'un contrôle en rangées ou en colonnes :



1. Dans votre boîte de dialogue, sélectionnez le contrôle à copier.
2. Cliquez sur l'outil Duplicate ou choisissez EditDuplicate. La boîte de dialogue Duplicate Control s'affiche.
3. Indiquez le nombre de rangées et colonnes souhaité, ainsi que l'espacement d'unités dialogue entre ces rangées et colonnes.
Si vous souhaitez, par exemple, placer huit cases à cocher en deux colonnes, indiquez quatre rangées et deux colonnes.
4. Choisissez OK. Les copies du contrôle s'affichent dans la boîte de dialogue, alignées à intervalles réguliers.



Lorsqu'un seul contrôle est sélectionné, l'outil Duplicate a le même effet que Edit Duplicate. Avec plusieurs contrôles sélectionnés, cet outil a le même effet que Align | Array (voir la section "Comment disposer les contrôles en colonnes et rangées" de ce chapitre).

Propriétés des contrôles (contrôles .VBX)

Pour afficher les propriétés des contrôles que vous ajoutez dans les boîtes de dialogue, choisissez Options | Show Properties. Lorsque Show Properties est activée, la boîte de dialogue Properties s'affiche (elle contient les propriétés du contrôle sélectionné).

Les contrôles .VBX peuvent être modifiés uniquement à partir de la boîte de dialogue Properties.

Si la boîte de dialogue Propriétés n'est pas affichée, vous pouvez double-cliquer sur le contrôle .VBX requis pour en afficher directement les propriétés.

Déplacement des contrôles

Vous pouvez déplacer un contrôle sélectionné en le faisant glisser à partir de n'importe quelle position à l'intérieur de son cadre de sélection. Vous pouvez aussi déplacer des contrôles groupés de la même façon. Ils se déplacent ensemble, en conservant leur position relative entre eux.

Pour déplacer un contrôle à l'aide du clavier, appuyez sur *Tab* pour sélectionner le contrôle. Utilisez ensuite les touches fléchées pour déplacer le contrôle et appuyez sur *Entrée* pour valider. Si vous changez d'avis, appuyez sur *Echap* au lieu de *Entrée*.

Redimensionnement des contrôles

Vous pouvez modifier la taille d'un contrôle sélectionné en faisant glisser le bord ou l'angle approprié. Vous pouvez utiliser à la fois le clavier et la souris pour affiner la taille :

1. Sélectionnez le contrôle et déplacez le curseur de la souris sur la partie du bord appropriée.
2. Lorsque le curseur devient une flèche double, maintenez enfoncé le bouton gauche de la souris.
3. Appuyez sur la touche fléchée appropriée pour déplacer le curseur de la souris et le cadre de sélection. Chaque appui sur la flèche déplace le curseur d'une seule unité dialogue.

Les unités dialogue sont définies à la section "Sélection d'une boîte de dialogue".

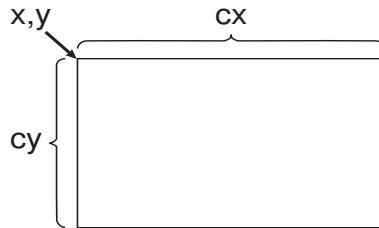
Placement et dimensionnement des contrôles en une seule opération

A l'aide de la boîte de dialogue Size Controls, vous pouvez spécifier simultanément la position et la taille d'un contrôle.

1. Sélectionnez le contrôle souhaité.
2. Choisissez Align | Size ou maintenez la touche *Alt* enfoncée tout en double-cliquant. La boîte de dialogue Size Controls s'affiche.
3. Pour définir la position de l'angle supérieur gauche du contrôle, spécifiez ses coordonnées X et Y en unités dialogue. Les coordonnées 0,0 placent le contrôle dans l'angle supérieur gauche de la fenêtre boîte de dialogue, directement sous la barre de titre.

Pour définir la hauteur et la largeur du contrôle, spécifiez ses valeurs CX et CY en unités dialogue (voir la figure suivante).

Figure 15.4
Coordonnées d'une
boîte de dialogue



Alignement des contrôles avec une grille

Vous pouvez afficher une grille sur votre boîte de dialogue et l'utiliser pour aligner vos contrôles :

1. Choisissez Align | Grid. La boîte de dialogue Set Grid Attributes s'affiche.
2. Spécifiez la largeur et la hauteur d'une cellule de grille (en unités dialogue).
3. Sélectionnez le type de grille. Voici les options disponibles :
 - *Absolute*. Déplace le contrôle jusqu'à la ligne de grille la plus proche.
 - *Relative*. Déplace le contrôle uniquement par incréments de largeur de grille (horizontalement) et de hauteur de grille (verticalement). Ainsi, si le contrôle n'a pas été placé initialement sur une ligne de la grille, vous ne pourrez pas le déplacer jusque sur une ligne lorsque cette option est sélectionnée.

Exemple : avec une grille réglée à 4x4 et un contrôle placé en (1,1), lorsque vous déplacez le contrôle, il se positionnera quatre unités plus loin dans chacun des axes. Les coordonnées possibles seraient (5,5), (5,9), (9,5), etc.

4. Cochez l'option Show Grid et choisissez OK.

Édition de contrôles

Pour modifier un contrôle dans votre boîte de dialogue, double-cliquez sur le contrôle pour afficher sa boîte de dialogue Style. Les options de cette boîte de dialogue varient selon le type de contrôle avec lequel vous travaillez. Par exemple, double-cliquer sur un bouton amène la boîte de dialogue Button Style.

Avec le clavier, utilisez *Tab* pour sélectionner le contrôle à éditer. Appuyez sur *Entrée* pour afficher la boîte de dialogue Style du contrôle sélectionné.

Bien que chaque type de contrôle dispose de sa propre boîte de dialogue Style, ces boîtes de dialogue ont de nombreuses options en commun (voir tableau ci-après).

Tableau 15.1
Options communes
aux boîtes de
dialogue Style

Option	Description
Caption	<p>Vous permet d'entrer le libellé qui s'affichera avec le contrôle. Différents types de contrôles affichent des libellés dans différentes zones. Par exemple, dans une boîte groupe, le libellé s'affiche en haut à gauche. Sur un bouton-poussoir, le libellé s'affiche à l'intérieur du bouton.</p> <p>Certains contrôles n'affichent pas de libellé. Par exemple, une boîte liste n'affiche pas le texte spécifié dans son libellé.</p> <p>A droite de l'endroit où vous saisissez le libellé, vérifiez l'option Text ou Number. Choisissez Text si vous voulez entourer le libellé avec des guillemets dans le code source du fichier dialogue ou du fichier .RC. Sélectionnez Number si vous ne voulez pas de guillemets.</p>
Control ID	<p>Permet de désigner un identificateur unique pour le contrôle. Les identificateurs de contrôle peuvent être un nombre entier ou une expression entière. Tapez l'identificateur de contrôle que vous souhaitez affecter à ce contrôle.</p> <p>Par convention, les contrôles statiques qui ne sont pas modifiés au moment de l'exécution reçoivent l'identificateur de contrôle -1.</p> <p>Si vous tapez un identificateur de contrôle alphanumérique, Resource Workshop vérifie si un identificateur existe en tant que <i>#define</i> ou <i>déclaration de constante</i>. Resource Workshop vous demande si vous souhaitez créer un identificateur. Voir le chapitre 14 pour plus d'informations sur les identificateurs.</p>
Scroll Bar	<p>Vous permet de choisir si vous souhaitez associer des barres de défilement verticales ou horizontales à votre contrôle.</p>

Certains attributs sont communs à la plupart des contrôles :

Tableau 15.2
Attributs de contrôle

Attribut	Description
Tab Stop	Permet à l'utilisateur d'appuyer sur <i>Tab</i> pour accéder au contrôle.
Group	Identifie le premier contrôle d'un groupe. Voir la section "Regroupement de contrôles liés" de ce chapitre pour plus de détails sur le regroupement des contrôles et l'accès aux contrôles.
Visible	Détermine si le contrôle est visible lors de l'affichage de la boîte de dialogue. Par défaut, cette option est activée (WS_VISIBLE). Si ce n'est pas le cas (NOT_WS_VISIBLE), le contrôle n'apparaît pas. L'application peut appeler la fonction ShowWindow au moment de l'exécution pour afficher le contrôle.
Disabled	Estompe (affiche en grisé) le contrôle pour indiquer qu'il ne répondra pas à l'entrée utilisateur.
Border	Trace une bordure autour du contrôle.

Chaque type de boîte de dialogue Style comporte des options spécifiques à un type de contrôle particulier. Ces options sont répertoriées à partir de la section "Contrôles boutons" de ce chapitre.

Comment ajouter des libellés aux contrôles

Pour ajouter un libellé à un contrôle, vous pouvez utiliser la boîte de dialogue Style, mais aussi la boîte de dialogue Properties ; sélectionnez le contrôle à compléter et suivez l'une des méthodes ci-dessous :

- Choisissez Options | Show properties et tapez un libellé dans la boîte de saisie supérieure.
- Double-cliquez sur le contrôle (dans la boîte de dialogue) et tapez son libellé dans la boîte de saisie Caption.

Modification d'une classe de contrôle

Si vous travaillez avec des contrôles personnalisés, la boîte de dialogue Generic Control Style vous sera utile. Affichez-la en maintenant la touche *Ctrl* enfoncée et en double-cliquant sur votre contrôle. Vous pouvez aussi utiliser *Tab* pour sélectionner le contrôle, puis maintenir la touche *Ctrl* enfoncée et appuyer sur *Entrée*.

Pour plus d'informations sur les contrôles personnalisés, voir la section "Contrôles personnalisés" de ce chapitre.

Dans la boîte de dialogue Generic Control Style, vous pouvez modifier la classe d'un contrôle. Vous pouvez aussi spécifier un libellé, un identificateur de contrôle et un style. Si vous tapez d'autres informations à côté de Info, la boîte de dialogue ne sera pas compatible avec le compilateur de ressources (Resource Compiler) Microsoft.

Comment définir des contrôles comme taquets de tabulation

Les outils Tab Set, Set Groups et Set Order affectent l'interaction clavier de l'utilisateur avec la boîte de dialogue.



Avec le clavier, les utilisateurs appuient généralement sur la touche *Tab* pour déplacer le pointeur d'un contrôle (ou groupe de contrôles) à un autre. Certains types de contrôles sont automatiquement définis comme taquets de tabulation lors de leur ajout dans une boîte de dialogue. En définissant l'état "taquet de tabulation" d'un contrôle, vous pouvez gérer les déplacements de l'utilisateur à l'intérieur de la boîte de dialogue au moyen de la touche *Tab*.

Deux méthodes permettent de modifier les taquets de tabulation :

- Utiliser l'outil Tab Set ou la commande Set Tabs.
- Utiliser la boîte de dialogue Style.

Pour définir ou modifier des tabulations, vous pouvez utiliser l'outil Tab Set ou la commande Set Tabs :

1. Cliquez sur l'outil Tab Set ou choisissez Options | Set Tabs. Le curseur devient l'icône Tab Set. Resource Workshop entoure tout contrôle actuellement défini comme taquet de tabulation par un cadre ombré.
2. Pour définir un taquet de tabulation, cliquez sur tout contrôle non entouré par un cadre ombré.
Pour annuler un taquet de tabulation, cliquez sur un contrôle entouré par un cadre ombré.
3. Lorsque vous modifiez les taquets de tabulation, cliquez sur le Sélecteur. Vous pourrez ainsi à nouveau éditer votre boîte de dialogue.

Vous pouvez aussi utiliser une boîte de dialogue Style pour modifier un taquet de tabulation :

1. Ouvrez la boîte de dialogue Style pour le contrôle (double-cliquez sur le contrôle, ou sélectionnez-le et appuyez sur *Entrée*).
2. Activez Tab Stop (sous Attributes) pour définir un taquet de tabulation, ou désactivez Tab Stop pour annuler ce taquet.

Vous pouvez tester votre boîte de dialogue (voir la section "Comment tester une boîte de dialogue" de ce chapitre) pour vérifier si vos nouveaux taquets de tabulation sont corrects.

Regroupement de contrôles liés



Vous pouvez définir des groupes de contrôles pour permettre à l'utilisateur de se déplacer parmi ces contrôles à l'aide des touches fléchées.

La définition des groupes est identique à celle des taquets de tabulation (décrite précédemment). Si vous utilisez l'outil Set Groups ou la commande Options | Set Groups, Resource Workshop spécifie qu'un contrôle est repéré comme appartenant à un groupe en l'entourant d'un cadre ombré. Vous pouvez aussi définir un groupe en activant l'attribut Group de la boîte de dialogue Style du contrôle considéré.

Notez que vous marquez uniquement le *premier* contrôle de chaque groupe avec l'attribut Group. En respectant la séquence d'ajout des contrôles, Resource Workshop traite tous les contrôles suivants comme des éléments du groupe, jusqu'à ce qu'il rencontre un autre contrôle doté de l'attribut Group.

Souvenez-vous des points suivants à propos des groupes :

- L'ordre d'ajout des contrôles est important.
- Lorsque vous sélectionnez un "groupe" de contrôles en faisant glisser un cadre de sélection ou à l'aide d'un *Maj*-clic, il ne s'agit pas d'un groupe avec le sens utilisé dans cette section. Il s'agit simplement de plusieurs contrôles sélectionnés. L'attribut Group peut seulement être affecté par l'outil Set Group, la commande Options | Set Group, ou la case à cocher Group de la boîte de dialogue Style.

Comment réordonner les contrôles (séquence de tabulation)

Vous pouvez spécifier l'ordre dans lequel les utilisateurs accéderont aux contrôles dans une boîte de dialogue. Comme spécifié dans la section précédente, l'ordre est très important lorsque vous définissez des groupes de contrôles liés.

Pour spécifier l'ordre de tabulation des contrôles dans une boîte de dialogue :

1. Sélectionnez les contrôles que vous souhaitez déplacer dans la séquence.
Pour spécifier l'ordre de *tous* les contrôles dans la boîte de dialogue, sélectionnez aucun d'entre eux.
2. Cliquez sur l'outil Set Order. Le curseur de la souris se transforme en icône Set Order.



Chaque contrôle est numéroté pour indiquer sa place dans la séquence. Si vous ne choisissez que certains contrôles dans l'étape 1, vous ne verrez que les numéros d'ordre de ces contrôles.

Notez le message `Next Item` au bas de l'éditeur Dialog. Il vous indique le numéro d'ordre que Resource Workshop affectera au prochain contrôle sur lequel vous cliquerez.

3. Cliquez sur les éléments auxquels vous souhaitez affecter de nouveaux numéros. L'éditeur Dialog affiche un cadre autour des contrôles déjà sélectionnés.

En affectant le nouveau numéro, vous pouvez "revenir en arrière" en cliquant à nouveau le dernier contrôle que vous venez de cliquer : vous revenez ainsi au numéro précédent. Vous pouvez continuer ainsi en cliquant sur les différents contrôles dans l'ordre inverse.

4. Une fois les nouveaux numéros affectés, cliquez sur le Sélecteur pour continuer l'édition.

Vous pouvez aussi ordonner vos contrôles avec la commande menu Options | Set Order. Ensuite, choisissez Options | Modify Controls pour continuer à éditer votre boîte de dialogue. (Vous n'avez pas besoin de choisir cette commande si vous cliquez sur tous les contrôles sélectionnés.)

**Agencement,
alignement, et
redimensionnement
des contrôles**

Une fois les contrôles ajoutés à votre boîte de dialogue, vous pouvez utiliser le menu éditeur Dialog pour aligner et redimensionner les contrôles, et les disposer en rangées et colonnes.

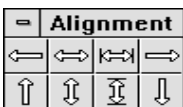
**Comment aligner
plusieurs contrôles**

Pour aligner des contrôles à l'aide de la commande Align :

1. Sélectionnez les contrôles à aligner.
Notez que le cadre de sélection entoure tous les contrôles sélectionnés. Les contrôles individuels qui seront affectés par les options d'alignement sont repérés par un cadre ombré.
2. Choisissez Align | Align. La boîte de dialogue Align controls s'affiche
3. Choisissez les options Vertical Alignment et Horizontal Alignment requises (voir ci-dessous) et cliquez sur OK pour exécuter le déplacement des contrôles sélectionnés.

Tableau 15.3
Options d'alignement

Option	Description
<i>Horizontal Alignment</i>	
No Change	Aucun changement dans l'alignement horizontal.
Left Sides	Aligne les contrôles afin que leurs côtés gauches coïncident avec le côté gauche du cadre de sélection.
Centers	Aligne les contrôles afin que leurs centres horizontaux coïncident avec le centre du cadre de sélection.
Right Sides	Aligne les contrôles afin que leurs côtés droits coïncident avec le côté droit du cadre de sélection.
Space Equally	Déplace les contrôles horizontalement à l'intérieur du cadre de sélection afin qu'ils soient régulièrement espacés.
Center in Dialog	Déplace le cadre de sélection horizontalement afin qu'il soit centré dans la boîte de dialogue. La position relative des contrôles individuels à l'intérieur du cadre de sélection reste inchangée.
<i>Vertical Alignment</i>	
No Change	Pas de changement dans l'alignement vertical.
Tops	Aligne les contrôles afin que leurs côtés supérieurs coïncident avec le côté supérieur du cadre de sélection.
Centers	Aligne les contrôles afin que leurs centres verticaux coïncident avec le centre du cadre de sélection.
Bottoms	Aligne les contrôles afin que leurs côtés inférieurs coïncident avec le côté inférieur du cadre de sélection.
Space Equally	Déplace les contrôles verticalement à l'intérieur du cadre de sélection afin qu'ils soient régulièrement espacés.
Center in Dialog	Déplace le cadre de sélection verticalement afin qu'il soit centré dans la boîte de dialogue. La position relative des contrôles individuels à l'intérieur du cadre de sélection reste inchangée.



Au lieu d'utiliser *Align* | *Align Controls*, vous pouvez utiliser la palette *Alignment*. Sélectionnez les contrôles à aligner, puis choisissez un outil dans la palette *Alignment*.

La seule option d'alignement qui n'apparaît pas dans la palette *Alignment* est l'option *Space Equally*. Vous pouvez naturellement utiliser la boîte de dialogue *Align Controls*, mais aussi espacer les contrôles régulièrement en "allongeant" le cadre de sélection :

1. Sélectionnez les contrôles à espacer régulièrement. Un cadre de sélection entoure les contrôles sélectionnés.

2. Agrandissez le cadre de sélection en maintenant la touche *Ctrl* enfoncée tout en faisant glisser le bord du cadre de sélection dans le sens où vous souhaitez espacer vos contrôles, soit :

Pour espacer les contrôles de manière régulière dans la direction horizontale, allongez la bordure gauche ou droite du cadre de sélection.

Pour espacer les contrôles de manière régulière dans la direction verticale, allongez la bordure inférieure ou supérieure du cadre de sélection.

Si vous allongez un *coin* du cadre de sélection, une boîte de dialogue s'affiche pour vous permettre de disposer les contrôles en rangées et en colonnes.

Comment disposer les contrôles en colonnes et rangées

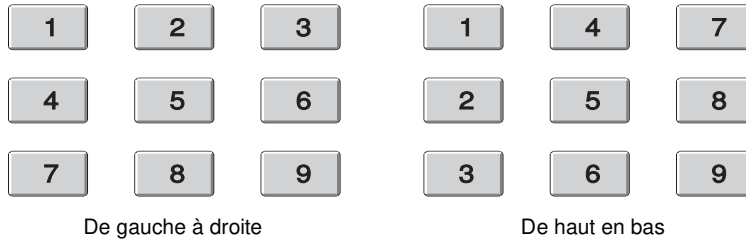
La commande Align | Array dispose les contrôles en colonnes et rangées, en les alignant horizontalement ou verticalement avec un espacement régulier et en les renumérotant afin qu'ils soient tous ordonnés.

Pour utiliser la commande Array :

1. Sélectionnez les contrôles que vous souhaitez disposer en colonnes et rangées.
Notez que le cadre de sélection entoure tous les contrôles sélectionnés. Les contrôles individuels qui seront affectés par la commande Array sont repérés par un cadre ombré.
2. Si nécessaire, agrandissez ou réduisez la taille du cadre de sélection pour entourer la zone qui contiendra les colonnes et les rangées. Par exemple, si vous agrandissez le cadre de sélection, la commande Array déplacera les contrôles vers les nouvelles limites définies par le cadre de sélection.
3. Choisissez Align | Array ou cliquez sur l'outil Duplicate.
4. Sous Array Layout, spécifiez le nombre de rangées et de colonnes souhaitées.
5. Sous Order, spécifiez l'ordre souhaité (de gauche à droite ou de haut en bas). La figure suivante montre comment les deux options Order disposent différemment neuf contrôles en trois colonnes.



Figure 15.5
Options d'ordre des
contrôles



6. Une fois les options spécifiées, cliquez sur OK.



Pour plusieurs contrôles sélectionnés, l'outil Duplicate a le même effet que Align | Array. Lorsqu'un seul contrôle est sélectionné, il a le même effet que Edit | Duplicate (voir la section "Ajout de plusieurs copies d'un contrôle" de ce chapitre).

**Comment
redimensionner
plusieurs contrôles**

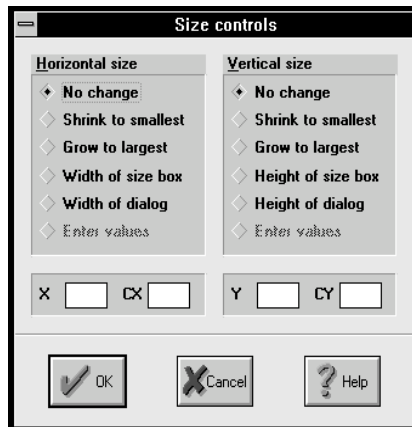
Vous pouvez redimensionner plusieurs contrôles sélectionnés avec les options de la boîte de dialogue Size Controls :

1. Sélectionnez les contrôles à redimensionner.

Notez que le cadre de sélection entoure tous les contrôles sélectionnés. Les contrôles individuels qui seront affectés par les options de modification de taille sont repérés par un cadre ombré.

2. Choisissez Align | Size. La boîte de dialogue ci-dessous s'affiche :

Figure 15.6
Boîte de dialogue
Size Controls



3. Pour redimensionner les contrôles sélectionnés, choisissez les options Vertical Size et Horizontal Size requises (voir ci-dessous) et cliquez sur OK.

Tableau 15.4
Options Size

Option	Description
<i>Horizontal Size</i>	
No Change	Aucun changement de taille horizontal.
Shrink to Smallest	Réduit la largeur des contrôles pour s'adapter au contrôle sélectionné le plus étroit.
Grow to Largest	Augmente la largeur des contrôles pour s'adapter au contrôle sélectionné le plus large.
Width of Size Box	Redimensionne les contrôles pour qu'ils occupent toute la largeur du cadre de sélection.
Width of Dialog	Redimensionne les contrôles pour qu'ils occupent toute la largeur de la boîte de dialogue.
<i>Vertical Size</i>	
No Change	Aucun changement de taille vertical.
Shrink to Smallest	Réduit la hauteur des contrôles pour s'adapter au contrôle sélectionné le plus réduit en hauteur.
Grow to Largest	Augmente la hauteur des contrôles pour s'adapter au contrôle sélectionné le plus important en hauteur.
<i>Height of Size Box</i>	Redimensionne les contrôles pour qu'ils occupent toute la hauteur du cadre de sélection.
Height of Dialog	Redimensionne les contrôles pour qu'ils occupent toute la hauteur de la boîte de dialogue.

Une fois que vous avez choisi les options de hauteur et de largeur voulues, sélectionnez OK

Pour annuler un redimensionnement, choisissez Edit | Undo, appuyez sur *Alt+Retour arrière* ou sélectionnez l'outil Undo (Défaire).

Options de dimensionnement pour un contrôle unique

Si vous sélectionnez un seul contrôle et que vous choisissez Align | Size, les options suivantes sont disponibles :

- No Change (horizontal et vertical)
- Width of Dialog (horizontal)
- Height of Dialog (vertical)
- Enter Values (horizontal et vertical)

Les options No Change, Width of Dialog et Height of Dialog fonctionnent comme indiqué au tableau 15.4.

L'option Enter Values permet de spécifier à la fois la taille et la position du contrôle sélectionné. Les valeurs X et Y définissent la distance entre le coin supérieur gauche du contrôle et le coin supérieur gauche de la boîte de dialogue (directement sous la barre de titre). Les valeurs CX et CY définissent la largeur et la hauteur du contrôle. (Toutes les valeurs sont mesurées en unités dialogue.)

Vous pouvez "défaire" toute édition exécutée dans l'éditeur Dialog (positionnement de contrôles, alignement, suppression, etc.), avec l'outil Undo ou la commande Edit | Undo. Undo s'applique aussi bien aux commandes qui portent sur des groupes de contrôles et aux commandes qui modifient des contrôles isolément.

Contrôles boutons

Les contrôles boutons incluent les boutons radio, les boutons-poussoirs, les cases à cocher et les boîtes de groupe. Pour ajouter des boutons à une boîte de dialogue, utilisez la palette Tools (voir la section "Palette Tools" de ce chapitre) ; pour définir les attributs des nouveaux boutons ou pour modifier les contrôles boutons existants, utilisez la boîte de dialogue Button Style (pour afficher cette boîte de dialogue, double-cliquez sur le contrôle bouton à modifier).

Pour modifier le type d'un bouton, choisissez une nouvelle option dans Button Type (voir tableau ci-dessous).

Tableau 15.5
Types de bouton

Type de bouton	Description
Push Button	Bouton portant un libellé. L'utilisateur clique sur le bouton qui envoie un message BN_CLICKED à la fenêtre parent.
Default Push Button	Identique à un bouton-poussoir, mais comprend également une bordure en gras qui indique qu'il s'agit de la réponse par défaut si l'utilisateur appuie sur <i>Entrée</i> .
Check Box	Bouton rectangulaire avec texte à sa gauche ou à sa droite. La case est marquée par un X lorsqu'elle est sélectionnée. L'application doit elle-même activer et désactiver la case.
Auto Check Box	Identique à une case à cocher, mais Windows active et désactive la case à la place de l'application.

Tableau 15.5
Types de bouton
(suite)

Type de bouton	Description
3-State	Identique à une case à cocher, mais avec un troisième état éventuel : le bouton apparaît en grisé pour indiquer que son état est inconnu ou indéterminé. C'est l'application qui doit activer, désactiver et estomper la case.
Auto 3-State	Identique à un bouton à 3 états, mais Windows active et désactive la case à la place de l'application.
Radio Button	Bouton circulaire avec un texte identificateur à sa gauche ou à sa droite. Le cercle se remplit d'un point noir lorsqu'il est sélectionné. C'est l'application qui doit remplir ou effacer le point. Les boutons radio doivent apparaître en groupes. En général, un groupe de boutons radio offre à l'utilisateur un ensemble d'options mutuellement exclusives. Lorsque l'utilisateur clique sur un bouton radio, il envoie un message BN_CLICKED à la fenêtre parent.
Auto Radio Button	Identique à un bouton radio mais Windows remplit ou efface le point à la place de l'application.
Group Box	Boîte rectangulaire utilisée pour regrouper les boutons visuellement. Vous pouvez également inclure un libellé qui s'affichera dans l'angle supérieur gauche de la boîte.
User Button	Boutons personnalisés pour compatibilité avec Windows 2.0. Il est conseillé de ne pas utiliser ces contrôles sous Windows 3.0 mais plutôt les boutons Owner Draw (voir ci-dessous).
Owner Draw	Option qui permet à l'application de créer le bouton. Lorsqu'il doit apparaître, le bouton envoie un message WM_DRAWITEM à son parent.

Les options de justification (*Alignment*) déterminent si le texte des boutons radio et des cases à cocher s'affiche à gauche ou à droite du bouton.

Valeurs d'identificateur des contrôles bouton-poussoir

Windows définit un ensemble de valeurs d'identificateur de contrôle pour les boutons-poussoirs standard utilisés pour quitter les boîtes de dialogue. Vous pouvez entrer l'un des noms (en lettres majuscules) ou l'une des valeurs d'identificateur du tableau 15.6. Cependant, à l'opposé des Contrôles Personnalisés Windows de Borland (BWCC, décrits à l'annexe B), le changement d'identificateur de contrôle d'un bouton Windows standard ne modifie pas automatiquement le texte du bouton. Par exemple, pour créer un bouton OK, vous devez initialiser l'identificateur de contrôle à IDOK et la chaîne *Caption* à OK.

Tableau 15.6
Contrôles boutons
Windows prédéfinis

Nom Identificateur	Identificateur Type	Valeur
IDOK	1	OK
IDCANCEL	2	Cancel
IDABORT	3	Abort
IDRETRY	4	Retry
IDIGNORE	5	Ignore
IDYES	6	Yes
IDNO	7	No

Contrôles barre de défilement

Les fenêtres, les boîtes de dialogue et les boîtes liste utilisent des barres de défilement pour préciser que des informations supplémentaires sont affichables. Par exemple, si une boîte liste de noms de fichiers permet d'afficher 10 noms et que le répertoire contient 20 fichiers, une barre de défilement indique à l'utilisateur l'existence de noms supplémentaires.

Une barre de défilement est un rectangle pourvu de flèches directionnelles à chaque extrémité. Entre les flèches, une icône carrée (appelée boîte de défilement ou *ascenseur*) indique la position approximative de l'affichage par rapport à l'ensemble des informations. Par exemple, si l'ascenseur d'une boîte liste contenant des noms de fichiers se trouve au milieu de la barre de défilement, cela signifie que l'utilisateur est en train de consulter les noms des fichiers situés au milieu de la liste.

Dans une boîte de dialogue, vous pouvez ajouter des barres de défilement vertical et horizontal à l'emplacement de votre choix.

Pour placer des barres de défilement dans une boîte de dialogue, utilisez la palette Tools ou les deux commandes de barre de défilement du menu Controls. Pour définir ou modifier les attributs d'une barre de défilement, double-cliquez sur celle-ci pour affichez la boîte de dialogue Scroll Bar Style.

La boîte de dialogue Scroll Bar Style contient les options communes et les options d'attributs de contrôle présentées à partir des tableaux 15.1 et 15.2, ainsi que les options qui alignent la barre de défilement à l'intérieur du cadre de sélection (cf. tableau ci-dessous).

Tableau 15.7
Options
d'alignement pour
barre de défilement

Option	Description
None	La barre de défilement remplit tout le cadre de sélection (valeur par défaut). Si vous redimensionnez le cadre de sélection, vous pouvez modifier les proportions de la barre de défilement, élargissant ainsi les boutons flèche et la boîte de défilement.
Top Left	Une barre de défilement horizontal apparaît en haut du cadre de sélection et occupe toute la largeur de ce cadre. Une barre de défilement vertical apparaît sur la gauche du cadre de sélection et occupe toute la hauteur de ce cadre. La barre de défilement apparaît toujours avec sa largeur standard.
Bottom Right	Une barre de défilement horizontal apparaît en bas du cadre de sélection et occupe toute la largeur de ce cadre. Une barre de défilement vertical apparaît sur la droite du cadre de sélection et occupe toute la hauteur de ce cadre. La barre de défilement apparaît toujours avec sa largeur standard.

Contrôles boîte liste

Une boîte liste est un rectangle qui contient une liste de chaînes texte. En général, l'utilisateur peut consulter puis sélectionner un ou plusieurs éléments dans une boîte liste. La boîte liste envoie alors à la fenêtre parent un message relatif au(x) élément(s) sélectionné(s).

Si la liste des éléments est plus longue ou plus large que la boîte liste, vous pouvez ajouter des barres de défilement à la boîte.

En dehors des options communes décrites dans le tableau 15.1, la boîte de dialogue List Box Style comprend également les options Owner Drawing et List Box (cf. tableaux ci-dessous).

Les options Owner Drawing permettent de déterminer si la liste d'une boîte liste doit être créée par la boîte liste ou par l'application. Sélectionnez l'un des attributs suivants :

Tableau 15.8
Options Owner
Drawing

Option	Description
Not Owner Draw	Le contrôle boîte liste crée la liste (valeur par défaut).
Fixed	L'application crée la boîte liste en réponse aux messages WM_DRAWITEM. L'application peut également répondre aux messages WM_COMPAREITEM, WM_DELETEITEM et WM_MEASUREITEM. Le contrôle boîte liste transmet un message WM_MEASUREITEM à l'application uniquement au moment de la création initiale de la boîte liste, ce qui fixe la hauteur des éléments de la boîte liste.

Tableau 15.8
Options Owner
Drawing (suite)

Option	Description
Variable	L'application crée la boîte liste en réponse aux messages WM_DRAWITEM. L'application peut également répondre aux messages WM_COMPAREITEM, WM_DELETEITEM et WM_MEASUREITEM. Le contrôle boîte liste envoie un message WM_MEASUREITEM à l'application pour chaque élément de la boîte liste. La hauteur de chaque élément peut varier.

Les options List Box permettent de mieux définir une boîte liste. Choisissez une ou plusieurs options du tableau ci-dessous :

Tableau 15.9
Options List Box

Option	Description
Notify	Envoie un message de saisie à la fenêtre parent lorsque l'utilisateur clique sur un élément de la liste (défaut).
Sort	Trie la liste dans l'ordre alphabétique.
Multiple Select	Permet à l'utilisateur de sélectionner plusieurs éléments. L'utilisateur peut également activer et désactiver certains éléments.
Don't Redraw	Évite de recréer une boîte liste modifiée.
Tab Stops	Organise les informations de la boîte liste en colonnes. La largeur de colonne par défaut est de 32 unités dialogue ou 8 caractères. Utilisez les caractères <i>Tab</i> (\x09) pour formater le texte. Si vous souhaitez modifier la largeur de colonne, l'application doit définir ses propres taquets de tabulation avec le message LB_SETTABSTOPS.
Integral Height	Réduit la hauteur de la boîte liste, si nécessaire, jusqu'au multiple entier le plus proche de la hauteur de police en cours (valeur par défaut). Par exemple, une boîte liste peut contenir trois éléments affichés en totalité, et un quatrième élément dont l'affichage est tronqué. Si Integral Height est activé, la taille de la boîte liste va diminuer au moment de l'exécution à l'espace requis pour trois éléments (trois fois la hauteur de police).
Multi Column	Crée une boîte liste dans laquelle le texte passe du bas d'une colonne au haut de la colonne suivante. Pour afficher du texte supplémentaire, l'utilisateur fait défiler la boîte liste horizontalement. Si vous activez cette option, l'application doit envoyer le message LB_SETCOLUMNWIDTH pour définir la largeur de toutes les colonnes (en pixels).
Pass Keyboard Input	Transmet à l'application les données saisies par l'utilisateur.

Tableau 15.9
Options List Box
(suite)

Option	Description
Extend Select	Avec les boîtes liste à sélection multiple, cet attribut modifie la manière dont la sélection multiple s'effectue. L'utilisateur peut ainsi sélectionner plusieurs éléments dans la liste.
Has Strings	Si vous avez choisi l'option Owner Drawing Fixed ou Variable, la boîte liste enregistre le texte de chaque élément de la liste à l'aide d'un message LB_INSERTSTRING ou LB_ADDSTRING. La boîte liste peut également restituer les éléments de liste à l'aide du message LB_GETTEXT.
Scroll Bar Always	(Windows 3.1 uniquement) La boîte liste affiche toujours une barre de défilement vertical, quel que soit le nombre d'éléments de cette liste. La constante WINDOWS.H pour ce style est LBS_DISABLENOSCROLL.

Contrôles boîte de saisie

Le contrôle boîte de saisie permet à l'utilisateur d'entrer du texte à partir du clavier. Ce contrôle est utilisé par exemple dans la boîte de dialogue File Open.

Figure 15.7
Contrôle boîte de saisie dans la boîte de dialogue File Open



Pour définir les attributs d'un contrôle boîte de saisie, double-cliquez sur ce contrôle dans la boîte de dialogue. La boîte de dialogue Edit Text Style s'affiche.

Cette boîte de dialogue contient les options communes et les options d'attributs de contrôles présentées à partir du tableau 15.1, ainsi que les options ci-après :

- Alignement
- Barres de défilement
- Défilement automatique
- Options ligne simple ou lignes multiples
- Options minuscule/majuscule
- Autres options, y compris les options spécifiques à Windows 3.1

Le tableau suivant décrit les options de la boîte de dialogue Edit Text Style.

Tableau 15.10
Options de la boîte
de dialogue Edit
Text Style

Option	Description
<i>Alignement</i>	
Left	Aligne à gauche un texte de plusieurs lignes (valeur par défaut).
Right	Aligne à droite un texte de plusieurs lignes.
Center	Centre un texte de plusieurs lignes.
<i>Barres de défilement</i>	
Horizontal	Lorsque cette option est activée, le contrôle boîte de saisie dispose d'une barre de défilement horizontal en bas de sa fenêtre.
Vertical	Lorsque cette option est activée, le contrôle boîte de saisie dispose d'une barre de défilement vertical sur le bord droit de sa fenêtre.
<i>Défilement automatique</i>	
Horizontal	Si l'utilisateur tape un caractère au niveau du bord droit de la limite de la boîte de saisie, le texte défile automatiquement de 10 caractères vers la droite. Si l'utilisateur appuie sur <i>Entrée</i> , le texte défile à nouveau jusqu'à la position zéro.
Vertical	Si l'utilisateur appuie sur <i>Entrée</i> à la dernière ligne du contrôle boîte de saisie, le texte redéfile sur une pleine page. Par exemple, si la longueur du contrôle est de 5 lignes et que vous appuyez sur <i>Entrée</i> à la dernière ligne, le texte remonte de 5 lignes. Le curseur revient sur la ligne du haut. Pour que cette option puisse s'appliquer, n'oubliez pas de définir le contrôle boîte de saisie autorisant plusieurs lignes. Remarque : L'appui sur <i>Entrée</i> est sans effet lorsque Resource Workshop est en mode Test, mais vous constaterez cet effet dans votre application.
<i>Majuscule/minuscule</i>	
Case Insensitive	Affiche le texte exactement comme il est saisi (valeur par défaut).
Upper Case	Affiche tout le texte en lettres majuscules, quel que soit le mode de saisie.
Lower Case	Affiche tout le texte en minuscules, quel que soit le mode de saisie.
<i>Ligne(s) Line</i>	
Single Line	Limite la boîte de saisie à une seule ligne (valeur par défaut).
Multiple Line	Permet à l'utilisateur de taper le texte sur plusieurs lignes. Pour autoriser le défilement d'un texte de plusieurs lignes, activez l'option Vertical Automatic Scroll.

Tableau 15.10
Options de la boîte
de dialogue Edit
Text Style (suite)

Option	Description
<i>Autres options</i>	
Password	Si vous sélectionnez Password, les lettres tapées ne sont pas affichées (elles sont remplacées par des astérisques). Cette option est utile pour les mots de passe et les données confidentielles.
Convert OEM	Convertit le texte saisi dans le contrôle selon le jeu de caractères courant OEM puis reconvertit le texte en ANSI. Cette option est utile dans les boîtes de saisie de fichiers car elle permet à tout nom de fichier entré d'être traduit dans le jeu de caractères OEM, qui est utilisé par le système de fichiers DOS.
Keep Selection	Conserve le texte sélectionné en surbrillance, même si ce contrôle n'est pas commandé par le clavier. Par exemple, si un utilisateur met en surbrillance du texte dans un contrôle boîte de saisie, puis passe à un autre contrôle, le texte perd sa surbrillance, sauf si vous avez affecté le contrôle boîte de saisie de l'attribut Keep Selection.

Les options suivantes sont uniquement disponibles sous Windows 3.1.

Tableau 15.11
Styles Windows 3.1

Type	Description
Read Only	Le texte est accessible en lecture seulement. La constante WINDOWS.H pour ce style est ES_READONLY.
Want Return	Pour un contrôle boîte de saisie à plusieurs lignes qui est commandé par le clavier, la touche <i>Entrée</i> impose un saut de ligne dans ce contrôle. Si le contrôle n'est pas commandé par le clavier, le retour chariot s'applique au bouton-poussoir par défaut. Si le contrôle ne dispose pas de cet indicateur, l'utilisateur doit appuyer sur <i>Ctrl+Entrée</i> pour créer un saut de ligne. La constante WINDOWS.H pour ce style est ES_WANTRETURN.

Contrôles statiques

Un contrôle statique affiche un texte ou un graphique non modifiable par l'utilisateur. Vous pouvez utiliser les contrôles statiques pour libeller certaines parties de votre boîte de dialogue ou pour présenter les informations sous forme graphique.

Les contrôles statiques de la palette Tools sont les suivants :

- Texte statique
- Icône
- Cadre
- Rectangle

Les outils de ces
contrôles sont illustrés
par la figure 15.3.

Vous pouvez définir les attributs des contrôles statiques à l'aide de la boîte de dialogue Static Style, qui s'affiche par un double-clic sur le contrôle statique requis dans sa boîte de dialogue.

La boîte de dialogue Static Style comprend les options communes et les options d'attributs de contrôles présentées à partir de la section "édition des contrôles" plus haut dans ce chapitre (excepté les options de barre de défilement), ainsi que plusieurs options spécifiques aux contrôles statiques.

La case à cocher No Character Underline permet de désactiver le soulignement de caractères. Vous pouvez souligner un caractère texte dans votre contrôle statique en le précédant du symbole &. Si vous activez No Character Underline, le soulignement est désactivé et les symboles & sont affichés en tant que caractères littéraux.

Les options Control Type vous permettent de définir plus en détail le type de contrôle statique. Choisissez une des options ci-après :

Tableau 15.12
Options Control Type

Option	Description
Left Text	Affiche le texte tout à fait à gauche dans la bordure du contrôle (défaut). Si le texte s'étend au-delà de la bordure du cadre, il passe automatiquement à une nouvelle ligne. Le texte, dans tous ces styles, utilise la couleur Window Text en cours du panneau de configuration.
Left Text -No Wrap	Affiche le texte tout à fait à gauche dans la bordure du contrôle. Toute ligne de texte qui s'étend au-delà de la bordure du cadre est coupé.
Centered Text	Affiche le texte centré à l'intérieur de la bordure du contrôle. Si le texte s'étend au-delà de la bordure du cadre, il passe automatiquement à une nouvelle ligne.
Right Text	Affiche le texte tout à fait à droite dans la bordure du contrôle. Si le texte s'étend au-delà de la bordure du cadre, il passe automatiquement à une nouvelle ligne. Dans tout le texte statique excepté Simple Text, vous pouvez réaliser des tabulations de texte en tapant \T, et forcer le passage à une nouvelle ligne avec \R.

Tableau 15.12
Options Control Type

Option	Description
Simple Text	<p>Affiche une seule ligne de texte tout à fait à gauche. N'utilisez pas des caractères de tabulation, et ne forcez pas le passage à une nouvelle ligne.</p> <p>Simple Text ne gère pas le message WM_CTLCOLOR. En plus de recevoir sa couleur texte à partir du panneau de configuration, son fond utilise la couleur du fond de fenêtre en cours.</p>
White Rectangle	Affiche un rectangle plein qui utilise le jeu de couleurs du fond de fenêtre en cours dans le panneau de configuration. La couleur Windows par défaut du fond de fenêtre est le blanc.
Gray Rectangle	Affiche un rectangle rempli qui utilise le jeu de couleurs du fond d'écran (le bureau) en cours dans le panneau de configuration. La couleur Windows par défaut du bureau est le gris.
Black Rectangle	Affiche un rectangle rempli qui utilise le jeu de couleurs du cadre de fenêtre en cours dans le panneau de configuration. La couleur Windows par défaut des cadres de fenêtre est le noir.
White Frame	Affiche un cadre vide avec un contour plein qui utilise le jeu de couleurs du fond de fenêtre en cours dans le panneau de configuration. La couleur Windows par défaut du fond de fenêtre est le blanc.
Gray Frame	Affiche un cadre vide avec un contour plein qui utilise le jeu de couleurs du fond d'écran (le bureau) en cours dans le panneau de configuration. La couleur Windows par défaut du bureau est le gris.
Black Frame	<p>Affiche un cadre vide avec un contour plein qui utilise le jeu de couleurs du cadre de fenêtre en cours dans le panneau de configuration. La couleur Windows par défaut des cadres de fenêtre est le noir.</p> <p>Remarque : Lorsque vous ajoutez un cadre à votre boîte de dialogue, il vous semble rempli avec la couleur du fond de fenêtre en cours. Si vous activez le mode Test, vous verrez le cadre tel qu'il apparaîtra lors de l'exécution.</p>
Icon	Affiche une icône. Utilisez le bouton Edit Icon pour lancer l'éditeur Bitmap de manière à pouvoir éditer l'icône.

Contrôles statiques icône

Resource Workshop vous permet d'afficher des icônes dans une boîte de dialogue. L'icône doit faire partie du projet courant en tant que ressource intégrée ou liée.

Pour placer un contrôle statique icône dans votre boîte de dialogue :



Si la boîte de saisie Caption ne contient pas l'identificateur ou le nom de la ressource icône, l'icône ne s'affichera pas.

1. Cliquez sur le contrôle statique icône dans la palette Tools et faites-le glisser dans votre boîte de dialogue. Placez le cadre à l'endroit où doit apparaître l'icône.
2. Double-cliquez à l'intérieur du cadre de sélection du contrôle pour afficher la boîte de dialogue Static Style.
3. Dans la boîte de dialogue Static Style, entrez le nom ou l'identificateur de la ressource icône dans la boîte de saisie Caption, et cliquez sur le bouton radio approprié : Text pour un nom, Number pour un identificateur.
Si vous utilisez un identificateur (son nom ou sa valeur numérique) dans la boîte de saisie Caption, vous devez sélectionner l'option Number. Par exemple, si vous avez créé l'identificateur GLOBE avec une valeur de 1300, vous pouvez entrer GLOBE ou 1300 dans la boîte de saisie Caption, mais dans les deux cas vous devez sélectionner Number.
Si l'icône se nomme GLOBE et que vous n'avez pas créé d'identificateur, tapez GLOBE dans la boîte de saisie Caption et sélectionnez le bouton radio Text.
4. Choisissez OK. L'icône apparaît dans votre boîte de dialogue.
5. Si vous souhaitez éditer l'icône, double-cliquez-la pour afficher à nouveau la boîte de dialogue Static Style. Le bouton Edit Icon est maintenant activé. Cliquez-le pour lancer l'éditeur Bitmap de manière à pouvoir éditer l'icône.

Contrôles boîtes à options combinées

Une boîte à options combinées associe une boîte liste (contrôle qui permet à l'utilisateur de consulter et de sélectionner des chaînes) avec un contrôle statique (texte non modifiable par l'utilisateur) ou avec un contrôle boîte de saisie (zone où un utilisateur peut entrer son texte).

Figure 15.8
Boîte à options combinées dans la boîte de dialogue Open File



Vous pouvez définir les attributs d'une boîte à options combinées en utilisant la boîte de dialogue Combo Box Style. Pour afficher la boîte de dialogue Combo Box Style, double-cliquez sur la boîte à options combinées de votre boîte de dialogue.

Les trois options Type vous permettent de définir la boîte à options combinées.

Tableau 15.13
Types de boîtes à options combinées

Option	Description
Simple	La liste déroulante s'agrandit toujours pour afficher les éléments de la liste. En outre, l'utilisateur peut éditer les éléments de la liste (défaut).
Drop Down	Lorsque la boîte de dialogue s'affiche, la boîte à options combinées comprend une seule ligne de texte éditable. L'utilisateur peut cliquer sur la flèche vers le bas pour agrandir la liste, et éditer tous les éléments de la liste.
Drop Down List	Fonctionne comme une liste déroulante mais elle est statique. L'utilisateur peut sélectionner des éléments mais ne peut apporter aucune modification.

Les options Owner Drawing vous permettent de déterminer si la liste contenue dans la boîte liste doit être dessinée par la boîte liste elle-même ou par l'application.

Tableau 15.14
Options Owner Drawing

Option	Description
No	Le contrôle boîte liste dessine la liste (défaut).
Fixed	L'application dessine la boîte liste en réponse aux messages WM_DRAWITEM. L'application peut aussi répondre aux messages WM_COMPAREITEM, WM_DELETEITEM, et WM_MEASUREITEM. Le contrôle boîte liste envoie le message WM_MEASUREITEM à l'application uniquement au moment où la boîte liste est dessinée. La taille de la boîte liste est donc fixe.
Variable	L'application dessine la boîte liste en réponse aux messages WM_DRAWITEM. L'application peut aussi répondre aux messages WM_COMPAREITEM, WM_DELETEITEM, et WM_MEASUREITEM. Le contrôle boîte liste envoie le message WM_MEASUREITEM à l'application pour chaque élément de la boîte liste. La taille de la boîte liste peut donc varier.
Has Strings	Si vous avez choisi Fixed ou Variable, la boîte liste mémorise le texte de chaque élément avec le message LB_SETTEXT. La boîte liste peut aussi restituer les éléments de liste à partir de LB_GETTEXT.

La boîte de dialogue Combo Box Style comprend les options communes et les options d'attributs de contrôles répertoriées à partir du tableau 15.1 ainsi que les options spécifiques aux contrôles de la boîte à options combinées.

Tableau 15.15
Attributs de la boîte
à options combinées

Option	Description
Vertical Scroll	Place une barre de défilement vertical dans la boîte liste.
Sorted	Trie automatiquement les éléments d'une boîte liste par ordre alphabétique.
Integral Height	Redimensionne la boîte liste pendant l'exécution pour que tous les éléments de la liste s'affichent complètement (défaut). Si vous voulez contrôler précisément la hauteur de la boîte liste, désactivez cette option.
OEM Conversion	Convertit le texte que l'utilisateur tape dans le jeu de caractères OEM courant, et reconvertit le texte en ANSI. Cette option est utile dans les boîtes de saisie des noms de fichiers, car tout nom de fichier entré pourra être traduit dans le jeu de caractères OEM, utilisé par le système de fichiers DOS.
AutoHorizontal	Défilement automatique du texte vers la gauche s'il dépasse la largeur du contrôle.
Vertical Scroll Always	(Windows 3.1 uniquement). La boîte à options combinées affiche toujours une barre de défilement vertical, quel que soit le nombre d'éléments qu'elle contient. La constante WINDOWS.H pour ce style est CBS_DISABLENOSCROLL.

Contrôles personnalisés

Si vous souhaitez utiliser un contrôle qui ne correspond pas à un des types prédéfinis de Windows, vous pouvez utiliser un *contrôle personnalisé*. Les contrôles prédéfinis dont nous avons déjà parlé dans ce chapitre (boîtes liste, barres de défilement, boutons, etc.) sont des contrôles standard. Ils ont été développés par Microsoft et font partie de Windows. Le contrôle personnalisé est une autre classe de fenêtres à inclure dans vos boîtes de dialogue.

Il existe deux types de contrôles personnalisés : les contrôles à installer et ceux spécifiques à l'application. Vous devez implémenter les contrôles personnalisés installables en utilisant un fichier bibliothèque liée dynamiquement. Les contrôles personnalisés spécifiques à une application sont implémentés dans l'application elle-même. Resource Workshop les dessine sous forme de boîtes grises ou de cadres vides.

Création de contrôles personnalisés

Si vous souhaitez créer vos propres contrôles personnalisés, vous devez les concevoir et les mémoriser dans les fichiers DLL. La création de classes de contrôles personnalisés est décrite dans le fichier CUSTCNTL.RW, qui a été copié sur votre disque dur par le programme d'installation de Resource Workshop.

Installation d'une bibliothèque de contrôles (.DLL ou .VBX)

Les contrôles personnalisés sont mémorisés dans des DLL. Pour ajouter des contrôles personnalisés à votre boîte de dialogue, installez les fichiers DLL ou VBX appropriés. Ensuite, les contrôles personnalisés de cette DLL seront disponibles comme tout autre contrôle Windows standard.

Pour installer un fichier DLL contenant une bibliothèque de contrôles personnalisés, procédez comme suit :

1. Dans l'éditeur Dialog, choisissez Options | Install Control Library. Vous verrez la boîte de dialogue Install a New Control Library.
2. Spécifiez le fichier .DLL ou .VBX du contrôle personnalisé.
3. Choisissez OK.

Vous pouvez maintenant ajouter les contrôles du fichier DLL dans votre boîte de dialogue.

Affichage des contrôles personnalisés

Avant d'ajouter des contrôles personnalisés à votre boîte de dialogue, choisissez Options | Preferences pour voir comment vos boîtes de dialogue vont s'afficher. Si Drawing Type est positionné à Normal, et si l'option Draw Custom Controls as Frames est activée, vos contrôles personnalisés s'afficheront comme des contours rectangulaires. Dans ce cas, vous voudrez sans doute changer un de ces paramètres, ou bien les deux. Reportez-vous au tableau 15.18 pour de plus amples informations sur les options Drawing Type, et à la section "Personnalisation de l'éditeur Dialog" pour de plus amples informations sur l'option Draw Custom Controls as Frames.

Ajout d'un contrôle personnalisé

Une fois le fichier DLL contenant les contrôles personnalisés installé, vous pouvez ajouter le contrôle voulu à vos boîtes de dialogue.

Si le type de vos contrôles personnalisés est reconnu par Resource Workshop (voir la section "Comment travailler avec les contrôles" de ce chapitre), leurs icônes apparaissent dans la colonne (ou les colonnes) de droite de la palette Tools. Vous pouvez les sélectionner directement à partir de la palette.

Si le type de vos contrôles personnalisés n'est pas reconnu par Resource Workshop, vous devez suivre les étapes ci-dessous :

1. Cliquez sur l'outil de contrôle personnalisé ou choisissez Control | Custom pour afficher la boîte de dialogue New Custom Control.

2. Dans la liste déroulante à côté de Class, choisissez le contrôle personnalisé souhaité. Resource Workshop affiche un exemple du contrôle personnalisé sélectionné au centre de la boîte de dialogue.
3. Lorsque vous avez sélectionné le contrôle personnalisé de votre choix, choisissez OK. Le curseur de la souris se transforme en croix, indiquant ainsi qu'il est prêt à positionner le contrôle personnalisé.
4. Cliquez dans la fenêtre de la boîte de dialogue à l'endroit où vous souhaitez placer le contrôle personnalisé.

Comment tester une boîte de dialogue



Pour tester votre boîte de dialogue afin que vous puissiez voir les effets des modifications apportées, sélectionnez l'outil Test ou choisissez Options | Test Dialog.

Appuyez sur *Tab* et sur les touches fléchées pour voir comment vous déplacer dans votre boîte de dialogue. Vous pouvez aussi taper du texte pour voir comment le texte défile dans un contrôle boîte de saisie. Vérifiez si vos contrôles sont ordonnés dans l'ordre souhaité.

Lorsque vous testez une boîte de dialogue, la ligne d'état au bas de l'éditeur Dialog indique `Test`.

Pour quitter le mode test et retourner au mode d'édition, exécutez une des étapes suivantes :

- Cliquez sur le bouton OK ou Cancel de la boîte de dialogue.
- Choisissez à nouveau Options | Test Dialog.
- Appuyez sur *Entrée*.
- Cliquez deux fois sur le Sélecteur (le premier clic vous bascule de la boîte de dialogue à l'éditeur Dialog).

Enregistrement d'une boîte de dialogue

Prenez l'habitude d'enregistrer vos changements au fur et à mesure, au lieu d'attendre un message de Resource Workshop vous invitant à le faire lors de la fermeture du projet. Exécutez une des étapes ci-dessous pour enregistrer les changements effectués à une boîte de dialogue.

- Enregistrer le projet entier
- Enregistrer la boîte de dialogue dans un fichier script de ressources de dialogue (.DLG).

Enregistrement du projet

Pour enregistrer le projet, choisissez File | Save Project. Resource Workshop compile les ressources modifiées depuis la dernière compilation et les mémorise dans le fichier projet.

Enregistrement d'une boîte de dialogue dans un fichier

Vous pouvez aussi stocker vos ressources de boîtes de dialogue dans un fichier script de ressources de dialogue .DLG. Un fichier dialogue contient normalement le script ressource pour une ou plusieurs boîtes de dialogue, mais il peut contenir toute information contenue dans un fichier .RC, y compris d'autres ressources (à la fois intégrées et liées) et projets. Si vous ajoutez un fichier dialogue à un projet, Resource Workshop ajoute automatiquement une référence **rcinclude** dans la fenêtre projet pour référencer un fichier dialogue.

Pour mémoriser des boîtes de dialogue dans un fichier .DLG, ajoutez un fichier dialogue au projet. Vous pourrez ensuite décider de mémoriser les boîtes de dialogue dans ce fichier dialogue à mesure que vous les créez. Pour ajouter un fichier dialogue à un projet et y mémoriser une nouvelle ressource :

Le chapitre 3 décrit la procédure d'ouverture d'un projet.

1. Choisissez File | Add to Project. La boîte de dialogue Add File to Project s'affiche.
2. Dans la boîte liste déroulante File Type, sélectionnez DLG Resource Script.
3. Tapez un nouveau nom de fichier dans la boîte de saisie File Name.
4. Choisissez OK. Une boîte de dialogue s'affiche avec le message : "`<filename>` does not exist. Create?".
5. Choisissez Yes.

Une instruction **rcinclude** s'ajoute à votre fenêtre projet. Si vous ne la voyez pas, sélectionnez la fenêtre projet, puis choisissez View | By File.

Vous avez créé un fichier .DLG vide. Maintenant, vous allez créer une nouvelle boîte de dialogue et l'ajouter au fichier.

1. Choisissez Resource | New. La boîte de dialogue New Resource s'affiche.
2. Sélectionnez le fichier script de dialogue créé dans la boîte à options combinées "Place resource in".
3. Dans la boîte liste Resource Type, double-cliquez DIALOG, ou sélectionnez DIALOG et choisissez OK.

Lorsque vous éditez votre boîte de dialogue et quittez l'éditeur Dialog, vous verrez une entrée DIALOG au-dessous de l'instruction **rcinclude** du fichier .DLG de la fenêtre projet.

Visualisation de deux boîtes de dialogue

Pour visualiser ou comparer deux boîtes de dialogue en même temps, suivez les étapes ci-après :

1. Dans la fenêtre projet, double-cliquez sur le nom de la première boîte de dialogue à visualiser. L'éditeur Dialog s'active et affiche cette boîte de dialogue.
2. Cliquez sur l'outil Test ou choisissez Options | Test Dialog.
3. Cliquez deux fois sur le bouton Réduction de l'éditeur Dialog (le premier clic vous bascule de la boîte de dialogue à la fenêtre de l'éditeur Dialog). Votre boîte de dialogue est maintenant une boîte non modale que vous pouvez déplacer.
4. Revenez à la fenêtre projet et double-cliquez sur le nom de la seconde boîte de dialogue à visualiser. Un second éditeur Dialog s'active.
5. Cliquez à nouveau sur l'outil Test ou choisissez Options | Test Dialog dans le second éditeur Dialog.
6. Cliquez deux fois sur le bouton Réduction du second éditeur Dialog.

Vous disposez maintenant de deux boîtes de dialogue non modales que vous pouvez placer côte à côte.

Pour quitter le mode Test, cliquez sur le bouton OK ou Cancel de votre boîte de dialogue, ou double-cliquez sur son icône menu Control.

Personnalisation de l'éditeur Dialog

Resource Workshop permet de modifier certaines parties de l'éditeur Dialog. Choisissez Options | Preferences pour afficher la boîte de dialogue Preferences.

Les unités de la ligne d'état déterminent l'unité de mesure que la ligne d'état utilise pour afficher les informations.

Tableau 15.16
Unités ligne d'état

Unité	Description
Dialog	Utilise l'unité dialogue comme unité de mesure sur la ligne d'état. Dans une unité dialogue, y est égal à $1/8$ de la hauteur de police et x à $1/4$ de la largeur de police.
Screen	Utilise le pixel comme unité de mesure sur la ligne d'état.

Les options Selection Border vous permettent de modifier l'aspect du cadre qui entoure les contrôles sélectionnés.

Tableau 15.17
Options Selection
Border

Option	Description
Thick frame	Le cadre de sélection est épais, comme le cadre standard autour d'une application Windows ou d'une fenêtre de boîte de dialogue (défaut).
Handles	Le cadre de sélection est un contour rectangulaire avec des poignées (petits carrés) à chaque angle et au milieu de chaque côté.

Les options Drawing Type déterminent le type d'affichage des éléments de votre boîte de dialogue dans l'éditeur Dialog.

Tableau 15.18
Options Drawing
Type

Type de dessin	Description
Draft	Dessine chaque contrôle sous forme d'un rectangle avec son identificateur de contrôle centré à l'intérieur. Cette option vous permet aussi de voir l'espace occupé par le cadre de sélection du contrôle.
Normal	Dessine les contrôles Windows standard comme ils apparaîtront au moment de l'exécution. Le dessin de contrôles personnalisés est déterminé par la case à cocher Draw Custom Controls as Frames, décrite brièvement.

Tableau 15.18
Options Drawing
Type (suite)

Type de dessin	Description
WYSIWYG	Lorsque cette option est sélectionnée, Resource Workshop crée les fenêtres dialogue et les contrôles fenêtre enfant, et les contrôles dessinent eux-mêmes. Cette option est plus lente, mais plus précise. Les contrôles personnalisés installables se dessinent eux-mêmes. C'est l'option par défaut.

Les options Selection "définissent les règles" sur la façon de sélectionner les contrôles. Si vous travaillez avec des contrôles peu espacés, vous activerez ces options pour obtenir une meilleure précision et pour éviter de sélectionner des contrôles par inadvertance.

Tableau 15.19
Options Selection

Option	Description
	Select Near Border, Cette option s'applique à l'action de cliquer pour sélectionner des contrôles. Si elle est activée, vous devez cliquer <i>sur</i> la bordure du contrôle. Si elle est désactivée, vous devez cliquer n'importe où à l'intérieur de la bordure du contrôle.
Selection Rectangle Surrounds	Cette option s'applique à l'action de faire glisser le cadre de sélection pour sélectionner des contrôles. Si elle est activée, vous devez entourer entièrement le contrôle (ou les contrôles) à l'aide du rectangle de sélection. Si elle est désactivée, le rectangle de sélection a seulement besoin de toucher le contrôle (ou les contrôles).

Pour plus d'informations sur le langage script ressource, utilisez le système d'aide.

Dans le langage script ressource, chaque type de contrôle dialogue a une syntaxe unique. Par exemple, le texte statique centré utilise l'instruction CTEXT. Néanmoins, l'instruction CONTROL peut spécifier tout type de contrôle de dialogue. Si vous souhaitez que Resource Workshop génère uniquement des instructions CONTROL dans votre script ressource, plutôt que des instructions de contrôle dialogue spécifiques, sélectionnez l'option Generate CONTROL Statements Only.

La case à cocher Draw Custom Controls as Frames est uniquement disponible lorsque Drawing Type est initialisé à Normal (voir le tableau 15.18). Lorsque l'option est activée, les contrôles personnalisés sont dessinés comme des contours rectangulaires vides. Lorsque l'option est désactivée, les contrôles personnalisés sont dessinés comme des rectangles gris avec leur texte (s'il existe) dans un rectangle blanc au centre. Dessiner des contrôles personnalisés comme des cadres peut accélérer le dessin de vos boîtes de dialogue à l'écran.

Si vous cochez la case Ctl3d.dll, l'éditeur Dialog applique aux contrôles (boutons radio, cases à cocher) l'aspect en relief utilisé sous Windows.

Création de menus

L'utilisateur choisit une commande à partir d'une liste de commandes, appelée menu. Grâce aux applications gérées par menus, l'utilisateur n'a pas besoin de connaître la syntaxe d'une ligne de commande complexe. Par conséquent, les menus facilitent l'utilisation d'une application.

La plupart des applications Windows ont une barre menu, en haut de l'écran, qui contient les noms des menus de l'application. Chaque menu contient un ensemble de commandes. Par exemple, la plupart des programmes Windows comprennent un menu Fichier, avec des commandes de création, d'ouverture, d'enregistrement et d'impression de fichiers.

L'éditeur Menu de Resource Workshop vous permet de créer et d'éditer facilement des menus pour votre application. Travailler avec des menus implique quatre étapes fondamentales :

1. Lancer l'éditeur Menu.

Si vous créez une nouvelle ressource menu, l'éditeur Menu vous présente un "modèle" de menu. Si vous modifiez une ressource menu existante, celle-ci apparaît dans l'éditeur Menu.

2. Créer et éditer le menu.
3. Tester le menu.
4. Enregistrer le menu.

Terminologie

Ce chapitre utilise plusieurs termes pour décrire les éléments d'une ressource menu :

- Une *commande surgissante* provoque l'affichage d'un menu. Elle peut apparaître dans la barre menu, comme les noms de menus Windows standard Fichier et Aide. Elle peut aussi apparaître à l'intérieur de menus surgissants, où elle provoque l'affichage d'un autre menu, appelé "menu en cascade".

Voir la documentation de votre compilateur pour obtenir des informations sur la fonction `TrackPopupMenu`.

- Un *menu surgissant* est une boîte rectangulaire qui contient une liste de commandes d'application, à partir de laquelle un utilisateur choisit une commande. Il existe deux formes de menus surgissants :
 - Les menus *déroulants* sont affichés à partir d'une barre menu ou depuis un menu. Ils sont liés à une commande surgissante et sont toujours affichés à partir de ce nom de commande. Par exemple, le menu déroulant File apparaît toujours directement sous la commande surgissante File de la barre menu.
 - Les menus *flottants* peuvent apparaître n'importe où dans la fenêtre application. Leur position est contrôlée par la fonction `TrackPopupMenu`.
- Un *élément de menu* est une commande qui apparaît dans un menu (par exemple, Open, Save ou Print).
- Un *séparateur de menu* est une ligne qui divise les éléments de menu en groupes logiques. Leur seule fonction est de faciliter la lecture et l'emploi d'un menu. Vous ne pouvez pas éditer des séparateurs de menu.

Lancement de l'éditeur Menu

Les deux sections suivantes décrivent la procédure de lancement de l'éditeur Menu pour créer un nouveau menu ou éditer un menu existant.

Création d'un nouveau menu

Le chapitre 14 décrit l'ouverture d'un projet.

Pour créer un nouveau menu :

1. Vérifiez que le projet auquel vous voulez ajouter un menu est ouvert. Vous pouvez choisir File | New Project pour créer un nouveau projet, ou File | Open Project pour ouvrir un projet existant.
2. Lorsque vous avez ouvert un projet, choisissez Resource | New afin de créer une nouvelle ressource pour ce projet. La boîte de dialogue New Resource apparaît.
3. Dans la boîte de dialogue New Resource, faites défiler le contenu de la liste Resource Type jusqu'à MENU, puis double-cliquez sur MENU, ou bien cliquez sur MENU et OK.

Resource Workshop affiche l'éditeur Menu, avec un menu par défaut que vous pouvez éditer.

Edition d'un menu existant

Pour éditer un menu existant, ouvrez le projet dans lequel le menu est stocké et exécutez une des procédures ci-dessous :

- Double-cliquez sur le nom de menu dans la fenêtre projet.
- Mettez en surbrillance le nom de menu, puis choisissez **Resource | Edit**. Resource Workshop affiche l'éditeur Menu, avec le menu choisi.

Ecran de l'éditeur Menu

L'éditeur Menu est divisé en trois volets : un volet Outline similaire au script source, un volet Test Menu et un volet Attribute qui vous permet d'éditer la ligne en surbrillance dans le volet Outline. Vous pouvez modifier la position des volets via la commande View (voir le tableau 16.1).

Volet Attribute

Le volet Attribute est l'endroit où vous éditez les commandes surgissantes et les éléments de menu, affectez des valeurs d'identificateurs, et définissez des attributs pour vos menus et éléments de menu. Vous pouvez également associer des raccourcis à chaque élément de menu. L'instruction en cours d'édition apparaît en surbrillance dans le volet Outline. Vos sélections prennent effet lorsque vous exécutez une des étapes suivantes :

- Appuyez sur *Entrée* pour entrer la modification.
- Appuyez sur *Inser* ou choisissez **Menu | New Menu Item** pour entrer la modification et insérer un nouvel élément.
- Utilisez *Ctrl+↑*, *Ctrl+↓* ou la souris pour déplacer le pointeur sur une autre instruction.
- Appuyez sur *Ctrl+P* ou choisissez **Menu | New Pop-up** pour insérer un nouveau menu surgissant.
- Appuyez sur *Ctrl+S* ou choisissez **Menu | New Separator** pour insérer un nouveau séparateur.

Les options du volet Attribute sont décrites dans le tableau 16.2.

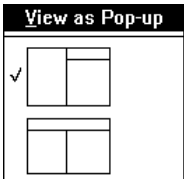
Volet Test Menu

Le volet Test Menu affiche votre menu et vous permet de le tester.

Le menu surgissant de la commande Pop-up par défaut contient une seule commande, appelée Item. L'éditeur Menu actualise automatiquement ce menu test à mesure que vous modifiez le volet Outline.

Vous pouvez utiliser le menu View pour modifier le type d'affichage du menu test : à la fois sur la barre menu et relativement aux autres volets.

Tableau 16.1
Sélections du menu
View



Sélection menu	Description
View as Pop-up,	<p>Contrôle si les commandes surgissantes,, dans le menu test,, sont affichées sur la barre menu ou dans un menu surgissant.</p> <p>Par défaut, cette option est désactivée, et les commandes surgissantes dans le menu test sont affichées sur la barre menu. Laissez cette option désactivée si votre ressource menu contient la structure de menu entière de l'application, et que vous voulez l'afficher comme elle devrait apparaître à l'utilisateur.</p> <p>Si vous travaillez sur un menu flottant,, activez cette option pour afficher le menu test tel qu'il devrait réellement apparaître. La commande <i>Pop-up</i> apparaît sur la barre menu. Sélectionnez Pop-up pour afficher le menu lui-même.</p>
First graphic	Ce graphique représente la configuration par défaut des volets, avec le volet Test Menu au-dessus du volet Outline, et à droite du volet Attribute.
Second graphic	Activez ce graphique pour placer le volet Test Menu en haut de la fenêtre d'édition, comme une barre menu normale.

Volet Outline

Le volet Outline affiche les commandes surgissantes, les éléments de menu et les séparateurs du nouveau menu en pseudocode. La ligne supérieure du volet correspond au nom du menu. Les autres lignes sont des *instructions* qui définissent les menus surgissants et les éléments de menu. Vous pouvez également afficher un TurboMenu : pointez sur le volet Outline et cliquez sur le bouton droit de la souris, ou appuyez sur *Alt+F10* n'importe où dans le volet.

Le pseudocode du volet Outline est conçu pour vous faciliter la tâche avec la structure du menu. Pour consulter le code complet avec tous les paramètres pour chaque instruction, éditez le script ressource du menu (voir la section "Edition d'un script ressource menu" de ce chapitre).

L'édition réelle du menu se fait dans le volet Attribute. Pour passer du volet Outline au volet Attribute, et vice-versa, appuyez sur *F6*.

Pour éditer une instruction, sélectionnez-la en exécutant une des opérations ci-dessous :

- Appuyez sur *Ctrl+↑* ou *Ctrl+↓*.
- Choisissez un élément de menu dans le volet Test Menu.
- Cliquez sur une ligne dans le volet Outline et appuyez sur *F6*.
- Appuyez sur *↑* ou *↓* dans le volet Outline, puis sur *F6*.

Pour vous déplacer à l'intérieur du volet Attribute, utilisez la souris pour positionner le pointeur à n'importe quel endroit et y faire des sélections.

Edition de menus

Une fois le menu chargé dans l'éditeur Menu, vous pouvez commencer à ajouter de nouvelles commandes de menu, de nouveaux menus surgissants ou séparateurs, à déplacer, copier ou supprimer toute partie du menu.

Ajout de nouvelles instructions

Pour plus d'informations sur l'ajout d'éléments de menu, reportez-vous à la section "Ajout d'éléments de menu et de séparateurs" de ce chapitre.

Pour ajouter une nouvelle instruction dans un menu (menu surgissant, élément de menu ou séparateur), placez le curseur dans la fenêtre Outline sur la ligne précédant l'endroit où vous voulez placer l'instruction. Pour insérer une instruction en première ligne, mettez en surbrillance la ligne du haut (MENU_1 ou le nom de la ressource menu).

Lorsque vous aurez décidé où placer la nouvelle instruction et que vous aurez mis en surbrillance la ligne appropriée, vous pourrez ajouter une nouvelle instruction en choisissant une des commandes du menu Menu.

Commande	Effet
New Pop-up	insère un nouveau menu surgissant avec un seul élément.
New Menu Item	insère un élément.
New Separator	insère un séparateur.
New File Pop-up	ajoute un menu File (Fichier) générique complet.
New Edit Pop-up	ajoute un menu Edit, avec les commandes Cut (Couper), Copy (Copier) et Paste (Coller).
New Help Pop-up	ajoute un menu Help (Aide) prédéfini.
La commande Check Duplicates	est décrite à la section "Test des menus".

Si vous voulez ajouter une commande surgissante qui apparaîtra en première position dans la barre menu (position généralement occupée par le menu File), mettez en surbrillance la première ligne du volet Outline (MENU_1 ou le nom de la ressource menu).

Ajout d'éléments de menu et de séparateurs

Pour ajouter un élément ou un séparateur dans un menu :

1. Choisissez la position du nouvel élément ou séparateur dans le menu et mettez en surbrillance dans le volet Outline la ligne qui précède cette position (l'éditeur Menu insère le nouvel élément ou séparateur sous la ligne en surbrillance).
2. Appuyez sur *INS* ou choisissez Menu | New Menu Item.
3. Tapez le nom du nouvel élément ou séparateur.
4. Appuyez sur *Entrée*

Vous pouvez également ajouter trois menus par défaut à votre projet :

- Menu | New file pop-up ajoute le menu File comme première commande menu (extrémité gauche de la barre menu). Le nouveau menu File contient les commandes New, Open, Save, Save As, Print, Page Setup, Printer Setup et Exit.
- Menu | New edit pop-up ajoute le menu Edit en deuxième position (de la gauche vers la droite) dans la barre menu. Le nouveau menu Edit contient les commandes Undo, Cut, Copy et Paste.
- Menu | New help pop-up ajoute le menu Help comme dernière commande menu (extrémité droite de la barre menu). Le nouveau menu Help contient les commandes Index, Keyboard, Commands, Procedures, Using help et About.

Edition des éléments de menu

Un nouveau élément de menu ajouté a la désignation générique "Item". Pour le rendre plus compréhensible, vous devez l'éditer.

Le premier élément de menu ajouté est automatiquement sélectionné, et vous pouvez l'éditer immédiatement. Par contre, si vous ajoutez d'autres éléments de menu, vous devez d'abord sélectionner un élément de menu avant de pouvoir l'éditer.

Utilisation du volet Attribute

Dans le volet Attribute, utilisez la souris, la touche *Tab* ou la touche *Maj.+Tab* pour positionner le pointeur sur le champ à éditer.

Le tableau suivant décrit les sélections disponibles dans le volet Attribute.

Tableau 16.2
Sélections du volet
Attribute de l'éditeur
Menu

Sélection	Description
Item Text	Le nom d'un menu surgissant ou d'un élément de menu (commande), et une description facultative de son raccourci, le cas échéant.
Item help	Une chaîne qui décrit l'élément de menu. Ce texte est stocké dans la table de chaînes de votre projet (si vous n'avez pas défini de table de chaînes pour votre projet, le système en ajoute une lorsque vous utilisez cette option). Le texte entré ici peut servir de ligne d'aide affichable dans une barre d'état.
Item ID	Un identificateur unique pour un élément de menu. Cette boîte de saisie n'est pas disponible lorsqu'un menu surgissant ou un séparateur de menu est sélectionné.
Item Type	Menu surgissant, élément de menu ou séparateur de menu.
Break Before	Contrôle le format des commandes menu dans la barre menu ou dans les menus surgissants. Choisissez une des options suivantes :
- No Break	Pas d'interruption avant cette commande.
- Menu Bar Break	Commence une nouvelle ligne dans la barre menu. Dans un menu surgissant, commence une nouvelle colonne et dessine une ligne verticale pour séparer les colonnes.
- Menu Break	Commence une nouvelle ligne dans la barre menu ou une nouvelle colonne dans un menu surgissant.
- Help Break	Déplace l'élément de menu ou la commande surgissante jusqu'à l'extrême droite de la barre menu. Utilisez cette option uniquement avec des instructions de niveau supérieur qui s'affichent dans la barre menu.
Initial state	\$!Menu (éditeur);options Initial StateContrôle l'état initial de l'élément de menu. Choisissez l'une des options suivantes :
- Enabled	Dans son état initial, la commande est activée. Utilisez la fonction <i>EnableMenuItem</i> pour changer l'état de l'élément de menu.
- Disabled	Dans son état initial, la commande est désactivée. L'utilisateur ne peut différencier les commandes Enabled et Disabled sur le menu. Utilisez la fonction <i>EnableMenuItem</i> pour changer l'état de l'élément de menu.
- Grayed	Dans son état initial,, la commande est désactivée et son texte apparaît en grisé. Ceci indique à l'utilisateur que la commande n'est pas actuellement disponible. Utilisez la fonction <i>EnableMenuItem</i> pour changer l'état de l'élément de menu.
- Checked	Place une coche à côté de la commande. Choisissez cette option si l'élément est une bascule et si vous souhaitez que la commande apparaisse initialement avec une coche. Utilisez la fonction <i>CheckMenuItem</i> pour changer l'état de la commande.

Tableau 5.2
Sélections du volet
Attribut de l'éditeur
Menu (suite)

Sélection	Description
Key	
Modifieurs	Spécifie la deuxième touche de raccourci associée à un élément de menu. Choisissez l'une des options suivantes :
- Alt	Le raccourci est une combinaison de touches avec <i>Alt</i> (par exemple, <i>Alt+W</i>) et autres.
- Shift	Le raccourci inclut la touche <i>Maj.</i> (par exemple, <i>Maj.+F1</i>).
- Control	Le raccourci inclut la touche <i>Ctrl</i> (par exemple, <i>Ctrl+F3</i>).
- Invert	Désactive la fonction de clignotement (inversion vidéo momentanée), fonction interne de Windows qui fait clignoter la commande associée au raccourci ; l'utilisateur peut ainsi identifier le menu qui contient le raccourci. Lorsque vous créez un raccourci
Key type	Décrit le type de raccourci associé à un élément de menu. Choisissez l'une des options suivantes :
- ASCII	Le raccourci est une touche ASCII standard (les touches ASCII doivent être placées entre guillemets). Le caret (^) indique que la touche est combinée avec la touche <i>Ctrl</i> . En général, les caractères ASCII ne s'utilisent pas seuls pour les raccourcis
- Virtual key	Une "touche virtuelle" est une touche de fonction, une touche fléchée ou une touche d'édition telle que <i>Home</i> ou <i>End</i> . Pour ces touches virtuelles

Saisie du texte de l'élément

Le texte de l'élément est le nom de menu ou la commande qui apparaît dans la barre menu ou dans le menu. Lorsque la sélection Item Text apparaît en surbrillance, vous pouvez taper une nouvelle chaîne texte directement dans la boîte. Vous pouvez aussi utiliser les touches *Origine Fin* et les touches fléchées pour déplacer le curseur dans la boîte de saisie.

Si vous voulez que l'utilisateur puisse choisir le menu ou la commande en tapant une lettre dans le nom de menu ou de commande, placez un symbole & immédiatement devant cette lettre. Windows affichera le texte avec cette lettre soulignée.

Voir le chapitre 17 pour plus d'informations sur les raccourcis.

Pour lier le raccourci *Ctrl+F4* à la nouvelle commande *Stored Order*, ajoutez le texte du raccourci à vos menus.

- Utilisez le caractère de tabulation (`\t`) pour séparer le titre de menu du texte de raccourci par une tabulation (par exemple, `&Stored Order\tCtrl+F4`).
- Utilisez le caractère d'alignement à droite (`\a`) pour aligner à droite le texte du raccourci (par exemple, `Stored Order\aCtrl+F4`).

Les applications Windows utilisent généralement le signe plus pour montrer des combinaisons de touches, telles que *Maj.+Suppr* ou *Ctrl+Maj.+F4*.

Saisie des identificateurs d'éléments

Pour plus de détails, reportez-vous au chapitre 14

Tous les éléments de menu doivent être identifiés de façon unique. Lorsque vous ajoutez un nouvel élément, Resource Workshop affecte automatiquement un identificateur, différent des autres identificateurs d'élément de cette ressource menu. Vous pouvez accepter cette valeur, ou bien la remplacer par un autre nom ou numéro unique.

Si vous tapez un nom, Resource Workshop vérifie si un identificateur de ce nom existe. Si ce n'est pas le cas, une boîte de dialogue vous demande si vous voulez créer un identificateur. Cliquez sur OK. Si vous voulez utiliser le nom en tant qu'identificateur d'élément, vous devez créer un identificateur. A partir de là, créez le nouvel identificateur comme spécifié à la section "Ajout d'identificateurs" du chapitre 14.

Déplacement et copie d'instructions

Vous pouvez utiliser les options *Cut*, *Copy* et *Paste* du menu *Edit* pour déplacer et copier les instructions dans le volet *Outline* de l'éditeur *Menu*.

Pour déplacer une instruction, mettez-la en surbrillance et choisissez *Edit | Cut*. Notez que vous ne pouvez pas couper la dernière (ou la seule) instruction du volet *Outline*. Il doit toujours y avoir au moins un élément de menu, une commande surgissante ou un séparateur de menu dans le volet *Outline*. Pour insérer l'instruction coupée ou copiée dans votre menu, mettez en surbrillance l'instruction située immédiatement devant le point d'insertion, et choisissez *Edit | Paste*.

Pour copier une instruction, mettez-la en surbrillance et choisissez *Edit | Copy*. L'instruction en surbrillance reste dans le volet *Outline*, et une copie est placée dans le presse-papiers.

Suppression d'instructions de menu

Mettez en surbrillance l'instruction à supprimer, puis appuyez sur *Suppr* ou choisissez Edit | Delete pour la supprimer. Vous pouvez aussi choisir Edit | Cut pour supprimer l'instruction et la copier dans le presse-papiers.

Notez les points suivants à propos de la suppression d'instructions de menu :

- Si vous supprimez une instruction `POPUP`, la commande surgissante qu'elle définit et tous les éléments contenus dans le menu surgissant sont aussi supprimés.
- Vous ne pouvez pas supprimer les instructions `_End Popup_`.
- Vous ne pouvez pas supprimer la dernière (ou la seule) instruction du volet Outline. Il doit toujours y avoir au moins un élément de menu, une commande surgissante ou un séparateur de menu dans le volet Outline.

Création de menus flottants



Un menu flottant peut s'afficher n'importe où dans l'espace fenêtre de l'application. Il n'est pas lié à une barre menu.

Chaque menu flottant doit être enregistré comme une ressource menu distincte à l'intérieur du fichier projet.

Pour créer un menu flottant :

1. Choisissez Resource | New et sélectionnez Menu à partir de la boîte de dialogue New Resource.
2. Choisissez View | View as Pop-up pour voir le menu flottant tel qu'il apparaîtra à l'écran pendant l'exécution.
Lorsque vous visualisez le menu dans le volet Test Menu, il semble toujours attaché à la barre menu, mais tant que votre code utilise correctement la fonction *TrackPopupMenu*, le menu sera flottant pendant l'exécution.
3. Sélectionnez la première ligne du volet Outline (MENU_1 ou le nom de la ressource menu).
4. Appuyez sur la touche *Inser* pour ajouter au moins un élément de menu en tête du volet Outline.
5. Sélectionnez la chaîne `POPUP "Pop-up"` dans le volet Outline.
6. Appuyez sur la touche *Suppr* pour supprimer l'instruction `POPUP`, son élément de menu et l'instruction `_End Popup_`.
7. Ajoutez d'autres éléments de menu.

8. Editez les éléments de menu dans le volet Attribute.
9. Enregistrez votre projet.

Test des menus

L'éditeur Menu permet de tester immédiatement les modifications apportées ; le menu est modifié en temps réel, et vous pouvez l'afficher à tout moment pour juger de son apparence. Lorsque vous choisissez un élément de menu, ce dernier s'affiche en surbrillance dans le volet Outline, et ses propriétés apparaissent dans le volet Attributes. (Vous pouvez annuler cette fonction en désactivant Menu | Track test menu.

L'éditeur Menu comprend aussi un outil de débogage intégré que vous pourrez utiliser pour rechercher les identificateurs d'élément de menu en double. Si vous choisissez Menu | Check Duplicates, l'éditeur Menu recherche les doublons et affiche, le cas échéant, une boîte de dialogue avec le message "Duplicate command value found".

Une fois la boîte de messages refermée, l'éditeur Menu met en surbrillance l'instruction qui contient ce doublon. Vous devez exécuter une des procédures suivantes :

Pour voir l'identificateur de menu dans votre ressource, choisissez Resource | Edit as Text.

- Si l'identificateur d'élément est un nombre, entrez un nouveau nombre qui n'entre pas en conflit avec les autres identificateurs d'élément.
- Si l'identificateur d'élément est une chaîne, la boîte Item ID contient une chaîne texte, et la boîte située à droite de Item ID contient un nombre. Pour changer la valeur d'identificateur,
 1. Choisissez Resource | Identifiers pour afficher la boîte de dialogue Identifiers.
 2. Faites défiler la liste des identificateurs jusqu'à celui que vous recherchez.
 3. Cliquez sur le bouton Change et tapez une nouvelle valeur qui n'entre pas en conflit avec les autres identificateurs d'élément.
 4. Cliquez sur OK ou appuyez sur *Entrée* pour modifier la valeur.
 5. Cliquez dans la fenêtre de l'éditeur Menu pour poursuivre l'édition de votre menu. Vous pouvez aussi laisser la boîte de dialogue Identifiers ouverte pour une prochaine utilisation.

Par exemple, si vous affectez la valeur 101 à deux identificateurs *wmnu_List* et *wmnu_Asc*, l'option Menu | Check Duplicates génère le message "Duplicate command value found" et l'éditeur Menu met *wmnu_Asc* en surbrillance (le second des deux identificateurs). Comme spécifié plus haut, vous devrez changer la valeur de *wmnu_Asc* par une valeur différente de 101, 102 ou 104 (les valeurs des autres identificateurs du menu).

Edition de scripts ressource menu

Pour travailler avec le script ressource d'un menu, sélectionnez le nom du menu de la fenêtre projet en le cliquant. Choisissez ensuite Resource | Edit As Text pour afficher le script ressource dans l'éditeur de texte interne.

Pour éditer, par exemple, le script ressource du menu exemple que vous créez dans la prochaine section, vous pouvez ouvrir le projet qui contient ce menu, puis mettre le menu en surbrillance et choisir Resource | Edit As Text. Resource Workshop ouvre son éditeur de texte interne et affiche les codes source comme suit :

Voir l'index de l'aide en ligne pour la description des options de script ressource pour les menus.

```
MENU_1 MENU
BEGIN
  POPUP "&Widgets"
  BEGIN
    MENUITEM "&List\tCtrl+L", wmnu_List
    MENUITEM "&Add...\tCtrl+A", wmnu_Add
    MENUITEM SEPARATOR
    POPUP "A&rrange List"
    BEGIN
      MENUITEM "&Ascending\tCtrl+F2", wmnu_Asc
      MENUITEM "&Descending\tCtrl+F3", wmnu_Desc
    END
  END
END
```

Utilisez l'éditeur pour modifier directement le script ressource. Par exemple, pour changer les deux options mémoire de menu LOADONCALL et MOVEABLE (valeurs par défaut) en PRELOAD et FIXED :

Vous pouvez aussi effectuer ces modifications en choisissant Resource | Memory Options.

1. Dans l'éditeur de texte, modifiez la première ligne du script comme suit :

```
MENU_1 MENU PRELOAD FIXED
```
2. Pour compiler votre saisie actuelle et vérifier si elle est correcte, choisissez Compile | Compile Now.

Le menu Compile est uniquement disponible dans l'éditeur de texte. Si vous retournez à l'éditeur Menu, puis choisissez Resource | Edit as Text pour basculer à nouveau dans l'éditeur de texte, Resource Workshop insère une option mémoire par défaut dans le script. La première ligne du script devient :

```
MENU_1 MENU PRELOAD FIXED DISCARDABLE
```

3. Si vous voulez quitter l'éditeur Menu, choisissez la commande Close dans la boîte Control-menu de la fenêtre de l'éditeur de texte. Resource Workshop vous invite à compiler. Si vous cliquez sur Yes, Resource Workshop compile le menu et active la fenêtre projet.

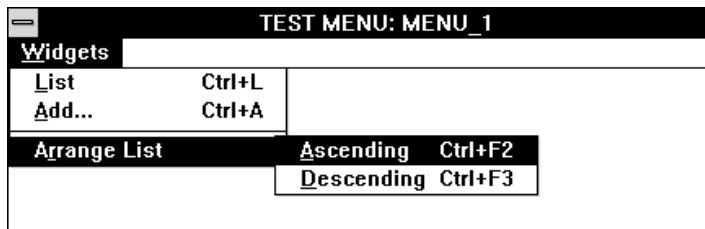


Ne perdez pas votre temps en insérant des commentaires dans votre script ressource, ou en formatant le texte. En effet, le compilateur incrémental de Resource Workshop exécute son propre formatage et abandonne tous les commentaires.

Exemple de menu

Cette section vous explique comment créer le menu surgissant simple Widgets avec l'éditeur Menu.

Figure 16.1
Menu exemple



Les deux premières commandes du menu (List et Add) vous permettent d'observer une liste d'objets existants, ou d'ajouter de nouveaux objets. La troisième commande, Arrange List, génère un menu surgissant avec deux commandes supplémentaires (Ascending et Descending) utilisées pour choisir l'ordre de tri de la liste d'objets.

Widgets et Arrange List sont des commandes surgissantes. Widgets affiche le menu Widgets, et Arrange List affiche le menu en cascade qui contient les commandes Ascending et Descending.

List, Add, Ascending et Descending sont des éléments (ou commandes) de menu. Si l'utilisateur clique sur une de ces commandes, l'application exécute une action. Les points de suspension (...) situés après la commande Add signifie que l'application affiche une boîte de dialogue lorsque l'utilisateur choisit cette commande. Notez le séparateur menu qui sépare les commandes List et Add de la commande surgissante Arrange List.

Création du menu exemple avec l'éditeur Menu

Grâce à certaines fonctions de l'éditeur Menu, la création du menu exemple est plus facile que sa création avec l'éditeur de texte. En outre, l'éditeur Menu gère les identificateurs de menu et les mémorise dans un fichier d'identificateurs distinct, si vous en avez créé un. De plus, vous pouvez tester le menu au fur et à mesure de sa création.

Pour créer le menu exemple :

1. Vérifiez si vous avez déjà ouvert un projet.
2. Lorsque le projet est ouvert, choisissez Resource | New.
3. Resource Workshop affiche la boîte de dialogue New Resource. Faites défiler la liste Resource Type jusqu'à MENU, puis double-cliquez. Resource Workshop ajoute une nouvelle ressource menu à la fenêtre projet. Ensuite, il affiche le nouveau menu dans l'éditeur Menu et la première instruction du volet Outline (POPUP "Pop-up") apparaît en surbrillance.
4. Pour renommer l'instruction initiale du menu (de Pop-up à Widgets), tapez &Widgets dans la boîte de saisie Item Text du volet Attribute, et appuyez sur *Entrée*.

L'éditeur Menu actualise à la fois les volets Menu Test et Outline.



Dans le menu test, notez que le W de Widgets est souligné. Vous pourrez donc appuyer sur *Alt+W* pour afficher le menu Widgets.

Ajout de commandes au menu Widgets

Ajoutez ensuite les commandes au menu Widgets.

1. Pour renommer le premier élément de menu et ajouter le texte du raccourci, appuyez sur *Ctrl+↓* pour mettre en surbrillance la deuxième ligne du volet Outline (MENUITEM "Item"), et tapez &List\tCtrl+L dans la boîte de saisie Item Text.

2. Tabulez jusqu'à la boîte de saisie Item ID et tapez `wmnu_List` pour entrer l'identificateur de cette commande. Appuyez sur *Entrée*.
Resource Workshop vous demande si vous souhaitez créer un nouvel identificateur. Appuyez sur *Entrée* pour afficher la boîte de dialogue New Identifier, puis à nouveau sur *Entrée* pour accepter la valeur affichée.
A ce stade, vous devriez ouvrir l'éditeur Accelerator pour créer la touche raccourcie *Ctrl+L*. Néanmoins, nous exécuterons cette procédure à la section "Création d'un exemple de table de raccourcis" du chapitre suivant, car nous n'avons pas encore traité les raccourcis.
3. Avec la commande List en surbrillance, ajoutez une nouvelle commande menu en appuyant sur *Inser* ou en choisissant Menu | New Menu Item.
4. Tapez `&Add. . . \tCtrl+A` dans la boîte Item Text pour modifier le texte du nouvel élément de menu.
5. Créez un identificateur pour l'élément. Pour cela, appuyez sur la touche *Tab* pour sélectionner le champ Item ID, tapez `wmnu_Add`, appuyez sur *Entrée*, et répondez aux messages comme précédemment.
6. Appuyez sur *Ctrl+S* pour placer un séparateur après la commande Add.
7. Avec la commande Add en surbrillance, appuyez sur *Ctrl+P* pour ajouter un nouveau menu surgissant. Changez le texte en `A&rrange List`.
Puisque vous voulez que des commandes supplémentaires apparaissent lorsque l'utilisateur choisit Arrange List, définissez-la en tant que commande surgissante et non comme un autre élément de menu. Une commande surgissante au milieu d'un menu crée un menu en cascade.

Ajout de commandes au menu Arrange List

Pour définir les deux commandes menu dans le menu surgissant Arrange List :

1. Appuyez sur *Ctrl+↓* jusqu'à ce que le premier élément du menu Arrange List apparaisse en surbrillance.
2. Changez l'élément de menu de "Item" à "`&Ascending\tCtrl+F2`".
3. Créez l'identificateur `wmnu_Asc` pour cette commande.
4. Appuyez sur *Inser* pour ajouter un nouvel élément de menu après "`&Ascending`", puis renommez-le en "`&Descending\tCtrl+F3`".
5. Créez l'identificateur `wmnu_Desc` pour cette commande.
6. Enregistrez le projet.

Test du menu

Testez le menu en cliquant sur la commande Widgets dans le volet Test Menu, et en faisant glisser le pointeur jusqu'à la commande Arrange List. Votre menu doit ressembler à celui de la figure 16.1, au début de ce chapitre.

Vous pouvez aussi vérifier les doublons des identificateurs de menu en choisissant Menu | Check Duplicates.

- S'il n'existe aucun doublon, le message "No duplicates found" s'affiche.
- S'il existe des doublons, le message "Duplicate command value found" s'affiche. La section "Tester des menus" de ce chapitre vous indique la procédure de correction des valeurs en double.

Création de raccourcis

Un *raccourci* est une combinaison de touches sur laquelle l'utilisateur appuie pour exécuter une tâche. Il remplace une commande menu et génère un message WM_COMMAND ou WM_SYSCOMMAND.

En général, vous créez des raccourcis pour dupliquer des commandes sur des menus surgissants. Si vous ouvrez le menu Edition dans la plupart des applications Windows, vous voyez les raccourcis suivants : *Alt+Retour arrière* (Annuler), *Maj+Suppr* (Couper), *Ctrl+Inser* (Copier) et *Maj+Inser* (Coller).

Vous mémorisez les définitions des raccourcis dans une table de raccourcis (ressource du raccourci). Chaque entrée de la table est un raccourci qui définit la combinaison de touches et la commande générée. Vous pouvez créer plusieurs tables de raccourcis (ou ressources) pour différentes parties de votre menu.

Resource Workshop vous propose l'éditeur Accelerator spécialement prévu pour créer et éditer les raccourcis de votre application. Lorsque vous travaillez avec les raccourcis, vous exécutez cinq tâches essentielles :

1. Lancer l'éditeur Accelerator.
2. Lancer l'éditeur Menu, de façon à définir des raccourcis correspondant à des éléments de menu.
3. Créer ou éditer une table de raccourcis.
4. Vérifier la présence de doublons dans la table de raccourcis.
5. Enregistrer la table de raccourcis.

Si vous ouvrez le projet démo (RWCDEMO.RC ou RWPDEMO.RC) et double-cliquez sur la ressource Accelerator, Resource Workshop affiche la table de raccourcis.

Voir la section "Création d'un exemple de table de raccourcis" pour illustrer l'utilisation de l'éditeur Accelerator.

Combinaisons de touches de la table des raccourcis

Les combinaisons de touches de votre table de raccourcis peuvent utiliser des touches *ASCII* ou des touches *virtuelles*.

- Une touche ASCII est une touche affichable : généralement un caractère alphanumérique ou un symbole de ponctuation.
- Une touche virtuelle est une touche de fonction, une touche fléchée ou une touche d'édition, telle que *Origine* ou *PgSuiv*. Bien que, dans certains cas, ces touches puissent afficher des caractères à l'écran, aucun standard ne spécifie les caractères qui apparaissent.

Touches ASCII

Toutes les touches ASCII doivent être entourées par des guillemets. Le symbole ^ signifie que la touche est combinée à la touche *Ctrl*. La case à cocher *Alt* du volet *Attribute* spécifie si la touche est combinée à la touche *Alt*.

Par exemple, les combinaisons de touches *Ctrl+W* et *Ctrl+Alt+W* sont représentées dans le volet *Outline* par "*^W*", mais une coche est présente dans la case à cocher *Alt* du volet *Attribute* pour *Ctrl+Alt+W*.



En principe, n'utilisez pas un caractère ASCII unique comme touche de raccourci. Vous devez plutôt le combiner avec la touche *Alt* ou *Ctrl* (*Ctrl+L* ou *Alt+L* au lieu du *L* tout seul).

Touches virtuelles

Windows a des identificateurs prédéfinis pour les touches virtuelles, tels que *VK_BACK* pour *Retour arrière* et *VK_F1* pour *F1*. Ces identificateurs, commençant tous par *VK_*, sont définis dans *WINDOWS.H*.

Les cases à cocher *Modifiers* du volet *Attribute* spécifient si la touche est combinée à *Ctrl*, *Alt*, *Maj* ou à toute combinaison des trois.

Par exemple, vous pourriez avoir deux raccourcis *VK_F1* dans votre table. Le premier raccourci pourrait être *Ctrl+F1* (case à cocher *Control* activée), et le second raccourci *Maj+F1* (case à cocher *Shift* activée).



Vous n'aurez pas besoin de rechercher ces identificateurs de touche virtuelle si vous utilisez le mode *Key Value* pour insérer la touche (voir la section "*Mode Key Value*" plus loin dans ce chapitre), car l'éditeur *Accelerator* prend la valeur correcte et l'insère à votre place.

Lancement de l'éditeur Accelerator

La procédure de lancement de l'éditeur Accelerator est différente selon que vous créez une nouvelle table de raccourcis ou modifiez une table existante.

Pour créer une nouvelle table de raccourcis

Vous pouvez créer une nouvelle table de raccourcis dans un nouveau projet ou dans un projet existant.

Pour lancer l'éditeur Accelerator et créer une nouvelle table de raccourcis :

Voir le chapitre 14 pour plus d'informations sur l'ouverture d'un projet.

1. Choisissez File | New Project pour créer un nouveau projet ou File | Open Project pour ouvrir un projet existant.
2. Choisissez Resource | New pour afficher la boîte de dialogue New Resource.
3. Dans la boîte de dialogue New Resource, double-cliquez sur ACCELERATOR dans la liste Resource Type.

Resource Workshop affiche l'éditeur Accelerator avec un modèle de table de raccourcis que vous pouvez éditer.

Pour éditer une table de raccourcis existante

Pour lancer l'éditeur Accelerator et éditer une ressource raccourci existante, ouvrez le projet dans lequel la ressource est mémorisée et exécutez une des procédures suivantes :

- Double-cliquez sur le nom de la ressource raccourci dans la fenêtre projet.
- Mettez en surbrillance le nom de la ressource raccourci dans la fenêtre projet, et choisissez Resource | Edit.

Resource Workshop affiche l'éditeur Accelerator avec la table de raccourcis chargée.

Exécution de l'éditeur Menu en même temps

Lorsque vous travaillez avec des raccourcis, il est préférable de lancer l'éditeur Menu et de charger le menu contenant les commandes associées. De cette façon, vous pouvez voir le texte de commande et les identificateurs d'élément dont vous aurez besoin lors de la définition des raccourcis. Comme expliqué à la section "Paramétrage de la valeur de commande" plus loin dans ce chapitre, chaque raccourci doit avoir un identificateur qui correspond à une commande du menu.

Pour de plus amples informations sur l'ajout de raccourcis à vos menus, reportez-vous aux sections "Saisie du texte de l'élément" et "Ajout de commandes au menu Widgets" du chapitre 16.

Comment utiliser l'éditeur Accelerator

L'écran de l'éditeur Accelerator est divisé en deux volets : le volet Outline et le volet Attribute. Vous pouvez vous déplacer d'un volet à un autre avec la souris ou la touche *F6*.

Volet Outline

Le volet Outline affiche, sous la forme script, tous les raccourcis définis dans la table. La ligne du haut de ce volet correspond au nom de la table de raccourcis. Les lignes au-dessous sont les entrées de raccourcis, comprenant deux parties :

- La première partie identifie la touche utilisée en tant que raccourci. Il s'agit soit d'une touche ASCII, soit d'une touche virtuelle.
- La seconde partie est l'identificateur d'élément de la commande associée au raccourci. Cet identificateur est soit un entier, soit un nom.

Pour sélectionner un raccourci dans le volet Outline, utilisez la souris ou les touches fléchées (vous pouvez également afficher un TurboMenu : pointez sur le volet Outline et cliquez sur le bouton droit de la souris, ou appuyez sur *Alt+F10* n'importe où dans le volet).

Voir la section "Combinaisons de touches de la table des raccourcis" pour plus de détails sur les touches ASCII et virtuelles.

Volet Attribute

En sélectionnant un raccourci du volet Outline, vous pouvez lire ses paramètres dans le volet Attribute. Lorsqu'un raccourci est sélectionné, vous pouvez effectuer des changements dans le volet Attribute, tels que spécifier une nouvelle combinaison de touches, ou associer le raccourci à une autre commande.

A partir du volet Attribute, utilisez la souris ou appuyez sur *Ctrl+↑* ou *Ctrl+↓* pour sélectionner un raccourci.

Utilisez la souris pour vous déplacer à l'intérieur du volet Attribute et y faire des sélections. En outre, vous pouvez utiliser les touches suivantes :

- *Tab* effectue un déplacement vers l'avant dans le volet Attribute, et *Maj+Tab* vers l'arrière. Notez que chaque case à cocher Modifiers est un taquet de tabulation.



Si vous tabulez à partir de la boîte de saisie Command vers la boîte de saisie Key, l'éditeur Accelerator passe au mode Key Value. Dans ce mode, vous pouvez appuyer sur toute touche pour entrer le raccourci. Pour sortir de ce mode, cliquez sur la souris ou appuyez sur *Alt+Echap*. Le mode Key Value est décrit à la section "Mode Key Value" de ce chapitre.

- Les touches fléchées effectuent une sélection parmi les boutons radio Key Type, et la *barre d'espace* bascule l'état des cases à cocher Modifiers.

Vos sélections sont validées lorsque vous appuyez sur *Entrée* (pour modifier le raccourci) ou *Inser* (pour créer un nouveau raccourci), ou encore lorsque vous vous déplacez jusqu'à un autre raccourci dans le volet Outline.

Le tableau suivant décrit les sélections que vous pouvez faire dans le volet Attribute.

Tableau 17.1
Sélections dans le
volet Attribute de
l'éditeur Accelerator

Sélection	Description
Command	Identificateur d'élément (entier ou nom) pour la commande exécutée par le raccourci. Cette valeur doit correspondre à la valeur dans la ressource menu associée.
Key	Touche de raccourci. Vous pouvez entrer la touche en mode manuel (avec des guillemets pour les touches ASCII et la syntaxe appropriée pour les touches virtuelles), ou en mode Key Value (l'éditeur Accelerator décide s'il s'agit d'une touche ASCII ou d'une touche virtuelle, et la rentre à votre place dans le format approprié). Si vous tabulez dans cette boîte de saisie à partir de la boîte de saisie Command, vous passez automatiquement en mode Key Value. Reportez-vous à la section "Mode Key Value" de ce chapitre pour obtenir la description du mode Key Value.
Key Type	Touche ASCII ou virtuelle. En mode Key Value, l'éditeur Accelerator définit automatiquement ces boutons radio.
- ASCII	Le raccourci utilise une touche ASCII (voir la section "Touches ASCII" de ce chapitre).
- Virtual Key	Le raccourci utilise une touche virtuelle (voir la section "Touches virtuelles" de ce chapitre).

Tableau 17.1
Sélections dans le
volet Attribute de
l'éditeur Accelerator
(suite)

Sélection	Description
Modifieurs	Les descriptions suivantes de ces cases à cocher détaillent l'option si elle est activée.
- Alt	Le raccourci inclut la touche <i>Alt</i> (par exemple, <i>Alt+W</i>).
- Shift	Le raccourci inclut la touche <i>Maj</i> (par exemple, <i>Maj+F1</i>).
- Ctrl	Le raccourci inclut la touche <i>Ctrl</i> (par exemple, <i>Ctrl+F3</i>).
- Invert Menu Item	Si vous utilisez le raccourci, la commande de la barre menu associée clignote (inversion momentanée).

Edition d'une table de raccourcis

Une fois la table de raccourcis chargée dans l'éditeur Accelerator, vous pouvez l'éditer. Grâce à cet éditeur, vous pouvez définir et modifier les raccourcis, ou spécifier une combinaison de touches de raccourci en vous contentant d'appuyer sur la combinaison souhaitée. Vous pouvez aussi copier ou supprimer tout raccourci dans la table, et vérifier s'il existe des valeurs d'identificateur en double.

Ajout d'un raccourci

Pour ajouter un nouveau raccourci dans la table de raccourcis, appuyez sur *Inser* ou choisissez Accelerator | New Item. La nouvelle touche apparaît dans le volet Outline sous la ligne actuellement sélectionnée, avec comme valeurs par défaut : 0 (zéro) pour la valeur de touche et une valeur entière unique pour l'identificateur de commande.

Lorsque vous ajoutez un nouveau raccourci, l'édition bascule dans le volet Attribute.

Sélection d'un raccourci

Pour sélectionner un raccourci, exécutez une des procédures suivantes :

- Appuyez sur *Ctrl+↑* ou *Ctrl+↓* pour mettre en surbrillance le raccourci dans le volet Outline, et basculer automatiquement l'édition vers le volet Attribute.
- Cliquez sur le raccourci dans le volet Outline, puis appuyez sur *F6* pour basculer automatiquement l'édition vers le volet Attribute.
- Si le pointeur se trouve déjà dans le volet Outline, utilisez les touches fléchées pour sélectionner le raccourci, puis appuyez sur *F6* pour basculer automatiquement l'édition vers le volet Attribute.

Comment utiliser le volet Attribute

Le volet Attribute a des boîtes de saisie, des boutons radio et des cases à cocher qui définissent le raccourci. Les options du volet Attribute sont décrites dans le tableau 17.1.

Paramétrage de la valeur de commande

Dans la boîte de saisie Command, tapez l'identificateur d'élément (un entier ou un nom) pour la commande exécutée par le raccourci.

La chaîne Command est automatiquement mise en surbrillance lorsque vous sélectionnez un raccourci avec *Ctrl+↑* ou *Ctrl+↓*, ou ajoutez un raccourci avec *Ins*. Tapez l'identificateur d'élément directement dans la boîte de saisie.

Si vous liez le raccourci à une commande d'un menu existant, lancez l'éditeur Menu et chargez la ressource menu. Notez l'identificateur d'élément de la commande et utilisez ce même identificateur dans la boîte de saisie Command du raccourci.

Notez les points suivants à propos des identificateurs :

- Si vous entrez un nom d'identificateur existant et que vous voyez la boîte de dialogue "Create a new identifier:", vous avez probablement commis une erreur. Cliquez sur No et vérifiez le nom.
- Si vous entrez volontairement un nouvel identificateur pour ajouter ultérieurement l'élément de menu associé au menu, Resource Workshop vous invite à créer un nouvel identificateur. Cliquez Yes ou appuyez sur *Entrée*, puis entrez une valeur d'identificateur unique dans la boîte de dialogue New Identifier.

Si vous n'obtenez pas la boîte de dialogue "Create a new identifier", l'identificateur existe déjà. Entrez un identificateur unique avant de poursuivre.

Spécification du raccourci

Pour spécifier la combinaison de touches, entrez la combinaison dans la boîte de saisie Key.

Votre raccourci doit être compatible avec les raccourcis des autres applications Windows. N'utilisez donc pas une combinaison de touches Windows (comme *Ctrl+Echap*). Pour obtenir des conseils sur le choix des combinaisons de touches appropriées, reportez-vous au *Systems Application Architecture Common User Access Advanced Interface Design Guide* de IBM.

Vous pouvez utiliser le mode manuel ou le mode Key Value pour entrer la touche.

Mode Key Value : En mode Key Value, toute touche ou combinaison de touches frappée est automatiquement entrée dans la boîte de saisie Key comme raccourci. L'éditeur Accelerator détermine si la touche est une touche ASCII ou une touche virtuelle, et sélectionne le bouton radio Key Type approprié. L'éditeur Accelerator active aussi les cases à cocher Modifiers appropriées.

Mode Manual : En mode Manual, vous fournissez toutes les informations qui définissent le raccourci. Vous devez déterminer si la touche est une touche ASCII ou une touche virtuelle. S'il s'agit d'une touche virtuelle, vous devez connaître l'identificateur Windows et le taper en lettres majuscules. Vous devez aussi sélectionner le bouton radio Key Type approprié (ASCII ou Virtual Key), et activer la combinaison appropriée des cases à cocher *Alt*, *Shift* et *Ctrl*.

Fonction Flash

La fonction *Flash* fait clignoter une commande barre menu lorsque l'utilisateur appuie sur le raccourci d'une commande associée à la commande de la barre menu.

Par exemple, si vous avez sélectionné un bloc de données dans plusieurs applications Windows et si vous appuyez sur *Maj+Suppr* (l'équivalent de la sélection Edit | Cut), Windows inverse temporairement le clignotement de la commande Edit de la barre menu. Cette fonction permet à l'utilisateur de connaître le menu associé au raccourci.

La fonction Flash (également appelée *élément menu inversé*) est activée par défaut lorsque vous créez un raccourci. Vous pouvez l'inhiber en désactivant l'option Invert Menu Item dans le volet Attribute de l'éditeur Accelerator.

Recherche des combinaisons de touches redondantes

Pour vérifier si vous n'utilisez pas la même combinaison de touches plusieurs fois, vous pouvez déboguer une table de raccourcis en recherchant les combinaisons en double comme suit :

1. Avec une table de raccourcis ouverte, choisissez Accelerator | Check Dup Keys.

2. Si deux raccourcis ont la même combinaison de touches, l'éditeur Accelerator affiche le message "Duplicate key value found" et met en surbrillance le second raccourci. Effectuez les modifications et continuez à déboguer votre table de raccourcis avec Check Dup Keys jusqu'à ce que le message "No duplicate key values found" s'affiche.

Création d'un exemple de table de raccourcis

Dans cette section, vous allez créer une table de raccourcis pour le menu Widgets, décrit à la section "Exemple de menu" du chapitre 16. Sans Resource Workshop, utilisez un éditeur de texte ou un traitement de texte pour créer le script ressource de la section précédente.



Si vous n'aviez pas enregistré le menu exemple dans le chapitre 16, vous pouvez quand même travailler avec cette section. Dans le cas contraire, plusieurs étapes vous proposent des instructions supplémentaires ou des alternatives. Pour créer une table de raccourcis avec l'éditeur Accelerator de Resource Workshop, procédez comme suit :

Le chapitre 14 décrit la procédure d'ouverture d'un projet.

1. Ouvrez un fichier projet ou, si vous aviez enregistré le menu exemple dans le chapitre 16, ouvrez le projet contenant votre menu Widgets.
2. Choisissez Resource | New.
3. Dans la boîte de dialogue New Resource, double-cliquez le type de ressource ACCELERATOR. Vous apercevez l'éditeur Accelerator avec une nouvelle entrée.

Si vous aviez enregistré le menu exemple du chapitre 16, ouvrez l'éditeur Menu en double-cliquant sur son nom dans la fenêtre projet.

Redimensionnez les fenêtres pour les éditeurs Menu et Accelerator afin de les visualiser simultanément.

Dans la fenêtre Outline de l'éditeur Menu, mettez en surbrillance l'élément de menu List et notez son identificateur (*wmmu_List*) et son raccourci (*Ctrl+L*).

Cliquez sur le nouveau raccourci dans l'éditeur Accelerator. Si nécessaire, appuyez sur *F6* pour mettre en surbrillance la boîte de saisie Command.

4. Dans la boîte de saisie Command, entrez le nom de l'identificateur pour la première commande du menu (*wmmu_List*).

5. Déplacez-vous par tabulations jusqu'à la boîte de saisie Key.
Notez que vous êtes en mode Key Value. Appuyez sur *Ctrl+L*. L'éditeur Accelerator entre la valeur ASCII "*^L*" et sélectionne le bouton radio ASCII pour vous.
6. Appuyez sur *Alt+Echap* pour sortir du mode Key Value, puis sur *Entrée* pour valider ces paramètres sur la touche raccourci en surbrillance dans le volet Outline.
Si vous ne travaillez pas avec un projet existant, Resource Workshop vous invite à créer un identificateur pour ce raccourci. Cliquez sur *Yes*, puis sur *OK* dans la boîte de dialogue New Identifier pour accepter la valeur par défaut.
7. Appuyez sur *Inser* pour créer un nouveau raccourci.
Si vous travaillez avec le menu Widgets du chapitre 5, cliquez sur l'éditeur Menu. Sélectionnez la commande suivante avec une touche raccourci, et notez son identificateur d'élément et sa touche de raccourci.
8. Ajoutez les raccourcis restants : *wmnu_Add*, *wmnu_Asc* et *wmnu_Desc* en substituant la combinaison de touches et l'identificateur appropriés, jusqu'à ce que tous les raccourcis soient définis.
9. Enregistrez le projet.

La table de raccourcis est maintenant terminée. Dans les étapes suivantes, vous allez volontairement créer un doublon (valeur de touche) afin de déboguer votre table de raccourcis au moyen de l'éditeur Accelerator.

1. Mettez en surbrillance le second raccourci et changez sa valeur de touche de *Ctrl+A* à *Ctrl+L*.
Appuyez sur *Ctrl+↑* pour mettre en surbrillance le raccourci, *Tab* pour activer le mode Key Value.
2. Tapez *Ctrl+L*, appuyez sur *Alt+Esc* et sur *Entrée* pour valider la modification.
3. Choisissez Accelerator | Check Dup Keys. Resource Workshop affiche la boîte de dialogue "Duplicate key value found". Appuyez sur *Entrée* pour la fermer.
4. Mettez maintenant en surbrillance *wmnu_Asc* ou *wmnu_Desc* dans le volet Outline et appuyez sur *Suppr*.
5. Affichez le menu Edit. Notez que le choix Undo Delete Item se réfère à votre suppression du dernier raccourci.

6. Choisissez Undo Delete Item pour restaurer le raccourci supprimé.
7. Affichez à nouveau le menu Edit. La commande Undo indique maintenant Undo Change Item. Lorsque vous choisissez cette commande, le raccourci de la commande Add change de *Ctrl+L* à *Ctrl+A*.
8. Choisissez à nouveau Accelerator | Check Dup Keys. Vous devriez obtenir le message "No duplicate key values found".

Cet exemple vous montre combien il est facile de créer des raccourcis en utilisant à la fois l'éditeur Menu et l'éditeur Accelerator. Vous pouvez passer alternativement d'un éditeur à un autre pour connaître les associations entre raccourcis et commandes menus. Vous pouvez aussi utiliser le mode Key Value pour entrer les touches, tout en laissant l'éditeur Accelerator travailler à votre place. Ensuite, vous pouvez vérifier si vous n'avez pas créé de touches en double. Dans l'affirmative, vous pourrez très facilement les modifier à la fois dans le menu et dans la table des raccourcis.

Création d'une table de chaînes

Une *table de chaînes* contient les messages d'erreur, les messages d'invite ou toute autre chaîne de texte que votre application devra afficher. Vous pouvez mémoriser plusieurs tables de chaînes dans votre fichier projet. En principe, vous allez définir une table de chaînes spécifique pour chaque regroupement logique de votre programme (voir la section "Comment travailler avec des tables de chaînes" plus loin dans ce chapitre).

Si vous définissez les chaînes de texte comme ressources séparées, il vous sera plus facile d'éditer du texte sans modifier votre code source. Par exemple, si vous traduisez une application Windows dans une langue étrangère, les tables de chaînes vous simplifieront le travail. Néanmoins, vous devrez traduire le texte d'autres ressources, comme les boîtes de dialogue.

Travailler avec des tables de chaînes implique quatre tâches essentielles :

1. Lancer l'éditeur String.
2. Créer et éditer des tables de chaînes.
3. Enregistrer la table de chaînes.
4. Compiler la ressource dans un fichier exécutable et tester la table de chaînes.

A la section "Création d'un exemple de table de chaînes" de ce chapitre, vous trouverez un petit exemple concernant la création et l'édition d'une table de chaînes.

Lancement de l'éditeur String

La procédure de lancement de l'éditeur String dépend de la tâche réalisée : création d'une nouvelle table de chaînes ou édition d'une table existante.

Création d'une nouvelle table de chaînes

Suivez la procédure ci-dessous pour lancer l'éditeur String afin de créer une nouvelle table de chaînes :

1. Ouvrez le projet auquel vous souhaitez ajouter la table.
2. Choisissez Resource | New. Resource Workshop affiche la boîte de dialogue New Resource.
3. Faites défiler le contenu de la boîte liste Resource Type et double-cliquez sur STRINGTABLE.

Resource Workshop ouvre l'éditeur String et place une référence à la nouvelle table de chaînes dans votre fenêtre projet.

Edition d'une table de chaînes existante

Suivez la procédure ci-dessous pour lancer l'éditeur String afin d'éditer une table de chaînes existante :

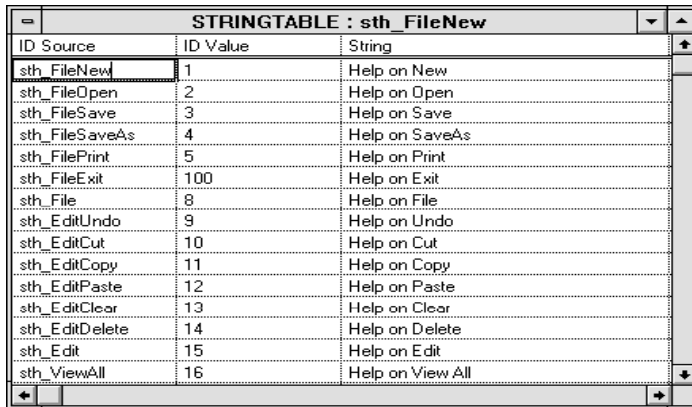
1. Ouvrez le projet contenant la table de chaînes à éditer.
2. Trouvez la table de chaînes dans la fenêtre projet.
3. Double-cliquez sur l'entrée de la table de chaînes, ou sélectionnez-la et choisissez Resource | Edit.

La table de chaînes sélectionnée apparaît dans l'éditeur String.

Comment travailler avec des tables de chaînes

Si vous ouvrez l'éditeur String, une table de chaînes apparaît. Si vous créez une nouvelle table de chaînes, vous voyez une entrée unique avec le texte générique "String". Si vous éditez une table de chaînes existante, vous voyez les entrées de chaînes, comme ci-dessous :

Figure 18.1
Editeur String avec
des entrées de table
de chaînes



ID Source	ID Value	String
sth_FileNew	1	Help on New
sth_FileOpen	2	Help on Open
sth_FileSave	3	Help on Save
sth_FileSaveAs	4	Help on SaveAs
sth_FilePrint	5	Help on Print
sth_FileExit	100	Help on Exit
sth_File	8	Help on File
sth_EditUndo	9	Help on Undo
sth_EditCut	10	Help on Cut
sth_EditCopy	11	Help on Copy
sth_EditPaste	12	Help on Paste
sth_EditClear	13	Help on Clear
sth_EditDelete	14	Help on Delete
sth_Edit	15	Help on Edit
sth_ViewAll	16	Help on View All

Chaque entrée requiert une valeur dans les colonnes suivantes : ID Source, ID Value et String.

- ID Source contient le nombre entier de la chaîne. Si vous affectez un nom d'identificateur, il apparaît dans cette colonne. Sinon, vous apercevez un nombre entier.
- ID Value contient toujours l'identificateur entier de la chaîne.
- Une chaîne, d'une longueur maximale de 255 caractères, est stockée dans la colonne String.

Vous pouvez cliquer sur un élément avec le bouton droit de la souris pour afficher un TurboMenu qui vous permet d'exécuter rapidement les tâches qui s'y rapportent (vous pouvez aussi sélectionner l'élément et appuyer sur *Alt+F10*). La section suivante décrit comment Windows gère les valeurs d'identificateur de chaînes, et vous suggère une méthode permettant une meilleure utilisation de la mémoire.

Chaque chaîne d'une table doit disposer d'un identificateur entier unique. Windows regroupe les chaînes en segments de 16 chaînes chacun. Les chaînes avec les identificateurs de 0 à 15 constituent le premier segment, les chaînes 16 à 31 le deuxième, etc. Lorsque vous compilez vos ressources, les chaînes s'ajoutent au fichier exécutable en segments, chargés en mémoire au moment de l'exécution.

Si votre application requiert une chaîne spécifique, Windows charge un segment de chaînes entier en mémoire. Si vous savez comment affecter les identificateurs de chaînes, vous pouvez réduire la quantité de mémoire requise par votre application.

Supposons que vous définissiez 32 chaînes pour votre application. Si vous affectez les identificateurs 0 à 31 à ces chaînes, votre fichier exécutable contient deux segments de 16 chaînes. Chaque fois que votre application requiert une chaîne et charge un segment, elle charge probablement plusieurs chaînes inutiles.

Pour une meilleure utilisation de la mémoire, regroupez vos chaînes logiquement. Par exemple, une partie de votre application a besoin de cinq chaînes, et une seconde partie requiert neuf chaînes. Si vous numérotez le premier groupe 0 à 4, et le second groupe 16 à 24, vous créez deux segments, le premier avec onze identificateurs inutilisés et le second avec sept identificateurs inutilisés. Chaque identificateur inutilisé occupe uniquement un octet en mémoire, comparé à l'espace très important occupé par les chaînes dont vous n'avez pas besoin. En organisant vos chaînes de cette façon, Windows charge les chaînes associées sans charger les chaînes inutiles.

Pour plus de détails, reportez-vous à voir la section "Comment utiliser les identificateurs" du chapitre 14.

Si vous indiquez un identificateur de chaîne unique, vous pouvez utiliser un nombre entier ou un identificateur alphanumérique (**#define** dans C ou C++, ou déclaration constante en Pascal) qui représente un nombre entier. Si vous utilisez des identificateurs alphanumériques pour vous rappeler plus facilement des identificateurs de chaînes, mémorisez-les dans un fichier identificateur (fichier en-tête pour C et C++; fichier unité ou fichier inclus pour Pascal). Vérifiez si ces fichiers existent avant d'ajouter des identificateurs à partir de l'éditeur String.

Entrée d'une nouvelle chaîne

Pour revenir au paramètre initial de ID Source, appuyez sur *Echap* avant *Tab* ou *Entrée*.

Pour revenir au paramètre initial du champ String, appuyez sur *Echap* avant *Tab* ou *Entrée*.



Pour entrer une nouvelle chaîne dans une table :

- Si votre table est nouvelle, entrez simplement les informations de la chaîne, comme décrit aux étapes 3, 4 et 5 ci-après.
- Si vous ajoutez une chaîne dans une table, commencez à l'étape 1 ci-après.

1. Sélectionnez la chaîne au-dessus de la ligne où vous souhaitez ajouter la nouvelle chaîne.
2. Appuyez sur *Inser* ou choisissez Stringtable | New Item.
3. Vous pouvez accepter le numéro que l'éditeur String place dans ce champ, ou vous pouvez taper un identificateur entier ou alphanumérique pour ID Source.

Si vous tapez une valeur entière, l'éditeur String entre automatiquement cette valeur dans le champ ID Value.

Si vous tapez une valeur alphanumérique, Resource Workshop vérifie si cette valeur existe déjà. Dans ce cas, l'éditeur String insère la valeur entière d'identificateur dans le champ ID Value lors du déplacement du pointeur à l'aide de la touche *Tab* vers le champ texte String. Si la valeur alphanumérique n'existe pas, l'éditeur String affiche la boîte de dialogue "Create a new identifier".

Pour plus de détails, reportez-vous au paragraphe "Comment utiliser les identificateurs" du chapitre 14.

4. Appuyez sur *Tab* ou cliquez dans la boîte au-dessous de String, et tapez la chaîne de texte.

Chaque chaîne contient 255 caractères maximum et peut aussi contenir toute séquence d'échappement de type C, y compris : \n (nouvelle ligne), \t (tab), \r (retour chariot), \ (barre oblique inverse), \" (guillemets).

Si le compilateur Resource Workshop rencontre une séquence d'échappement de type C dans une entrée de chaîne, il génère la valeur hexadécimale ASCII correspondante dans le code objet. C'est votre programme qui interprétera la valeur correctement. Par exemple, si le compilateur analyse `\b\040\x7F`, il génère la séquence hexadécimale `07207F`. Votre code peut interpréter cette séquence comme les caractères ASCII BEL, SPC et DEL, ou encore affecter une autre signification à ces valeurs hexadécimales.

5. Appuyez sur *Entrée* (pour accepter la nouvelle valeur) ou *Inser* (pour accepter la valeur et en insérer une nouvelle).

Edition de chaînes existantes

L'éditeur String vous permet de modifier facilement des chaînes. Pour sélectionner une chaîne avec la souris, cliquez sur la chaîne à éditer. Avec le clavier, appuyez sur *Tab*, ↑ ou ↓ pour vous déplacer dans la table. Placez le curseur sur la chaîne à éditer.

Modification d'une chaîne

Vous pouvez effacer les valeurs des champs ID Source et String d'une chaîne et taper de nouvelles valeurs. Vous ne pouvez pas modifier directement les valeurs qui s'affichent dans le champ ID Value. L'éditeur String actualise la valeur suivant les informations saisies dans le champ ID Source.

Edition du script ressource d'une table de chaînes

Vous pouvez utiliser l'éditeur de texte interne pour éditer le script ressource d'une table de chaînes. Pour cela, sélectionnez la table de chaînes dans la fenêtre projet et choisissez Resource | Edit As Text.

Le texte du script ressource apparaît, prêt pour l'édition.

Modification de chaînes

Pour éditer une chaîne, procédez comme suit :

1. Recherchez la chaîne à éditer et apportez les modifications requises. Modifiez uniquement le texte qui s'affiche entre guillemets.
2. Pour compiler votre saisie et rester dans l'éditeur String, choisissez Compile | Compile Now.

Notez que le menu Compile est uniquement disponible dans l'éditeur de texte.

3. Si vous voulez quitter l'éditeur String, choisissez la commande Close dans la boîte Control-menu de la fenêtre de l'éditeur de texte.

Resource Workshop vous invite à compiler. Lorsque vous cliquez sur Yes, il compile le menu et quitte la fenêtre projet.

En cas d'erreur syntaxique, Resource Workshop revient dans l'éditeur de texte pour que vous puissiez corriger l'erreur.

Création d'un exemple de table de chaînes

L'exemple qui suit crée quelques chaînes que Resource Workshop utilise pour décrire les options de menus.

Sans Resource Workshop, vous utiliseriez le script ressource ci-après pour créer ces chaînes :

Ces chaînes devraient apparaître sur la ligne d'état Resource Workshop.

```
STRINGTABLE
BEGIN
    MENU_FILE, "Create, open, or close files"
    MI_FILENEW, "Create a new project, resource, or file"
    MI_FILEOPEN, "Open a resource file"
    MI_FILESAVE, "Save this resource file"
END
```

La chaîne alphanumérique en majuscules qui précède chaque chaîne est un identificateur unique pour la chaîne. Comme toutes les ressources Windows, chaque chaîne doit être désignée par un nombre entier. Sans Resource Workshop, vous devriez définir séparément les valeurs entières de tous ces identificateurs dans un fichier en-tête (programme C) ou dans un fichier inclus ou une unité (programme Pascal).

Nous allons vous expliquer comment créer ces chaînes exemple avec l'éditeur String de Resource Workshop :

1. Vérifiez si vous avez déjà ouvert un projet. Si vous avez déjà travaillé sur des exemples en utilisant MYPROJ.RC, vous pouvez ouvrir ce projet.
2. Si vous ne possédez pas déjà un fichier identificateur (fichier en-tête pour les **#define** du C, unité ou fichier inclus pour les constantes Pascal) pour le projet, créez-le maintenant et appelez-le MYPROJ.H ou MYPROJ.PAS.
3. Choisissez Resource | New et double-cliquez STRINGTABLE pour lancer l'éditeur String.
4. Faites des retours arrière sur le numéro de la boîte de saisie au-dessous de ID Source et tapez l'identificateur de la chaîne. Pour la première chaîne, c'est MENU_FILE.
5. Appuyez sur *Tab* pour aller jusqu'à String.

Avant que l'éditeur String vous laisse passer au champ String, il recherche une valeur entière d'identificateur pour la chaîne courante. Tout d'abord, il vérifie ce que vous avez tapé au-dessous de ID Source. Si vous avez entré un nombre entier, l'éditeur String place ce même nombre au-dessous de ID Value et vous laisse passer au champ String.

Pour plus de détails, reportez-vous à la section "Comment utiliser les identificateurs" du chapitre 14.

Pour plus de détails, reportez-vous à la section "Comment utiliser les identificateurs" du chapitre 14.

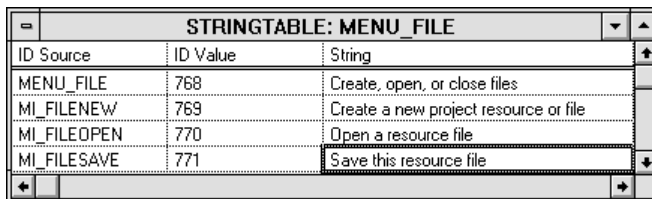
Puisque vous avez entré un identificateur alphanumérique (MENU_FILE dans ce cas), l'éditeur String recherche une déclaration de constante C **#define** ou Pascal qui désigne le nombre entier. Cette déclaration n'existant pas, il vous demande si vous souhaitez créer un nouvel identificateur.

6. Cliquez sur Yes pour appeler la boîte de dialogue New Identifier.
7. Faites défiler la liste File jusqu'à MYPROJ.H (ou MYPROJ.PAS), puis double-cliquez sur ce fichier pour le sélectionner comme le fichier identificateur dans lequel le nouvel identificateur sera écrit.
8. Entrez une valeur entière d'identificateur dans la boîte de saisie Value. Tapez 768 pour le premier identificateur.
9. Appuyez sur *Entrée* ou cliquez sur OK pour accepter le nouvel identificateur et le placer dans MYPROJ.H ou MYPROJ.PAS.
10. Puisque vous en avez terminé avec l'identificateur, vous revenez au champ String. Tapez le texte de la chaîne, c'est-à-dire `Create, open, or close files` pour la première chaîne.
11. Pour définir la chaîne suivante, appuyez sur *Insér* ou choisissez Stringtable | New Stringtable Item.

Répétez les étapes 4 à 11 pour définir les trois autres chaînes représentées au début de la section : MI_FILENEW, MI_FILEOPEN et MI_FILESAVE (voir la figure 18.2).

Vous avez remarqué que pour chaque nouvelle chaîne, l'éditeur String incrémente de 1 la dernière valeur entière d'identificateur. Vous n'avez pas besoin de sélectionner ce nombre. L'éditeur String le met simplement ici à votre disposition.

Une fois les quatre chaînes créées, votre table de chaînes s'affiche comme suit :



ID Source	ID Value	String
MENU_FILE	768	Create, open, or close files
MI_FILENEW	769	Create a new project resource or file
MI_FILEOPEN	770	Open a resource file
MI_FILESAVE	771	Save this resource file

Figure 18.2
Editeur String avec
quatre chaînes
définies

En dernière étape, fermez la table de chaînes en choisissant Close dans le menu Control de la fenêtre de l'éditeur String. L'éditeur String donne à la nouvelle table le nom du premier identificateur de la table. Si la première entrée ID Source est un numéro, il devient le nom de la table de chaînes. Pour modifier le nom d'une table de chaînes, vous devez donc modifier la première entrée ID Source de la table.



Cette convention sur les noms est logique. En effet, les chaînes sont chargées par segments de 16 chaînes chacun. En outre, la valeur entière de l'identificateur d'une chaîne indique l'endroit où la chaîne apparaît dans un segment. Celle de la première chaîne de la table indique l'endroit où la table commence dans un segment.

Comment utiliser l'éditeur Bitmap

L'éditeur Bitmap de Resource Workshop est l'outil utilisé pour créer ou éditer une ressource bitmap, y compris les ressources bitmap standard :

- Icônes
- Curseurs
- Graphiques par points
- Polices

Les fonctions spécifiques aux différents types de ressources sont décrites dans les chapitres 20 à 22.

Ce chapitre décrit les fonctions et les outils de l'éditeur Bitmap identiques à tous les types de ressources bitmap. Les chapitres relatifs aux ressources spécifiques décrivent les fonctions et les outils de l'éditeur Bitmap propres à chaque type de ressource.

Lancement de l'éditeur Bitmap

Resource Workshop lance automatiquement l'éditeur Bitmap lorsque vous créez une nouvelle ressource bitmap, ou éditez une ressource bitmap existante. Les étapes spécifiques au lancement de l'éditeur Bitmap sont fournies dans les chapitres relatifs aux ressources individuelles.

Si vous créez une ressource bitmap alors que le Presse-papiers de Windows en contient déjà une, les dimensions de celle-ci apparaissent dans la boîte de dialogue New Bitmap attributes ; si le Presse-papiers ne contient pas de ressource bitmap, les dimensions par défaut (largeur et hauteur) sont de 64 pixels.

Couleurs de premier plan et de fond, et pixels

Voir la section "Utilisation des couleurs de premier plan et de fond" de ce chapitre pour plus d'informations sur les couleurs de premier plan et de fond.

Les images bitmap dessinées avec l'éditeur Bitmap sont créées sur une grille de "points" approximativement carrés, appelés *pixels*. Vous créez l'image en paramétrant chaque pixel comme une couleur de premier plan ou de fond. Les pixels s'assemblent comme une mosaïque pour former l'image bitmap.

Puisque les pixels cohabitent sur un plan unique, il n'y a aucune distinction entre le premier plan et le fond. En termes simples, la couleur de premier plan est la couleur que vous sélectionnez et dessinez avec le bouton *gauche* de la souris; et la couleur de fond est la couleur que vous sélectionnez et dessinez avec le bouton *droit* de la souris.

Vous pouvez utiliser une variété de couleurs de premier plan et de fond pour les caractéristiques de votre dessin (lignes, boîtes, ombre, etc.), aussi bien pour le "fond" de l'image (qui est, après tout, une illusion, étant donnée la nature bi-dimensionnel de l'image). La possibilité d'affecter des couleurs aux deux boutons de la souris signifie que vous avez deux couleurs à votre disposition à tout moment.



Il existe une différence importante entre la couleur de premier plan et de fond. Lorsque vous effacez ou déplacez un bloc de pixels dans votre image (voir la section "Outil rectangle de sélection" plus loin dans ce chapitre), la couleur de fond actuellement sélectionnée remplace la couleur dans les pixels qui ne sont plus occupés par le bloc.

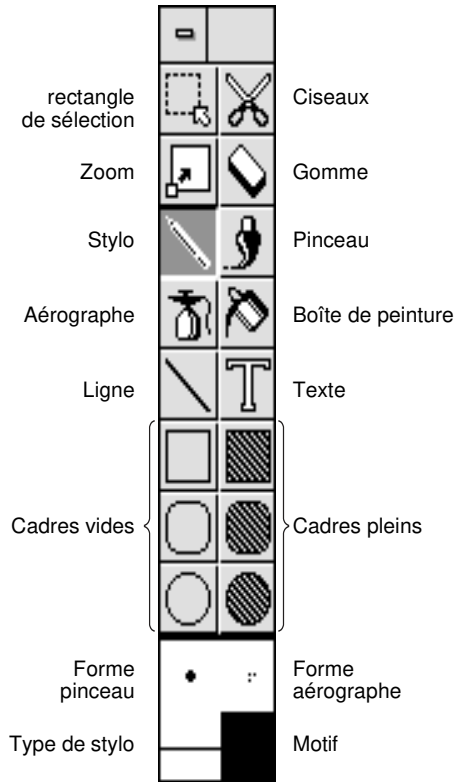
Si vous utilisez l'outil Gomme, les boutons sont inversés : le bouton gauche génère la couleur de fond, et le bouton droit génère la couleur de premier plan.

La couleur de premier plan courante est indiquée par *FG* sur la palette Colors, et la couleur de fond courante par *BG*. Si vous sélectionnez la même couleur pour le premier plan et le fond, le carré de la palette Colors contient *FB*.

Utilisation de la palette Tools (Outils)

Lorsque vous ouvrez une ressource dans l'éditeur Bitmap, la palette Tools est située dans l'angle supérieur droit de la fenêtre d'édition. Utilisez la palette Tools pour choisir l'outil de l'éditeur Bitmap avec lequel travailler. Au bas de la palette Tools figurent quatre sélections de style concernant les formes de pinceaux, les motifs de remplissage et les largeurs de ligne.

Figure 19.1
Palette Tools de
l'éditeur Bitmap



La palette Tools est similaire à une fenêtre : vous pouvez la déplacer, la fermer et l'ouvrir.

La plupart des outils vous permettent de dessiner soit avec la couleur de premier plan, soit avec la couleur de fond.

Si la palette Colors indique FB, la même couleur est sélectionnée pour le premier plan et le fond

- Utilisez le bouton gauche de la souris pour dessiner avec la couleur de premier plan (marquée FG dans la palette Colors).
- Utilisez le bouton droit pour la couleur de fond (signalée par BG dans la palette Colors).

Outil Rectangle de sélection



Choisissez Edit | Select
All pour sélectionner
toute l'image

L'outil Rectangle de sélection sélectionne une zone rectangulaire de votre image, en vue de sa copie, son déplacement ou son effacement. Pour sélectionner une zone, placez le bout du pointeur sur un angle du rectangle, et faites glisser le pointeur sur l'angle diagonalement opposé. Lorsque le contour clignotant entoure la zone voulue, relâchez le bouton de la souris. Pour désélectionner la zone, cliquez hors du contour clignotant, ou bien appuyez sur *Entrée* ou *Echap*.

Lorsque vous sélectionnez une zone, vous pouvez utiliser les commandes du menu Edit pour couper, copier, effacer, dupliquer, ou coller la zone sélectionnée, ou encore utiliser la souris pour déplacer ou dupliquer la zone.

Outil Ciseaux



Fondamentalement, l'outil Ciseaux réalise la même fonction que l'outil Rectangle de sélection : la sélection d'une zone d'image. Cependant, avec les ciseaux, vous pouvez sélectionner et déplacer des zones de toute forme, et non plus simplement des rectangles.

Pour sélectionner une zone, faites glisser les ciseaux jusqu'à ce que le contour entoure la zone voulue, puis relâchez le bouton de la souris. La zone sélectionnée est indiquée par l'affichage d'un motif en clignotement. Vous pouvez couper, copier, effacer, dupliquer ou déplacer la zone sélectionnée, comme avec l'outil Rectangle de sélection (procédure décrite dans la section précédente).

Outil Zoom



L'éditeur Bitmap utilise
le centre de l'image
comme point de
référence lors d'un zoom
sur toute l'image.

Vous pouvez utiliser l'outil Zoom pour zoomer en avant ou en arrière sur l'image entière, ou encore pour zoomer sur une zone sélectionnée.

Pour zoomer en avant sur toute l'image, double-cliquez sur l'icône Zoom de la palette Tools. Resource Workshop effectue un Zoom jusqu'au grossissement supérieur : 400%, 800% ou 1600%. Vous pouvez aussi choisir View | Zoom In pour exécuter la même fonction sur la fenêtre actuellement sélectionnée.

Lorsque vous effectuez un zoom en avant sur l'image, utilisez l'outil Main (voir la section "Outil Main" plus loin dans ce chapitre) ou les barres de défilement pour vous déplacer dans l'image zoomée.

Pour zoomer en arrière sur toute l'image, maintenez enfoncée la touche *Maj*, puis double-cliquez sur l'icône Zoom. Pour zoomer sur une partie de l'image, sélectionnez la zone voulue en faisant glisser le rectangle de l'outil Zoom, puis relâchez le bouton de la souris. Resource Workshop effectue un zoom arrière jusqu'au grossissement inférieur : 800%, 400% ou 100%. Vous pouvez aussi choisir View | Zoom Out pour exécuter la même fonction.

Tableau 19.1
Commandes de zoom

Commande de zoom	Raccourci	Opération de la souris sur l'icône Zoom
Zoom avant	<i>Ctrl+Z</i>	Double-clic
Zoom arrière	<i>Ctrl+O</i>	Maj.+double-clic
Taille normale	<i>Ctrl+A</i>	—

Pour pouvoir contrôler l'image point par point, vous pouvez afficher une grille de pixels portant sur l'image zoomée en cours. Pour cela, choisissez Options | Editor Options et cochez la case Grid on Zoomed Window. Chaque carré de la grille correspond à un pixel.



Lorsque vous travaillez avec deux volets de fenêtre (voir la section "Utilisation des deux volets de la fenêtre" plus loin dans ce chapitre), l'action de zoomer affecte uniquement la fenêtre courante (active).

Outil Gomme



Si la mention *FB* apparaît dans la palette Colors, la même couleur est sélectionnée comme couleur de fond et de premier plan en cours.

Vous pouvez utiliser l'outil Gomme pour effacer toute l'image, ou bien comme outil de dessin. Les affectations de couleurs aux boutons de souris pour l'outil Gomme sont inversées par rapport à celles des autres outils de dessin.

- Si vous double-cliquez sur la gomme dans la palette Tools, toute l'image est remplacée par la couleur de fond en cours (*BG* sur la palette Colors).
- Si vous faites glisser le pointeur à l'aide du bouton gauche de la souris, la gomme dessine une ligne de 1 pixel de large avec la couleur de fond en cours.
- Si vous faites glisser le pointeur à l'aide du bouton droit de la souris, la gomme dessine une ligne de 1 pixel de large avec la couleur de premier plan en cours (*FG* sur la palette Colors).

Avant d'utiliser la gomme, vous pouvez vérifier les couleurs en cours dans la palette Colors.

Outil Stylo



L'outil Stylo dessine des lignes au format libre au moyen du type de stylo en cours affiché dans la palette Tools (voir la figure 19.1). Pour dessiner avec l'outil Stylo, appuyez sur un bouton de la souris, et faites glisser le stylo dans votre image. Lorsque vous avez fini de dessiner, relâchez le bouton de la souris. Pour dessiner des lignes droites, utilisez l'outil Ligne à la place de l'outil Stylo.

Outil Pinceau



L'outil Pinceau dessine des motifs libres en utilisant le style de pinceau et le motif en cours affichés dans la palette Colors (voir la figure 19.1). Pour dessiner, faites glisser le pinceau dans votre image. Lorsque vous avez terminé, relâchez le bouton de la souris.

Lorsque vous sélectionnez le pinceau, le curseur prend la forme de pinceau en cours. La zone dessinée par le pinceau est toujours proportionnelle à la taille du cadre de l'image. En d'autres termes, si le pinceau représente la moitié de la largeur de l'image et que vous zoomez sur cette image, le pinceau représentera toujours la moitié de l'image zoomée.

Avant d'utiliser le pinceau, vous pouvez spécifier la forme, le motif et les couleurs.

Outil Aérographe



L'outil Aérographe dessine des motifs libres en utilisant le style de l'aérographe et le motif en cours affichés dans la palette Colors (voir la figure 19.1). Pour utiliser l'aérographe, exécutez une des deux procédures suivantes :

- Vous pouvez le faire glisser à travers l'image. L'aérographe est différent du pinceau. En effet, si vous le faites glisser doucement, il dessine un motif épais. Par contre, si vous le faites glisser rapidement, il dessine un motif plus fin et dispersé.
- Vous pouvez le cliquer de façon répétitive, comme si vous appuyiez sur le diffuseur d'une bombe de peinture.



Lorsque vous sélectionnez l'aérographe, le curseur prend la forme de pinceau en cours. La zone dessinée par l'aérographe est toujours proportionnelle à la taille du cadre de l'image. En d'autres termes, si l'aérographe représente la moitié de la largeur de l'image et que vous zoomez sur cette image, l'aérographe représentera toujours la moitié de l'image zoomée.

Avant d'utiliser l'aérographe, vous pouvez spécifier la forme, le motif et les couleurs.

Outil Boîte de peinture



L'outil Boîte de peinture remplit une zone de votre image avec la couleur sélectionnée. Pour utiliser la boîte de peinture, placez le viseur dans la partie de l'image que vous voulez remplir, puis cliquez sur un bouton de la souris. La boîte de peinture remplace la couleur située sous le curseur par la couleur sélectionnée, et remplit autour de ce point jusqu'à ce qu'elle rencontre une couleur différente.

Par exemple, si la couleur sélectionnée est rouge et que vous cliquez sur un carré bleu, tout le bleu autour de ce point sera remplacé par du rouge. La boîte de peinture ne remplacera pas les autres couleurs. Si le carré bleu fait partie d'une zone rectangulaire bleue entièrement délimitée par du vert, seul le rectangle bleu passera en rouge.



Si vous cliquez sur une zone qui n'est pas entièrement délimitée par d'autres couleurs, la couleur s'étendra sur toutes les autres parties de l'image de même couleur que la zone d'origine.



A cause des problèmes inhérents aux pilotes vidéo, le remplissage total d'une image bitmap avec la boîte de peinture ne fonctionne pas toujours correctement. Pour résoudre ce problème, Resource Workshop offre une alternative à l'algorithme de remplissage standard, plus efficace mais plus lente. Pour valider cet algorithme pour l'un des éditeurs de ressources bitmap, ajoutez la ligne suivante dans la section [RWS_Icon] de WORKSHOP.INI :

```
RWS_OwnFloodFill=1
```

L'exemple suivant illustre la section RWS_Icon modifiée :

```
[RWS_Icon]
RWS_OwnFloodFill=1
PercentLeft=69
ZoomLeft=8
ZoomRight=1
bVert=1
```

Outil Ligne

Pour les lignes libres,
utilisez le stylo.



L'outil Ligne dessine les lignes droites. Appuyez sur le bouton de la souris, et faites glisser l'outil Ligne dans votre image. Lorsque le dessin de la ligne est terminé, relâchez le bouton de la souris.

Pour contraindre les lignes à des incréments de 45 degrés (horizontalement, verticalement ou en diagonale), maintenez enfoncée la touche *Maj* lorsque vous dessinez.

Avant d'utiliser l'outil Ligne, vous pouvez spécifier le style et les couleurs en cours.

Outil Texte



Pour ajouter du texte à votre image, choisissez l'outil Texte, et cliquez sur l'endroit où votre texte doit débiter. Un curseur clignotant vous invite à saisir le texte.

Pour spécifier comment et où le texte est affiché, vous pouvez :

- Utiliser Text | Font pour spécifier la police, la taille et le style du texte.
- Utiliser les commandes Text | Align pour spécifier l'alignement du texte.

Pour plus d'informations sur les commandes du menu Text, reportez-vous à la section "Ajout de texte à une ressource" plus loin dans ce chapitre.

Il est inutile de choisir des commandes du menu Text avant d'entrer le texte ; vous pouvez aussi les choisir immédiatement après la saisie (avant de cliquer à nouveau). Par exemple, si vous notez pendant la saisie que le texte est trop grand pour s'adapter à l'image, vous pouvez arrêter la saisie et choisir la commande Font pour diminuer la taille du texte.

Le texte est toujours affiché dans la couleur de premier plan en cours. Avant de saisir le texte, vous pouvez spécifier la couleur de premier plan, en cliquant le bouton gauche de la souris sur la couleur souhaitée dans la palette Colors. Comme pour la police et la taille, vous pouvez modifier la couleur de texte en cours, sous réserve de le faire immédiatement après la saisie.

Dessin de cadres vides



Trois outils vous permettent de dessiner des cadres vides en utilisant le style et la couleur de ligne en cours : le rectangle, le rectangle arrondi et l'ellipse.

Pour dessiner un cadre vide, sélectionnez l'outil voulu et amenez un cadre dans l'image. Placez le viseur du curseur sur un angle du cadre et faites-le glisser sur l'angle opposé. Relâchez le bouton de la souris lorsque le cadre vous convient. Appuyez sur *Maj* pour dessiner un carré ou un cercle.

Avant de dessiner un cadre, vous pouvez spécifier l'épaisseur de trait et la couleur.

Dessin de cadres pleins



Trois outils vous permettent de dessiner des cadres pleins dans votre image : le rectangle plein, le rectangle arrondi plein et l'ellipse pleine.

Pour dessiner un cadre plein, sélectionnez l'outil voulu et amenez un cadre dans l'image. Placez le viseur du curseur sur un angle du cadre et faites-le glisser sur l'angle opposé. Relâchez le bouton de la souris lorsque le cadre vous convient.

Ces outils utilisent le style de ligne en cours pour le contour. Spécifiez une largeur de stylo nulle si vous ne souhaitez pas que Resource Workshop dessine un contour autour du motif plein.

Le motif en cours est affiché dans l'angle inférieur droit de la palette Tools. Si vous faites glisser le pointeur à l'aide du bouton gauche de la souris, le contour et le motif prennent la couleur de premier plan en cours (FG), et le remplissage du motif prend la couleur de fond en cours (BG). Si vous faites glisser le pointeur à l'aide du bouton droit de la souris, les couleurs sont inversées.

La boîte de dialogue Set Pattern comprend aussi deux motifs pleins, un noir et un blanc.

- Si vous sélectionnez le motif noir, le bouton gauche de la souris génère un remplissage total avec la couleur de premier plan, et un pourtour avec la couleur de fond. Comme d'habitude, le bouton droit de la souris génère un effet inverse.
- Si vous sélectionnez le motif blanc, le bouton gauche de la souris génère un remplissage total avec la couleur de fond, et un pourtour avec la couleur de premier plan. A nouveau, le bouton droit de la souris génère un effet inverse.

Avant de dessiner un cadre plein, vous pouvez spécifier le style de ligne, la couleur et le motif.

Outil Main

Vous pouvez aussi utiliser les barres de défilement pour déplacer l'image.



Parfois, lorsque vous affichez une image zoomée, elle ne s'adapte pas entièrement à l'écran. Vous pouvez utiliser l'outil Main pour déplacer l'image et en voir d'autres parties. A la différence des autres outils, l'outil Main n'est pas compris dans la palette Tools. Mais vous pouvez temporairement changer un outil (excepté l'outil Texte) en main, en maintenant enfoncée la touche *Ctrl*. A l'aide de la main, vous pouvez attraper l'image et la faire glisser dans la direction voulue.

Sélections de style



Au bas de la palette Tools figure une boîte qui affiche (dans le sens des aiguilles d'une montre, en partant de l'angle supérieur gauche) la forme du pinceau, la forme de l'aérographe, le motif en cours et le style de ligne.

Vous pouvez cliquer chaque style que vous voulez changer, ou encore utiliser les commandes de menu pour choisir les styles. Pour plus d'informations sur le choix des styles, voir les sections "Choix des formes de pinceau", "Choix des motifs de dessin", et "Choix d'un style de ligne" plus loin dans ce chapitre.

Utilisation des deux volets de la fenêtre

Dans l'éditeur Bitmap, vous pouvez obtenir deux vues différentes de l'image créée ou éditée. Vous pouvez diviser la fenêtre verticalement ou horizontalement pour afficher deux vues côte à côte, ou bien une vue au-dessus de l'autre. Vous pouvez aussi choisir la façon de zoomer sur chaque vue.

Pour diviser la fenêtre, choisissez View | Split Horizontal ou View | Split Vertical.

Lorsque la fenêtre est divisée, un des volets est *actif*. Le volet actif est celui dans lequel vous travaillez. Pour activer un volet, cliquez avec la souris dans ce volet.

Pour privilégier la vue d'une image par rapport à l'autre, déplacez le curseur sur la ligne divisant les images (la barre séparatrice). Lorsque le curseur devient une double flèche, faites glisser cette barre séparatrice. Par exemple, les fenêtres étant divisées verticalement, faites glisser la barre séparatrice vers la droite pour mieux voir une image zoomée.

Pour retirer la division d'une fenêtre et retourner à une seule vue, faites glisser la barre séparatrice complètement vers la gauche ou vers la droite (division verticale), vers le haut ou vers le bas (division horizontale).

Lecture de la ligne d'état

La ligne d'état située au bas de la fenêtre de l'éditeur Bitmap est divisée en deux parties. Le côté droit fournit des informations sur l'outil de dessin, et le côté gauche affiche des informations sur les commandes menus de l'éditeur Bitmap.

Lorsque vous cliquez sur un menu, ou utilisez des touches de raccourcis pour choisir une commande menu, le côté gauche de la ligne d'état de l'éditeur Bitmap affiche des informations supplémentaires concernant la commande en surbrillance.

Le côté droit de la ligne d'état indique l'outil de dessin utilisé, et son emplacement sur l'écran. Selon l'outil, vous voyez aussi des informations de couleurs.

Le message d'état d'outil affiché dépend de l'outil sélectionné et de son emplacement sur l'écran. Voici deux exemples de messages, avec les explications afférentes.

Line x: 18 y:32

Ce message indique que vous avez sélectionné l'outil Ligne, et qu'il occupe les coordonnées en pixels 18,32. Les coordonnées en pixels sont comptées à partir de l'angle supérieur gauche de l'image. Lorsque vous vous déplacez vers le côté droit de l'image, la valeur X augmente ; lorsque vous vous déplacez vers le bas de l'image, la valeur Y augmente.

Brush x:20 y:37 R:128 G: 0 B:0 Palette Index: 1

Ce message indique que vous avez sélectionné le pinceau, et qu'il occupe les coordonnées en pixels 20,37. Les valeurs R, G et B indiquent les valeurs rouge, vert, et bleu de la couleur située à ces coordonnées. La couleur de l'image située aux coordonnées 20,37 est la couleur 1 de l'index de la palette Colors, qui comporte une valeur de 128 en composante de rouge, 0 en composante de vert et 0 en composante de bleu.

Vous pouvez aussi voir l'index de palette et les valeurs RGB de la couleur à l'emplacement du curseur, si vous sélectionnez le stylo, l'aérographe ou la boîte de peinture.

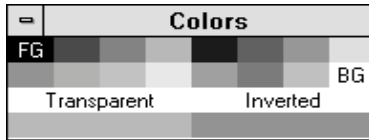
Vous pouvez obtenir les numéros d'index de palette et les valeurs RGB en double-cliquant sur une couleur dans la palette Colors.

Voir la section "Personnalisation des couleurs", pour plus d'informations sur les valeurs RGB et les index de palette

Comment travailler avec les couleurs

Pour choisir les couleurs désirées lorsque vous éditez une ressource dans l'éditeur Bitmap, utilisez la palette Colors. Vous pouvez travailler avec la palette Colors même si votre image est en noir et blanc, et vous pouvez cacher ou afficher la palette Colors à tout moment.

Figure 19.2
Palette Colors à 16
couleurs



Vous pouvez utiliser la palette Colors pour :

- Choisir une couleur de premier plan.
- Choisir une couleur de fond.
- Choisir des zones transparentes et inversées (ressources icône et curseur uniquement).

Choix du nombre de couleurs d'une ressource

Lorsque vous créez un nouveau graphique par points ou une nouvelle icône, Resource Workshop affiche une boîte de dialogue pour vous permettre de choisir le nombre de couleurs que vous voulez inclure dans votre ressource. Lorsque vous éditez un graphique par points ou une icône, vous pouvez changer le nombre de couleurs de l'image avec la commande Size and Attributes. Cette commande est située dans le menu Bitmap ou le menu Icon, selon le type de ressource éditée.

Certains pilotes Windows 3.1 haute résolution nécessitent plus de 512 ko de mémoire.

Pour les ressources bitmap et icône, vous pouvez inclure jusqu'à 256 couleurs dans votre ressource. Le nombre de couleurs disponibles (apparaissant dans la palette Colors) dépend du type de pilote d'affichage utilisé sous Windows.

Utilisation des couleurs de premier plan et de fond

Si la mention *FB* apparaît dans la palette Colors, la même couleur est sélectionnée pour le premier plan et le fond.



Pour utiliser une couleur de premier plan :

1. Cliquez avec le bouton *gauche* de la souris sur la couleur voulue dans la palette Colors. Les lettres *FG* apparaissent sur cette couleur.
2. Sélectionnez un outil de dessin et cliquez ou faites glisser le pointeur à l'aide du bouton gauche de la souris pour dessiner avec la couleur de premier plan.

Pour utiliser une couleur de fond :

1. Cliquez avec le bouton *droit* de la souris sur la couleur voulue dans la palette Colors. Les lettres *BG* apparaissent sur cette couleur.
2. Sélectionnez un outil de dessin et cliquez ou faites glisser le pointeur à l'aide du bouton droit de la souris pour dessiner avec la couleur de fond.

La gomme opère de façon inverse aux outils de dessin. En la faisant glisser avec le bouton gauche de la souris, vous produisez la couleur de fond, et avec le bouton droit de la souris, vous produisez la couleur de premier plan.

Zones couleur transparentes et inversées

L'idée d'une zone couleur *transparente* ou *inversée* est propre aux ressources icône et curseur.

- Une zone transparente "s'efface" au moment de l'exécution, permettant à la couleur située derrière l'icône ou le curseur de transparaître. Cette caractéristique est particulièrement utile dans les curseurs, où vous n'utiliserez généralement pas la zone image entière pour le curseur lui-même.
- L'emploi d'une couleur inversée dans votre icône ou curseur "inverse" la couleur sur laquelle vous placez l'icône ou le curseur au moment de l'exécution.
Par exemple, si vous créez un viseur de curseur à partir d'une couleur inversée, le viseur apparaîtra au moment de l'exécution dans la couleur inverse de la zone du bureau au-dessous de lui. Si vous placez le curseur au-dessus d'une zone de bureau noire, le viseur apparaîtra en blanc. Si vous le déplacez au-dessus d'une zone rouge, il apparaîtra en bleu sardes.

L'utilisation des zones couleur transparente et inversée est décrite en détail dans le chapitre 20, "Création d'icônes", et le chapitre 21, "Création de curseurs" de ce manuel.

Définition des couleurs transparentes et inversées

Les couleurs transparentes et inversées n'apparaissent pas dans votre icône ou curseur au moment de l'exécution. Elles sont remplacées par la couleur du bureau situé en-dessous ou par son inverse. Les couleurs que vous définissez avec l'état Transparent ou Inversé doivent être des couleurs que vous n'utilisez pas dans votre icône ou curseur.

La couleur transparente par défaut est la couleur du bureau en cours définie dans la palette de couleurs du panneau de configuration Windows. Si le bureau utilise une couleur tramée, la couleur transparente par défaut est la couleur unie la plus proche que Resource Workshop peut fournir. Si vous disposez d'un périphérique à 256 couleurs, cette restriction ne s'applique pas; la couleur transparente correspondra toujours à la couleur du bureau.

Vous pouvez changer la couleur transparente en une couleur différente du bureau, mais elle revient toujours à la couleur du plan de travail à chaque lancement de Resource Workshop. Néanmoins, les régions désignées comme transparentes, en utilisant la couleur par défaut ou une couleur de votre choix, restent transparentes et prennent la couleur transparente en cours. Notez cependant que vous pouvez changer les couleurs de telle sorte que des régions transparentes et des régions non transparentes utilisent la même couleur.

Pour changer les couleurs que l'éditeur Bitmap affiche pour les zones transparentes et inversées, exécutez une des procédures suivantes :

- Dans la palette Colors, double-cliquez sur la barre sous Transparent ou Inverted.
- Sélectionnez Transparent ou Inverted comme couleur de premier plan ou de fond. Choisissez ensuite Icon | Edit Foreground Color ou Icon | Edit Background Color.

Resource Workshop affiche la boîte de dialogue Set Transparent Color.

Cette boîte de dialogue fonctionne comme la boîte de dialogue Edit Colors à une exception près : elle change à la fois les couleurs transparente et inversée.

La boîte de dialogue est toujours Set Transparent Color, que vous partiez de la barre de couleurs Transparent ou de la barre de couleurs Inverted.

Remarquez les trois boîtes couleur en haut de la boîte de dialogue. Lorsque vous ajustez les valeurs RGB (Red, Green, Blue), la couleur réelle créée par ces pourcentages RGB est affichée dans la boîte Requested. La boîte Granted affiche la couleur disponible la plus proche (pour les périphériques à 256 couleurs ou plus, ce sera la couleur Requested), et la boîte Inverse montre automatiquement son inverse.

La couleur Granted est affectée à la barre couleur Transparent, et la couleur Inverse à la barre couleur Inverted.

**Comment afficher
et cacher la palette
Colors**

Pour cacher la palette Colors, fermez-la en double-cliquant sur l'icône du menu système, dans l'angle supérieur gauche de la palette.

Vous pouvez aussi cacher la palette Colors en choisissant la commande Hide Palette. Le nom du menu où vous trouverez cette commande dépend du type de ressource en cours d'édition. Si vous éditez une icône, la commande Hide Palette se trouve dans le menu Icon. Si vous éditez un curseur, le menu se trouve à la même position sur la barre menu, mais est appelé le menu Cursor.

Une fois que vous avez caché la palette, le nom de la commande devient Show Palette pour vous permettre d'afficher à nouveau la palette.

Personnalisation des couleurs

Vous ne pouvez pas
éditer les couleurs dans
les curseurs ou les
polices.

Si vous éditez un graphique par points ou une icône en couleur, vous pouvez modifier la palette Colors pour inclure toute couleur supportée par votre pilote d'affichage. Cette opération est inutile si la palette Colors comprend déjà toutes les couleurs gérées par votre ordinateur. Mais si votre pilote d'affichage est capable de restituer 256 couleurs, et que vous travaillez avec une image en 16 couleurs, vous pouvez inclure n'importe laquelle des 256 couleurs dans la palette en 16 couleurs.

Les sections suivantes décrivent comment éditer une des couleurs dans la palette Colors, y compris les couleurs transparentes et inversées.

Edition de couleurs dans la palette Colors

Pour sélectionner une couleur pour l'éditer, exécutez une des procédures suivantes :

Double-cliquez sur la couleur dans la palette Colors.

- Sélectionnez-la comme couleur de premier plan ou de fond, puis choisissez Edit Foreground Color ou Edit Background Color à partir du menu Icon ou Bitmap.

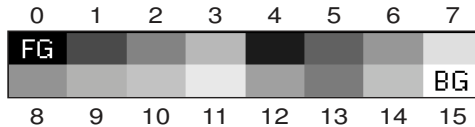
Resource Workshop affiche la boîte de dialogue Edit Color.

Index de palette

Pour vous aider à identifier l'emplacement de la couleur sélectionnée dans la palette Colors, la boîte de dialogue Edit Color comprend une valeur d'index de palette. Chaque case de la palette Colors porte un numéro d'index, en commençant par zéro à l'angle supérieur gauche et en comptant de la gauche vers la droite sur chaque ligne. Des valeurs d'index ne sont pas affectées aux barres Transparent et Inverted dans la palette Colors.

Par exemple, dans la palette de 16 couleurs par défaut, les couleurs sont numérotées conformément à la figure suivante :

Figure 19.3
Index de la palette de 16 couleurs



Lorsque l'outil sélectionné est le stylo, le pinceau, l'aérographe ou la boîte de peinture, la partie droite de la ligne d'état affiche l'index de palette et les valeurs RGB pour la couleur située sous la position du curseur dans la fenêtre d'édition.

Edition d'une couleur

Sur un périphérique capable d'afficher 256 couleurs (ou plus), la distinction Requested/Granted ne s'applique pas. Vous voyez des couleurs vraies, et non des couleurs tramées.

Pour éditer une couleur, vous pouvez modifier ses valeurs RGB, soit en tapant de nouvelles valeurs dans la colonne gauche, soit en utilisant les barres de défilement, sur le côté droit de la boîte de dialogue. Resource Workshop affiche la couleur la plus proche correspondant aux nouvelles valeurs RGB, dans la boîte de couleur Requested. Pour un périphérique de 16 couleurs, une couleur tramée peut apparaître dans cette boîte. Dans la boîte de couleur Requested, Resource Workshop affiche la couleur disponible la plus proche, selon les limites du périphérique en cours. Vous ne pouvez voir cette boîte droite changer de couleur que si la correspondance la plus proche est différente de la couleur en cours.

Lorsque vous avez terminé de modifier les couleurs, cliquez sur OK ou appuyez sur *Entrée* pour placer la nouvelle couleur dans la palette Colors.



Avant de quitter l'éditeur Bitmap, vous devez désactiver l'option Save With Default Device Colors, pour permettre à Resource Workshop d'enregistrer votre palette personnalisée. Choisissez Options | Editor Options, et vérifiez qu'aucune marque de sélection n'apparaît à côté de Save With Default Device Colors.

Le bouton Default extrait de la palette par défaut (objet standard Windows) la couleur portant le même numéro que l'index de palette (affiché dans le haut de la boîte de dialogue).

Le bouton System extrait de la palette système la couleur portant le même numéro que l'index de palette (figurant dans le haut de la boîte de dialogue). Ce bouton est désactivé pour les affichages VGA standard (qui ne gèrent pas les palettes logiques), mais est activé pour les périphériques qui supportent 256 couleurs ou plus.

Ajout de texte à une ressource



Vous pouvez ajouter du texte à toute ressource créée avec l'éditeur Bitmap. Par exemple, vous voudrez peut-être ajouter du texte à une icône, pour représenter le nom du programme.

Pour ajouter du texte à une ressource, cliquez sur l'outil Text (ci-contre), puis cliquez dans l'image à l'emplacement où débutera le texte. Resource Workshop affiche un curseur texte clignotant. Enfin, tapez le texte souhaité.

Pour définir l'alignement, la police, la taille et le style (gras, italique ou souligné) par défaut, choisissez Text | Align ou Text | Font avant de taper. Pour changer l'alignement, la police, la taille et le style de votre texte, choisissez Text | Align ou Text | Font immédiatement avant de cliquer la souris à nouveau.

De plus, immédiatement après la saisie du texte (et avant de cliquer la souris à nouveau), vous pouvez en modifier la couleur, en sélectionnant une nouvelle couleur dans la palette Colors.

Vous pouvez uniquement entrer du texte ou changer sa couleur avec le bouton gauche de la souris.



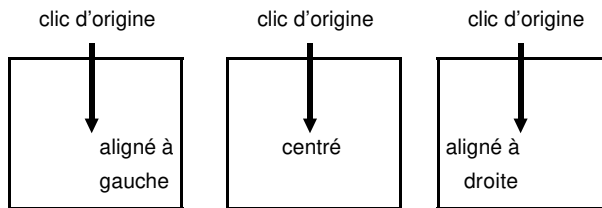
Si vous cliquez sur un autre outil, ou dans une autre zone de l'image, après la saisie du texte, vous ne pouvez rien modifier dans le texte que vous venez d'entrer. A ce stade, le texte devient simplement une autre partie du bitmap, comme si vous l'aviez dessiné.

Alignement du texte

Pour aligner du texte, utilisez les commandes Text | Align.

Les commandes Align contrôlent l'endroit où le texte est affiché, par rapport au point sur lequel vous avez cliqué avec l'outil Text (le point d'insertion). Vous pouvez utiliser une commande Align avant de taper du texte ou immédiatement après.

Figure 19.4
Alignement de texte



Vous ne pouvez modifier l'alignement du texte que vous venez d'entrer si vous avez cliqué pour effectuer une autre sélection.

Choix des polices, de la taille et du style de texte



Pour choisir l'aspect d'un texte, utilisez la commande Text | Font, soit avant d'entrer le texte, soit immédiatement après la saisie.

Vous ne pouvez modifier la police du texte si vous avez cliqué pour effectuer une autre sélection.

Lorsque vous avez choisi la commande Font, Resource Workshop affiche la boîte de dialogue Select Font.

Vous pouvez choisir la police, la taille et le style de texte souhaités. Par exemple, choisissez Arial comme police, 12 points pour la taille et gras italique comme style.

Notez les caractères affichés au bas de la boîte de dialogue Select Font. Ils changent pour indiquer la police, le style et la taille en cours que vous avez sélectionné.

Choix des formes de pinceaux

Dans une ressource, vous pouvez dessiner des images en utilisant le pinceau ou l'aérographe. Resource Workshop vous permet de choisir leur forme.

La forme du pinceau et de l'aérographe en cours est toujours affichée au bas de la palette Tools (voir la section "Sélections de style" plus haut dans ce chapitre).

Choisissez une forme de pinceau ou d'aérographe en suivant une des méthodes ci-dessous :

- Pour une forme de pinceau, utilisez Options | Brush Shape.
- Pour une forme d'aérographe, utilisez Options | Airbrush Shape.
- Cliquez sur la zone représentant la forme de pinceau ou d'aérographe en cours.

Une boîte de dialogue apparaît, et vous indique toutes les formes de pinceaux disponibles. Dans le haut de la boîte de dialogue, Resource Workshop vous signale la forme actuellement sélectionnée.

A la prochaine utilisation du pinceau ou de l'aérographe, Resource Workshop utilisera la forme spécifiée. Cette forme restera la même jusqu'à la spécification suivante.

Choix des motifs de dessin

Vous pouvez choisir un motif à dessiner sur votre image, en utilisant un des outils suivants :

- Le pinceau
- L'aérographe
- Le rectangle plein
- Le rectangle arrondi plein
- L'ellipse pleine

Notez que la plupart des motifs sont constitués de points ou de lignes noires sur un "champ" blanc. Si vous dessinez avec le bouton gauche de la souris, les points ou les lignes prennent la couleur de premier plan en cours, et le champ prend la couleur de fond en cours. Si vous dessinez avec le bouton droit de la souris, l'effet est inversé : les points ou les lignes prennent la couleur de fond en cours, et le champ prend la couleur de premier plan en cours.

L'angle inférieur droit de la palette Tools affiche toujours le motif en cours.

Vous pouvez choisir un motif avec les méthodes suivantes :

- Choisissez Options | Pattern.
- Cliquez sur la zone représentant le motif en cours.

Une boîte de dialogue apparaît, et vous indique tous les motifs disponibles. Au sommet de la boîte de dialogue, Resource Workshop vous signale le motif actuellement sélectionné. Pour sélectionner un nouveau motif, cliquez sur un des motifs de la boîte de dialogue, puis cliquez sur OK.

A la prochaine utilisation d'un outil capable de restituer un motif, Resource Workshop utilise le motif spécifié. Ce motif reste le même jusqu'à la spécification suivante.

Choix d'un style de ligne

Vous pouvez contrôler le style de ligne produit au moyen d'un des outils suivants :

- Le stylo
- L'outil Ligne
- Le rectangle vide
- Le rectangle arrondi vide
- L'ellipse vide

Vous pouvez choisir un style de ligne avec les méthodes suivantes :

- Choisissez Options | Pen Style.
- Cliquez sur la sélection de style de ligne, dans la palette Tools.

Une boîte de dialogue apparaît, et vous indique tous les styles disponibles. Dans le haut de la boîte de dialogue, Resource Workshop vous signale le style actuellement sélectionné.

Notez le choix "null" de largeur de stylo. Utilisez-le pour dessiner un rectangle plein, un rectangle arrondi ou une ellipse sans bordure.

A la prochaine utilisation d'un outil capable de dessiner une ligne, Resource Workshop utilise l'épaisseur spécifiée. L'épaisseur de stylo choisie reste la même jusqu'à la prochaine spécification.

Alignement d'une zone sélectionnée

Vous pouvez aligner une zone sélectionnée d'une image sur le sommet, le bas, les côtés ou le centre de la fenêtre d'édition en cours. L'alignement de la zone sélectionnée la déplace sur l'emplacement spécifié, comme si vous aviez sélectionné la zone, puis déplacé avec la souris. Pour aligner une zone sélectionnée, choisissez Options | Align, puis les options voulues dans la boîte de dialogue Align Selection.

Choisissez les options sous le mot Vertical, pour aligner la zone sélectionnée sur l'axe vertical de la fenêtre d'édition, et les options sous le mot Horizontal pour l'aligner sur l'axe horizontal.

Par exemple, si vous choisissez Top et Left Side et que vous cliquez sur OK, la zone sélectionnée se déplace sur l'angle supérieur gauche de la fenêtre. Si vous choisissez Vertical | Center et Right Side et cliquez sur OK, la zone sélectionnée est centrée entre le sommet et le bas de la fenêtre, et elle est déplacée aussi loin vers la droite que possible.

L'éditeur Bitmap utilise la couleur de fond en cours pour peindre l'emplacement qu'occupait la zone sélectionnée, comme il l'aurait fait si vous aviez déplacé la zone sélectionnée avec la souris.

Redimensionnement d'une zone sélectionnée

Vous pouvez redimensionner ou déplacer une zone d'image sélectionnée avec l'outil Pick Rectangle, en choisissant Options | Size. Si vous redimensionnez la zone, l'éditeur Bitmap dilate ou comprime en conséquence l'image de la zone de sélection.

Lorsque vous choisissez Options | Size, l'éditeur Bitmap affiche la boîte de dialogue Stretch Selection.

Les valeurs Top et Left de cette boîte de dialogue sont les coordonnées en pixels de l'angle supérieur gauche de la zone actuellement sélectionnée, et les autres valeurs représentent sa largeur et sa hauteur en cours (en pixels). Lorsque la boîte de dialogue s'ouvre, les mêmes valeurs apparaissent à la fois dans les colonnes Old Position | Size et New Position | Size.

Pour déplacer ou redimensionner la zone (ou les deux), reprenez des valeurs en pixels qui tombent dans les limites du cadre en cours. Entrez-les dans les boîtes sous la rubrique New Position | Size. Appuyez sur *Tab* pour passer d'un champ à l'autre.

Par exemple, pour déplacer une zone rectangulaire sur l'angle supérieur gauche, et lui donner une largeur de 30 pixels et une hauteur de 5 pixels, procédez comme suit :

1. A l'aide de l'outil Rectangle de sélection, définissez la zone à déplacer ou à redimensionner.
2. Entrez 0 comme valeur pour Left, puis appuyez sur *Tab*.
3. Entrez 0 comme valeur pour Top, et appuyez sur *Tab*.
4. Entrez 30 comme largeur, et appuyez sur *Tab*.
5. Entrez 5 comme hauteur.
6. Cliquez sur OK, ou appuyez sur *Entrée*.

La boîte est maintenant positionnée dans l'angle supérieur gauche du cadre, avec les nouvelles largeur et hauteur. De plus, l'éditeur Bitmap remplit toute zone ouverte située auparavant derrière la boîte avec la couleur de fond en cours, comme il l'aurait fait si vous aviez déplacé le rectangle de sélection avec la souris.

Choix des options globales de l'éditeur Bitmap

Pour définir les options globales de l'éditeur Bitmap, choisissez Options | Editor Options pour afficher la boîte de dialogue Set Paint Editor Options.

Draw on both images

Lorsque deux vues d'une même image sont affichées dans l'éditeur Bitmap, vous pouvez choisir que Resource Workshop actualise chaque image pendant que vous dessinez.

Si vous ne choisissez pas cette option, Resource Workshop n'actualise pas l'autre image tant que vous n'avez pas terminé de dessiner un élément. Par exemple, lorsque vous faites glisser un pinceau dans l'image, Resource Workshop n'affiche la ligne que sur l'image réellement dessinée. Cependant, lorsque vous relâchez le bouton de la souris, Resource Workshop actualise alors l'autre image.

Grid on zoomed windows

Sur des fenêtres zoomées, vous pouvez afficher une grille qui indique comment l'image est dessinée, pixel par pixel.

Save with default device colors

Désactivez cette option si vous voulez enregistrer une palette de couleurs personnalisée. Lorsque cette option est activée, toute couleur personnalisée dans la palette Colors revient à la couleur par défaut, lorsque vous refermez l'éditeur Bitmap.

Vous ne pouvez enregistrer des palettes de couleurs personnalisées que si votre pilote d'affichage peut afficher 256 couleurs, et s'il accepte les palettes logiques. Pour plus de détails, reportez-vous à la section "Personnalisation des couleurs", plus haut dans ce chapitre.

Création d'icônes

Les *icônes* sont des petites images bitmap, de taille 64×64 , 32×32 ou 32×16 pixels. En général, les programmes Windows ont recours à des icônes spéciales pour représenter des fenêtres réduites.

Pour concevoir vos icônes, utilisez l'éditeur Bitmap de Resource Workshop.

L'éditeur Bitmap comprend divers outils de dessin, et une palette Colors d'emploi facile pour sélectionner les couleurs. Il vous permet aussi de zoomer votre image, et affiche plusieurs vues de l'icône créée. Le chapitre 19 explique comment utiliser l'éditeur Bitmap.

Travailler avec des icônes implique quatre étapes principales :

1. Lancer l'éditeur Bitmap.
Si vous créez une nouvelle ressource icône, l'éditeur Bitmap vous présente une grille icône vide. Si vous modifiez une icône existante, elle apparaît dans l'éditeur Bitmap.
2. Créer ou modifier l'icône.
3. Tester l'icône.
4. Enregistrer l'icône.

Création d'une nouvelle icône

Vous pouvez ajouter une nouvelle icône dans un fichier projet, ou créer un fichier icône autonome d'extension .ICO.

Ajout d'une icône dans un fichier projet

Pour ajouter une icône dans un fichier .RC (ou .DLG) :

1. Ouvrez un nouveau projet ou un projet existant (le chapitre 14 décrit la procédure d'ouverture d'un projet).
2. Choisissez Resource | New. Resource Workshop affiche la boîte de dialogue New Resource.

Les ressources intégrées et liées sont traitées à la section "Ressources intégrées ou liées" du chapitre 14.

Voir la section "Choix du nombre de couleurs d'une ressource" du chapitre 19 pour une description des formats de couleurs.

Création d'un fichier icône autonome

3. Faites défiler la boîte liste Resource Type et sélectionnez ICON.
Le nom du fichier projet en cours apparaît dans la boîte sous la rubrique "Place resource in". Vous pouvez faire défiler le contenu de cette liste pour choisir un autre fichier, ou cliquer sur OK pour accepter le fichier projet en cours.
4. Resource Workshop vous demande si vous voulez créer l'icône au format source ou au format binaire compatible Microsoft.
Choisissez Source pour créer une icône intégrée dans le fichier projet. Resource Workshop affiche alors la boîte de dialogue New Icon.
Si vous choisissez Binary, vous créez un fichier .ICO autonome (voir la section suivante).
5. La boîte de dialogue New Icon Image vous demande de spécifier la taille (en pixels) de la nouvelle icône et le nombre de couleurs à gérer. Si le Presse-papiers de Windows contient une image en mode point, les caractéristiques de celle-ci (dimensions et couleurs) deviennent les valeurs par défaut.

Choisissez la taille d'image et le format de couleur désiré, puis cliquez sur OK. Resource Workshop place le nom de la nouvelle icône dans la fenêtre projet et lance l'éditeur Bitmap.

Vous pouvez créer un fichier icône autonome en exécutant une des procédures suivantes :

- Choisir Binary dans la boîte de dialogue qui vous demande si vous voulez créer votre icône au format Source ou Binary.
- Choisir File | New Project et sélectionner .ICO dans la boîte de dialogue New Project.

Pour lier un fichier icône autonome à un fichier projet, ouvrez le fichier projet, puis choisissez File | Add to Project (voir la section "Ajout d'une ressource" du chapitre 14).

**Option de format
Binary**

Lorsque vous choisissez le format Binary, Resource Workshop affiche la boîte de dialogue New File Resource afin de nommer votre fichier icône et spécifier le projet dans lequel il sera référencé. Lorsque le type de ressource sélectionné est une icône, la boîte de saisie File Name indique *.ICO.

1. Entrez un nom pour votre fichier .ICO dans la boîte de saisie File Name. Assurez-vous que le nom de votre projet apparaît dans la boîte liste intitulée "A reference to the resource...". Pour changer le répertoire de stockage du fichier, définissez le chemin en double-cliquant sur les icônes appropriées dans la boîte liste Directories. Le chemin choisi est affiché dans la boîte de saisie Path.
2. Lorsque vous avez effectué vos sélections, cliquez sur OK. Resource Workshop affiche une boîte de dialogue vous spécifiant que le fichier nommé n'existe pas, et vous demandant si vous voulez créer le fichier. Cliquez sur Yes.
3. Lorsque vous quittez la boîte de dialogue New File Resource, Resource Workshop affiche la boîte de dialogue New Icon Image. Lorsque vous sélectionnez les couleurs et la taille de l'icône, Resource Workshop lance l'éditeur Bitmap.

Vous pouvez aussi choisir un nom de fichier dans la boîte liste Files. Dans ce cas, vous écraserez le fichier existant. Par précaution, Resource Workshop vous demande si vous voulez écraser le fichier.

Fichier projet icône

Si vous choisissez File | New Project et sélectionnez .ICO dans la boîte de dialogue New Project, vous créez automatiquement un fichier icône autonome. Resource Workshop affiche immédiatement la boîte de dialogue New Icon Image. Après avoir spécifié les couleurs et la taille, Resource Workshop lance l'éditeur Bitmap.

Edition des icônes

Les sections suivantes décrivent plusieurs techniques d'édition de vos icônes.

**Visualisation
d'autres résolutions**

La résolution standard EGA/VGA est de 32 × 32, mais vous pouvez créer des icônes selon deux autres résolutions : 32 × 16 et 64 × 64. Le menu View comprend des commandes qui vous permettent de visualiser l'icône selon d'autres résolutions d'affichage.

- Si vous créez l'icône en résolution 32×32 , la commande View | CGA Resolution [32×16] vous permet de voir l'aspect de l'icône sur un écran CGA.
- Si vous créez l'icône en résolution 32×16 (CGA), la commande View | EGA-VGA Resolution [32×32] vous permet de voir l'aspect de l'icône sur un écran EGA ou VGA.
- Si vous créez l'icône en résolution 64×64 , les deux commandes sont disponibles. Vous pouvez voir l'aspect de l'icône sur un écran CGA, ou bien sur un écran EGA ou VGA.

Pour sortir de ces modes de visualisation, choisissez View | Actual Size.



Avec ces commandes, vous ne changez que la vue de l'icône, et non l'icône elle-même. Elle est toujours de même résolution.

Utilisation des zones couleur transparentes et inversées

Les couleurs inversées sont généralement plus utiles pour les curseurs que pour les icônes.

Comme spécifié à la section "Zones couleur transparentes et inversées" du chapitre 19, vous pouvez utiliser des zones couleur transparentes et inversées dans vos icônes. Au moment de l'exécution, la zone transparente "s'efface" et laisse apparaître la couleur du bureau au-dessous. Toute partie de l'icône peinte avec une couleur inversée prend la couleur opposée du bureau au-dessous : blanc sur noir, sarcelle sur rouge, par exemple.

Lorsque vous commencez une nouvelle icône, il s'agit initialement d'une grille constituée entièrement de la couleur transparente en cours. Si vous dessinez une simple ligne d'une autre couleur sur la grille, puis testez l'icône, vous verrez cette ligne sur le bureau, mais vous ne verrez pas le contour de la grille ni la zone remplie avec la couleur transparente.

Ajout d'une image à une ressource icône

En général, vous créez une nouvelle ressource icône pour chaque conception d'icône (appelée simplement *icône*), et vous ne placez pas différentes icônes dans la même ressource icône. Cependant, il est probable que vous voudrez placer divers formats de couleur de la même icône dans la même ressource icône. Ces différences de couleur sur la même icône sont appelées *images*. Par exemple, si vous voulez une version bicolore et une version en 16 couleurs de la même conception d'icône, vous pouvez les stocker dans la même ressource icône.

La ressource icône accepte divers formats de couleur. En effet, Windows choisit un format de couleur selon la capacité du matériel d'affichage à gérer le format. Windows choisit un format bicolore pour un pilote d'affichage monochrome, et un format en 16 couleurs pour le pilote VGA Windows standard.

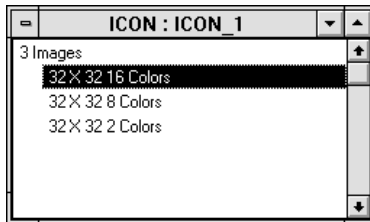


Windows 3.x ne supporte pas entièrement la version en 256 couleurs d'une icône, même si votre matériel d'affichage l'accepte. Votre programme doit assurer lui-même la gestion des 256 couleurs.

Voici comment ajouter une nouvelle image à une ressource icône existante.

1. Ouvrez un projet.
2. Double-cliquez sur l'entrée ICON dans la fenêtre projet, ou sélectionnez l'entrée ICON, puis choisissez Resource | Edit. Resource Workshop affiche la fenêtre Icon.

Figure 20.1
Fenêtre icône



3. Pour créer une nouvelle version d'une icône existante, choisissez Images | New Image. Resource Workshop affiche la boîte de dialogue New Icon Image dans laquelle vous pouvez choisir la taille de l'image et le format de couleur.
4. Choisissez la même taille que l'image existante et un nouveau format de couleur, puis cliquez sur OK. Resource Workshop affiche la nouvelle entrée image dans la fenêtre Icon.
5. Double-cliquez sur la nouvelle icône, ou bien sélectionnez-la et choisissez Image | Edit Image. Vous verrez alors l'éditeur Bitmap.
6. De façon typique, l'étape suivante consiste à ouvrir l'une des images icône existantes, et à la copier dans la nouvelle image (encore vierge). Vous pouvez aussi vouloir éditer l'image si les couleurs sont restituées avec une modification de la forme de l'icône.

Modification des attributs d'une icône

Pour changer les attributs (résolution ou format de couleur) de l'image icône en cours, exécutez la procédure suivante :

1. Choisissez Icon | Size and Attributes pour afficher la boîte de dialogue Icon Image Attributes.

A l'exception d'un bouton-poussoir supplémentaire (décrit ci-après), la boîte de dialogue Icon Image Attributes est identique à la boîte de dialogue New Icon Image.

2. Sélectionnez une résolution ou un format de couleur, puis cliquez sur OK.



La modification des attributs d'une icône change réellement l'icône, par opposition à la visualisation de l'icône à d'autres résolutions.

Affichage des informations périphérique

Le bouton-poussoir supplémentaire de la boîte de dialogue Icon Image Attributes est appelé Device Info. Lorsque vous cliquez sur ce bouton, Resource Workshop affiche la boîte de dialogue Display Device Information, qui fournit les informations suivantes sur votre périphérique d'affichage :

- Nombre de bits par pixel
- Nombre de plans de couleurs
- Nombre de couleurs gérées
- S'il s'agit d'un périphérique palette

Si le périphérique supporte les palettes couleur logiques, la boîte de dialogue Display Device Information énumère les informations suivantes :

- Le nombre d'entrées dans la palette système
- Le nombre d'entrées réservées dans la palette système
- La résolution couleur du périphérique en bits par pixel

Création d'une icône exemple



Cette section explique comment créer un exemple d'icône représenté ci-contre.

Pour créer la nouvelle icône, vous devez d'abord ouvrir un projet, puis suivre une série d'étapes pour afficher l'éditeur Bitmap contenant la nouvelle image vierge.

Le chapitre 14 décrit la procédure d'ouverture d'un projet.

1. Choisissez File | New Project pour créer un nouveau projet, ou File | Open Project pour ouvrir un projet existant.
2. Choisissez Resource | New, et indiquez à Resource Workshop qu'il doit créer une nouvelle ressource. Lorsque Resource Workshop vous demande le type de ressources souhaité, choisissez ICON.
3. Choisissez Source pour stocker l'icône en tant que script ressource dans le fichier .RC.
4. Resource Workshop ouvre la boîte de dialogue New Icon Image. Activez 32x32 et 16 Colors, puis cliquez sur OK.
5. Resource Workshop ouvre l'éditeur Bitmap.

Dessin de la calculatrice

Pour dessiner l'icône calculatrice :

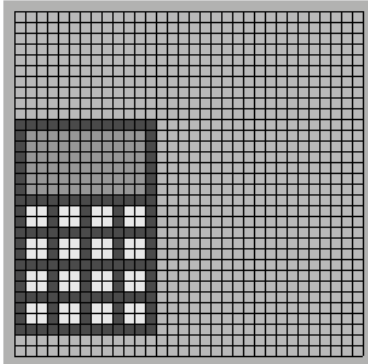
1. Choisissez Options | Editor Options. Activez Grid On Zoomed Windows, pour vous aider à aligner les touches de la calculatrice, puis cliquez sur OK.
La grille vous indique les pixels individuels lorsque vous zoomez votre icône dans l'éditeur Bitmap. Il est plus facile de voir ce que vous dessinez sur une vue agrandie de l'icône.
2. Gardez l'icône affichée en taille réelle dans la fenêtre droite. Pour zoomer l'image de la fenêtre gauche, double-cliquez sur l'icône Zoom, dans la palette Tools, jusqu'à ce que l'image atteigne la taille maximale, tout en restant visible sur l'écran.
Sur la vue zoomée, chaque carré de la grille représente un pixel.
Maintenant vous pouvez dessiner.
3. Choisissez le rouge sombre pour la calculatrice. Lorsque vous cliquez dans la palette Colors, les lettres *FG* (*foreground*, premier plan) apparaissent sur cette couleur.

4. Cliquez sur le rectangle plein de la palette Tools. Cet outil carré est situé à droite, sous l'outil Text (le grand "T").
5. Dessinez la calculatrice dans la partie inférieure gauche de l'icône.

Veillez à laisser deux pixels sous la calculatrice, pour placer l'ombrage.

Vous pouvez utiliser le rectangle plein pour dessiner toutes les parties de la calculatrice : la face avant, la zone d'affichage et les touches. Choisissez diverses couleurs pour ces éléments, en cliquant dans la palette Colors avant de dessiner (essayez le jaune pour les touches et le cyan sombre pour l'écran). Lorsque vous avez terminé, doit ressembler à ceci :

Figure 20.2
Calculatrice avant
l'ajout de l'ombre
portée



Si vous commettez une erreur, vous pouvez utiliser la fonction de suppression (Edit | Undo ou *Alt+Retour arrière*) pour la rectifier. Vous pouvez aussi utiliser l'outil Rectangle de sélection ou l'outil Ciseaux, pour délimiter une zone et l'effacer (utilisez la touche *Suppr* ou Edit | Delete). Enfin, vous pouvez utiliser la gomme.

Si vous voulez tout effacer et recommencer, double-cliquez simplement sur la gomme de la palette Tools.

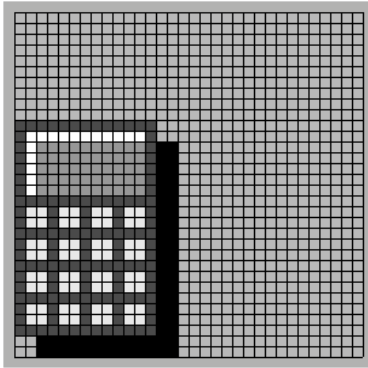
Ajout d'un effet à trois dimensions

Pour ajouter un effet à trois dimensions à la calculatrice :

1. Choisissez la couleur noire pour l'ombre.
2. Utilisez l'outil Ligne pour remplir une ligne ombrée, sur le côté et sous la calculatrice. La ligne doit avoir une épaisseur de deux pixels, débiter deux pixels au-dessous du sommet, et se terminer à deux pixels du côté gauche.
3. Vous pouvez aussi donner un aspect en trois dimensions à la fenêtre calculatrice en dessinant une ligne blanche le long du bord gauche de la fenêtre et de son sommet.

Voici à quoi ressemblerait maintenant la calculatrice.

Figure 20.3
Calculatrice avec
effet d'ombre



Lorsque vous avez terminé de dessiner la calculatrice, choisissez File | Save Project pour enregistrer votre nouvelle image icône. Il est toujours judicieux d'enregistrer fréquemment.

Dessin de la page du livre de comptes

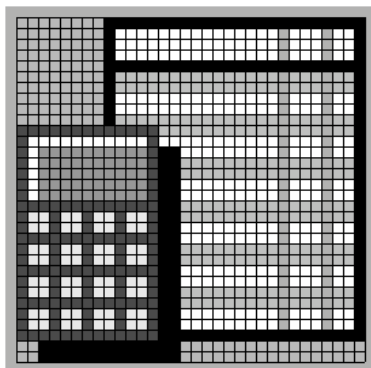
Pour dessiner la page du livre de comptes :

1. Choisissez le noir, puis l'outil Ligne.
2. Utilisez l'outil Ligne pour dessiner une ligne noire verticale. Commencez sur le cinquième pixel à gauche de l'angle supérieur droit de la calculatrice. Poursuivez la ligne jusqu'au sommet de la fenêtre d'édition, puis jusqu'à son angle supérieur droit, et redescendez deux pixels au-dessus du bas de l'éditeur Bitmap. Enfin, continuez jusqu'à atteindre l'ombrage noir à droite de la calculatrice. Vous venez de dessiner le contour de la page du livre de comptes.
3. Choisissez la couleur blanche et sélectionnez l'outil Boîte de peinture. Cliquez la boîte de peinture sur la page du livre de comptes, pour la remplir de blanc.
4. Choisissez cyan (le bleu le plus clair) et l'outil Ligne.
5. Cliquez sur la case du style de ligne dans l'angle inférieur droit de la palette Tools. Dans la boîte de dialogue Current Pen Style, choisissez l'épaisseur de ligne de deux pixels (à droite de Null).

Utilisez la commande
Edit | Undo pour
corriger les erreurs.

6. Commencez deux pixels au-dessus du bas de la page du livre de comptes, et à un pixel de l'endroit où vous voulez commencer la ligne (deux pixels de large), dessinez une série de lignes hautes de deux pixels, au travers de la partie visible de la page. Chaque ligne doit être espacée de deux pixels. Les trois lignes inférieures seront reliées sur la gauche par l'ombrage de la calculatrice.
7. Choisissez le rouge clair et l'outil Ligne.
8. Cliquez sur le style Ligne, dans l'angle inférieur droit de la palette Tools. Dans la boîte de dialogue Current Pen Style, choisissez l'épaisseur de ligne d'un pixel (au-dessus de Null).
9. En commençant à trois pixels du bord droit du livre de comptes, dessinez une ligne verticale aussi longue que la page du livre. A une distance de quatre pixels sur la gauche, dessinez une seconde ligne verticale rouge.
10. Choisissez le noir et dessinez une ligne horizontale débutant immédiatement au-dessus de la zone supérieure cyan, et parcourant la largeur de la page du livre de comptes.
11. Lorsque vous avez terminé votre icône, la fenêtre de l'éditeur Bitmap aura l'aspect suivant :

Figure 20.4
Icône Home Budget
terminée



Création de curseurs

Les *curseurs* sont des images bitmap d'une taille de 32x32 pixels qui représentent l'emplacement en cours du pointeur de la souris sur l'écran. Une application Windows comporte souvent un certain nombre de curseurs différents qui représentent différentes fonctions du programme.

Windows procure un ensemble de curseurs standard que vous utilisez dans vos programmes. De plus, vous pouvez créer vos propres curseurs spéciaux pour représenter diverses fonctions. Un de ces curseurs spéciaux est le curseur Boîte de peinture de Resource Workshop qui s'affiche lorsque vous choisissez un outil Boîte de peinture dans l'éditeur Bitmap (décrit au chapitre 19).

L'éditeur Bitmap comprend de nombreux outils de dessin, et une palette Colors qui facilite la sélection des couleurs. Il vous permet aussi de zoomer une image, et montre de nombreuses vues du curseur créé.

Travailler avec des curseurs implique quatre étapes principales

1. Lancer l'éditeur Bitmap

Si vous créez une nouvelle ressource curseur, l'éditeur Bitmap vous présente une grille curseur vide. Si vous modifiez un curseur existant, il apparaît dans l'éditeur Bitmap.

2. Créer ou modifier le curseur.
3. Tester le curseur.
4. Enregistrer le curseur.

Création d'un nouveau curseur

Vous pouvez ajouter un nouveau curseur dans un fichier projet, ou créer un fichier curseur autonome avec l'extension .CUR.

Ajout d'un curseur dans un fichier projet

La boîte de dialogue New Resource est décrite à la section "Ajout d'une ressource" du chapitre 14.

Pour ajouter un curseur dans un fichier .RC (ou .DLG)

1. Ouvrez un nouveau projet ou un projet existant (le chapitre 14 décrit la procédure d'ouverture d'un projet).
2. Choisissez Resource | New. Resource Workshop affiche la boîte de dialogue New Resource.
3. Sélectionnez CURSOR dans la boîte liste Resource Type.

Le nom du fichier projet en cours apparaît dans la boîte sous la rubrique "Place resource in". Vous pouvez faire défiler le contenu de cette liste pour choisir un autre fichier, ou cliquez sur OK pour accepter le fichier projet en cours.

4. Resource Workshop vous demande si vous voulez créer le curseur au format source ou au format binaire compatible Microsoft.
5. Choisissez Source pour créer un curseur intégré dans le fichier projet. Resource Workshop lance alors l'éditeur Bitmap.

Si vous choisissez Binary, vous créez un fichier .CUR autonome (voir la section suivante).

Les ressources intégrées et liées sont traitées à la section "Ressources intégrées ou liées" du chapitre 14.

Création d'un fichier curseur autonome

Vous pouvez créer un fichier curseur autonome en exécutant une des procédures suivantes

- Choisir Binary dans la boîte de dialogue qui vous demande si vous voulez créer votre curseur au format Source ou Binary (voir la figure 9.3).
- Choisir File | New Project et sélectionner .CUR dans la boîte de dialogue New Project.

Pour lier un fichier curseur autonome à un fichier projet, ouvrez le fichier projet, puis choisissez File | Add to Project (voir la section "Ajout d'une ressource" du chapitre 14).

**Option de format
Binary**

Lorsque vous choisissez le format Binary, Resource Workshop affiche la boîte de dialogue New File Resource afin de nommer votre fichier curseur et spécifier le projet dans lequel il sera référencé. Lorsque le type de ressource sélectionné est un curseur, la boîte de saisie File Name indique *.CUR.

1. Entrez un nom pour votre fichier .CUR dans la boîte de saisie File Name. Assurez-vous que le nom de votre projet apparaît dans la boîte liste intitulée "A reference to the resource...". Pour changer le répertoire de stockage du fichier, définissez le chemin en double-cliquant sur les icônes appropriées dans la boîte liste Directories. Le chemin choisi est affiché dans la boîte de saisie Path.
2. Lorsque vous avez effectué vos sélections, cliquez sur OK. Resource Workshop affiche une boîte de dialogue vous spécifiant que le fichier nommé n'existe pas, et vous demandant si vous voulez créer le fichier. Cliquez sur Yes.
3. Lorsque vous quittez la boîte de dialogue New File Resource, Resource Workshop lance l'éditeur Bitmap.

Vous pouvez aussi choisir un nom de fichier dans la boîte liste Files. Dans ce cas, vous écraserez le fichier existant. Par précaution, Resource Workshop vous demande si vous voulez écraser le fichier.

**Fichier projet
curseur**

Si vous choisissez File | New Project et sélectionnez .CUR dans la boîte de dialogue New Project, vous créez automatiquement un fichier curseur autonome. Resource Workshop lance immédiatement l'éditeur Bitmap.

Edition des curseurs

Les sections suivantes décrivent plusieurs techniques d'édition de vos curseurs.

**Palette Colors pour
les curseurs**

La palette Colors pour les curseurs diffère de celle des autres ressources bitmap en un point important : seules deux couleurs de premier plan et de fond sont disponibles, le noir et le blanc. Un curseur typique consiste en une forme blanche (une flèche ou un point d'interrogation, par exemple) avec un contour noir.

**Comment travailler
avec les zones
transparentes et
inversées**

Comme spécifié à la section "Zones couleur transparentes et inversées" du chapitre 19, vous pouvez utiliser des zones couleur transparentes et inversées dans vos curseurs. Au moment de l'exécution, la zone transparente "s'efface" et laisse apparaître la couleur du bureau au-dessous. Toute partie du curseur peinte avec une couleur inversée prend la couleur opposée du bureau au-dessous : blanc sur noir, sarcelle sur rouge, par exemple. Les couleurs inversées sont généralement utilisées pour le "point actif" du curseur.

Lorsque vous commencez un nouveau curseur, il s'agit initialement d'une grille constituée entièrement de la couleur transparente en cours. Si vous dessinez une simple ligne d'une autre couleur sur la grille, puis testez le curseur, vous verrez cette ligne sur la couleur du bureau, mais vous ne verrez pas le contour de la grille ni la zone remplie avec la couleur transparente.

Ne spécifiez pas *noir et blanc* pour les couleurs inversées ou transparentes, ou bien votre curseur ne sera pas visible au moment de l'exécution ou changera constamment de couleur à tout déplacement de la souris sur la surface d'application.

**Réglage du point
actif du curseur**

Lorsque vous éditez un curseur, il est important de déterminer le point actif du curseur. Le point actif est le pixel du curseur qui fixe l'emplacement où l'utilisateur place le curseur et clique pour faire une sélection.

Pour définir un point actif, déterminez ses coordonnées exactes en pixels. Ce sont des unités horizontales (x) et verticales (y). Le pixel supérieur gauche de l'image curseur est $x=0$ et $y=0$. Le pixel inférieur droit d'un curseur de 32×32 est $x=31$ et $y=31$, et pour un curseur de 32×16 , $x=31$ et $y=15$.

Pour définir le point actif, suivez les étapes ci-dessous

1. Le curseur apparaissant dans l'éditeur Bitmap, zoomez en avant sur l'image curseur jusqu'à ce qu'elle soit suffisamment grande pour vous permettre de choisir précisément les coordonnées en pixels du point actif.
2. Vérifiez que la grille est affichée sur l'image zoomée. Le cas échéant, choisissez Options | Editor Options, et activez Grid On Zoomed Windows.
3. Sélectionnez un outil de dessin vous permettant de pointer précisément sur un pixel. L'outil Ligne est un bon choix, car il comprend un viseur indiquant exactement l'emplacement du point actif.

4. Avec l'outil choisi, pointez, sur l'image agrandie, l'emplacement souhaité pour le point actif. Examinez les coordonnées affichées sur la ligne d'état, et notez-les.
5. Choisissez **Cursor | Set Hot Spot**, et tapez les coordonnées.

Création de polices

Ce manuel utilise les termes "police" et "ressource police" de façon interchangeable.

Selon la définition standard, une *police* est un jeu de caractères de taille et d'un style donné. Une *ressource police* est une collection de données utilisée par un ordinateur pour dessiner des images bitmap individuelles (lettres ou autres caractères) sur un périphérique de sortie, tel qu'un moniteur d'affichage ou une imprimante.

La police contient des données qui décrivent la collection générale d'images, comme le nom de la police, la taille suggérée, le jeu de caractères, les lettres de la police, etc. La police contient aussi des informations nécessaires à l'ordinateur pour dessiner chaque image bitmap.

Travailler avec des ressources police implique cinq étapes principales :

1. Lancer l'éditeur Paint en chargeant une nouvelle police ou une police existante.
2. Créer ou éditer une image police avec l'éditeur Paint.
3. Enregistrer la ressource police en tant que partie du fichier projet, ou dans un fichier distinct.
4. Quitter Resource Workshop et ajouter la ressource police dans une DLL (ressource uniquement) spécifique avec l'extension .FON.
5. Insérer un appel au fichier .FON dans votre programme, le compiler, et tester la ressource police.

Types de polices

Windows supporte deux types de polices : les polices tramées et les polices vectorielles. Resource Workshop crée et édite uniquement les polices tramées.

- Les polices tramées contiennent une image bitmap pour chaque caractère.
- Les polices vectorielles contiennent des commandes de dessin pour chaque caractère. Généralement, elles contiennent aussi des "conseils" pour produire des images de meilleure qualité à des tailles différentes. Les polices vectorielles acceptées par Windows ne comportent pas ces conseils. Ces polices limitées sont appelées polices vecteur, dont Roman, Script et Modern sont des exemples. En tant qu'utilisateur Windows, vous disposez d'une technologie de police plus sophistiquée, grâce aux générateurs de polices de fournisseurs tierce partie.

Pourquoi placer des bitmaps dans une police

Vous pouvez utiliser Resource Workshop pour créer des lettres personnalisées. Cependant, vous voudrez que ce travail soit effectué par un package spécialisé de développement de polices. Vous utiliserez probablement Resource Workshop pour créer des *polices image bitmap* : de petits graphiques par points à grouper ensemble.

Dans certains cas, les ressources police et les ressources bitmap ne sont pas interchangeables. Si vous créez un pinceau pour peindre une zone de l'écran, vous devrez le créer comme une ressource bitmap. Pour créer des images bitmap pour les afficher simplement à l'écran, telle qu'une bombe ou un panneau STOP, vous pourriez utiliser les ressources bitmap ou les ressources police. Par exemple, cette image de bombe a été créée en tant que ressource police.



Plusieurs raisons peuvent vous amener à définir des images en tant que partie d'une ressource police au lieu de ressources bitmap distinctes :

- Il est plus facile d'écrire du code Windows pour charger une police en mémoire et la dessiner, que de charger et de dessiner la même image stockée en tant que graphique par points.
- Une police peut stocker jusqu'à 256 graphiques par points. Si votre programme utilise généralement un certain ensemble d'images bitmap en même temps, vous pouvez ranger ces images dans une ressource police, et le programme peut alors charger uniquement cette ressource lorsqu'il a besoin de cet ensemble d'images.

- Une police peut stocker plusieurs images de manière plus efficace que le stockage des mêmes images en tant que ressources bitmap séparées.

Notez cependant qu'une ressource police dispose d'une certaine quantité de mémoire au début de celle-ci. A chaque chargement d'une police en mémoire, Windows charge aussi l'en-tête de police (voir la section "Définition d'un en-tête pour une ressource police" plus loin dans ce chapitre). Si vous créez une seule image et ne savez pas si vous devez la définir en tant que ressource police ou ressource bitmap, choisissez "ressource bitmap" si une utilisation efficace de la mémoire est importante.

Création d'une nouvelle ressource police

Vous pouvez ajouter une nouvelle ressource police dans un fichier projet, ou créer un fichier police autonome avec l'extension .FNT.

Ajout d'une police dans un fichier projet

Pour ajouter une police dans un fichier .RC (ou .DLG) :

1. Ouvrez un nouveau projet ou un projet existant (le chapitre 14 décrit la procédure d'ouverture d'un projet).
2. Choisissez Resource | New. Resource Workshop affiche la boîte de dialogue New Resource.
3. Sélectionnez FONT dans la boîte liste Resource Type.
Le nom du fichier projet en cours apparaît dans la boîte sous la rubrique "Place resource in". Vous pouvez faire défiler le contenu de cette liste pour choisir un autre fichier, ou cliquez sur OK pour accepter le fichier projet en cours.
4. Resource Workshop vous demande si vous voulez créer la police au format source ou au format binaire compatible Microsoft.

Choisissez Source pour créer une police intégrée dans le fichier projet. Resource Workshop lance alors l'éditeur Paint.

Si vous choisissez Binary, vous créez un fichier .FNT autonome (voir la section suivante).

Les ressources intégrées et liées sont traitées à la section "Ressources intégrées ou liées" du chapitre 14.

Création d'un fichier police autonome

Vous pouvez créer un fichier police autonome en exécutant une des procédures suivantes :

- Choisir Binary dans la boîte de dialogue qui vous demande si vous voulez créer votre police au format Source ou Binary.
- Choisir File | New Project et sélectionner .FNT dans la boîte de dialogue New Project.

Pour lier un fichier police autonome à un fichier projet, ouvrez le fichier projet, puis choisissez File | Add to Project (voir la section "Ajout d'une ressource" du chapitre 14).

Option de format Binary

Lorsque vous choisissez le format Binary, Resource Workshop affiche la boîte de dialogue New File Resource afin de nommer votre fichier police et de spécifier le projet dans lequel il sera référencé. Lorsque le type de ressource sélectionné est une police, la boîte de saisie File Name indique *.FNT.

1. Entrez un nom pour votre fichier .FNT dans la boîte de saisie File Name. Assurez-vous que le nom de votre projet apparaît dans la boîte liste intitulée "A reference to the resource...". Pour changer le répertoire de stockage du fichier, définissez le chemin en double-cliquant sur les polices appropriées dans la boîte liste Directories. Le chemin choisi est affiché dans la boîte de saisie Path.
2. Lorsque vous avez effectué vos sélections, cliquez sur OK. Resource Workshop affiche une boîte de dialogue vous spécifiant que le fichier nommé n'existe pas, et vous demandant si vous voulez créer le fichier. Cliquez sur Yes.
3. Lorsque vous quittez la boîte de dialogue New File Resource, Resource Workshop lance l'éditeur Paint.

Vous pouvez aussi choisir un nom de fichier dans la boîte liste Files. Dans ce cas, vous écraserez le fichier existant. Par précaution, Resource Workshop vous demande si vous voulez écraser le fichier.

Fichier projet police

Si vous choisissez File | New Project et sélectionnez .FNT dans la boîte de dialogue New Project, vous créez automatiquement un fichier police autonome. Resource Workshop lance immédiatement l'éditeur Paint.

Edition d'une ressource police

Lorsque vous ouvrez une nouvelle ressource police ou une ressource police existante dans l'éditeur Paint, les caractères ou les images dans la police en cours d'édition sont affichés dans une "liste" située sur la bordure droite de la fenêtre.

Lorsque vous lancez l'éditeur Paint, il charge automatiquement la première image police de la liste. Pour charger une autre image afin de l'éditer, cliquez-la dans la bordure droite de l'éditeur Paint. Utilisez ensuite les outils de l'éditeur Paint pour effectuer les changements souhaités.

Définition et ajout de caractères pour une police

Si vous créez une nouvelle ressource police, elle ne comprend qu'une image de 8x8 pixels. Généralement, vous voudrez plusieurs images dans votre ressource police. Vous pouvez aussi vouloir spécifier une taille d'image différente. Pour spécifier plusieurs images dans une ressource police et pour en modifier la taille, utilisez l'option Font | Font Size.

Définition de la taille de police

Pour définir la taille de police, suivez les étapes ci-dessous :

1. Choisissez Font | Font Size pour afficher la boîte de dialogue Font Size Information.
2. Effectuez votre choix parmi les options de taille décrites dans le tableau 22.1. Les images d'une ressource police peuvent être de *largeur variable* ou de *largeur fixe*.

Voir la section "Création de polices proportionnelles" de ce chapitre pour plus d'informations sur les polices de largeur variable.

- Les polices contenant des lettres ou des images variables en largeur sont appelées polices de largeur variable ou polices *proportionnelles*. Par exemple, dans une police proportionnelle, la lettre "m" est nettement plus large que la lettre "i". Les livres (y compris cet ouvrage) sont presque toujours imprimés avec une police proportionnelle.
- Les polices dont les caractères ou les images sont toutes de même largeur sont appelées polices de largeur fixe ou polices à *espacement fixe*. La plupart des machines à écrire utilisent des polices à espacement fixe.

Les caractères de largeur variable occupent généralement moins de place et sont plus agréables à regarder que les caractères d'une police à espacement fixe. Comparez les deux lignes suivantes, et plus particulièrement l'espacement autour de la lettre "i".

Minimum (police proportionnelle)

Minimum (police à espacement fixe)

Tableau 22.1
Options de taille de
police

Option	Description
Width	Si vous voulez que toutes les images aient la même largeur (largeur fixe), tapez la largeur en pixels. Si vous voulez des largeurs différentes (largeur variable), tapez 0 (zéro) et spécifiez une largeur maximale.
Height	Tapez la hauteur, en pixels, des images police.
Average Width	Resource Workshop calcule une largeur moyenne d'image police, si vous avez spécifié 0 pour Width (police proportionnelle). Pour une police à espacement fixe, l'option Average Width est identique à l'option Width. La largeur moyenne est calculée lorsque vous ouvrez cette boîte de dialogue. Vous ne verrez pas la modification de cette valeur si vous tapez d'autres changements dans cette boîte de dialogue.
Maximum Width	Pour les polices proportionnelles, spécifiez la largeur maximale en pixels. Cette option n'est disponible que si vous avez tapé 0 en regard de Width.
Stretch Current Chars	Activez cette option si vous voulez augmenter ou diminuer la hauteur ou la largeur d'images existantes, sur la base des modifications de largeur et de hauteur tapées dans cette boîte de dialogue. Les caractères sont alors déformés proportionnellement.

Définition du nombre de caractères

La boîte de dialogue Font Size Information vous permet aussi de choisir le nombre d'images à inclure dans votre ressource police. La première et la dernière valeur des options Character (voir le tableau 22.2) déterminent le nombre de caractères que la ressource police contiendra.

Si le nombre est trop bas, vous pouvez changer les valeurs pour augmenter le nombre de caractères disponibles.

Projection du jeu de caractères

Choisissez une plage de codes décimaux pour établir une correspondance entre vos images police et le jeu de caractères ANSI. Par exemple, pour établir une correspondance entre une image police et le caractère *a*, spécifiez le code décimal 97. Il n'est pas nécessaire que l'image elle-même soit le caractère *a*, à moins que vous ne le souhaitiez. Ainsi, si l'image de bombe présentée plus haut est projetée sur *a* dans une police donnée, la seconde image police sera projetée sur le caractère *b*, code décimal 98.



Les valeurs ANSI avec lesquelles vous établissez une correspondance avec un ensemble d'images bitmap sont arbitraires, mais doivent être comprises entre 0 et 255. Ces valeurs ANSI deviennent importantes lorsque vous chargez la police dans votre programme et affichez les images bitmap, car vous utilisez la valeur ANSI associée à l'image bitmap pour l'afficher, de même que vous utilisez une valeur ANSI pour afficher un caractère.

Utilisez les options Character suivantes pour projeter le jeu de caractères ou d'images dans votre ressource police :

Tableau 22.2
Options Character

Option	Description
First	Tapez un code décimal ANSI pour définir la première image de votre police. Par exemple, si vous souhaitez que la première image corresponde à a, tapez 97.
Last	Tapez un code décimal ANSI pour définir la dernière image de votre police. Par exemple, si vous voulez que la dernière image corresponde à z, tapez 122.
Default	Tapez un code décimal ANSI pour définir l'image police par défaut affichée lorsque vous éditez cette ressource police. La valeur par défaut doit appartenir à la plage de caractère définie par les valeurs First et Last. Par exemple, si vous avez tapé 97 pour la valeur First et 122 pour Last, vous ne pouvez pas taper 88 pour Default.
Break	Tapez un code décimal ANSI pour définir un caractère de séparation pour votre ressource police. Par exemple, le caractère d'espacement est utilisé pour compléter des lignes justifiées. La valeur Break doit être comprise entre les valeurs First et Last.

Création de polices proportionnelles

Pour créer des caractères ou des images de largeur variable dans votre ressource police :

1. Choisissez Font | Font Size.
2. Face à Width, tapez 0 (zéro).
3. Face à Maximum Width, tapez la largeur maximale (en pixels) pour tous les caractères ou images de votre ressource police.

Définition de la largeur d'un caractère ou d'une image

Après le choix de la valeur 0 dans la boîte de dialogue Font Size Information, vous pouvez choisir la largeur d'une image ou d'un caractère particulier au moyen de la commande Font | Character.

Par exemple, utilisez l'option Font | Font Size pour définir les tailles suivantes pour votre ressource police, qui contient des images bitmap :

Largeur	0
Hauteur	32
Largeur maximale	32

En fonction de ces tailles, toutes les images auront 32 pixels en hauteur et une largeur maximale de 32 pixels. Lors de l'édition d'une image particulière, vous pouvez utiliser l'option Font | Character Width pour définir la largeur de cette image.

Choisissez Font | Character Width pour afficher la boîte de dialogue Character Width.

Utilisez l'option Stretch Current Chars pour déformer l'image par rapport à ses proportions normales.

A côté de la rubrique Width, tapez une valeur inférieure ou égale à la largeur maximale. En outre, vous pouvez activer l'option Stretch Current Chars, si vous voulez que l'image existante soit dilatée ou contractée, sur la base de la modification de largeur tapée dans cette boîte de dialogue.

Définition d'un en-tête pour une ressource police

Chaque ressource police comprend un en-tête qui contient des informations générales concernant la police, comme le nom de la police et les informations de copyright. Si vous définissez une ressource police comprenant des caractères alphanumériques, l'en-tête définit le style et la taille de tous les caractères de la police.

Pour définir l'en-tête d'une police,

1. Affichez la police dans l'éditeur Paint en double-cliquant sur le nom de police dans la fenêtre projet, ou en utilisant Resource | New pour créer une nouvelle ressource police.
2. Choisissez Font | Header pour spécifier les informations d'en-tête. Vous obtenez la boîte de dialogue Font Header Information.

3. Définissez l'en-tête de vos polices personnalisées. Le tableau suivant décrit les choix possibles :

Tableau 22.3
Options d'en-tête

Option	Description												
Face Name	Tapez le nom à affecter à votre police.												
Device	Tapez un nom de périphérique pour votre police afin d'informer vos programmes qu'elle ne peut être utilisée que sur un périphérique particulier.												
Copyright	Tapez les informations de copyright de votre police personnalisée.												
Font version	La version de police 2.00 est supportée dans tous les cas. Vous pouvez utiliser la version 3.0 si vous créez une application Windows 3.x qui fonctionnera dans un environnement en mode protégé (mode standard ou mode 386 étendu) sur un processeur 80386 (ou supérieur).												
Attributes													
- Italic	La police contient des caractères en italique.												
- Underline	La police contient des caractères soulignés.												
- Strikeout	La police contient des caractères biffés.												
- Variable Pitch	La police est une police proportionnelle.												
- Weight	La police est normale (400) ou en caractère gras (700).												
- Family	Décrit la famille de police. Les valeurs acceptables sont : <table border="0" style="margin-left: 20px;"> <tr> <td>0</td> <td>Indifférent</td> <td>3</td> <td>Modern</td> </tr> <tr> <td>1</td> <td>Roman</td> <td>4</td> <td>Script</td> </tr> <tr> <td>2</td> <td>Swiss</td> <td>5</td> <td>Decorative</td> </tr> </table>	0	Indifférent	3	Modern	1	Roman	4	Script	2	Swiss	5	Decorative
0	Indifférent	3	Modern										
1	Roman	4	Script										
2	Swiss	5	Decorative										
- Char set	Définit le jeu de caractères. La valeur est comprise entre 0 et 255. 0, 2 et 255 ont les significations prédéfinies suivantes : <table border="0" style="margin-left: 20px;"> <tr> <td>0</td> <td>ANSI, le jeu de caractères Windows par défaut.</td> </tr> <tr> <td>2</td> <td>Symbole, pour les formules mathématiques et scientifiques.</td> </tr> <tr> <td>255</td> <td>OEM un jeu de caractères spécifique à la machine.</td> </tr> </table>	0	ANSI, le jeu de caractères Windows par défaut.	2	Symbole, pour les formules mathématiques et scientifiques.	255	OEM un jeu de caractères spécifique à la machine.						
0	ANSI, le jeu de caractères Windows par défaut.												
2	Symbole, pour les formules mathématiques et scientifiques.												
255	OEM un jeu de caractères spécifique à la machine.												
Sizes													
- Horiz Res	Nombre horizontal de pixels par pouce logique sur votre affichage vidéo.												
- Vert Res	Nombre vertical de pixels par pouce logique sur votre affichage vidéo.												
- Points	Taille du caractère. Un point correspond à 1/72 de pouce. Un caractère est mesuré du haut de l'ascendante au bas de la descendante. La valeur entrée à ce niveau ne doit pas inclure d'espace pour les signes diacritiques.												
- Internal Leading	Espace en pixels réservé aux signes diacritiques.												
- External Leading	Espace supplémentaire en pixels entre les lignes de caractères.												
- Ascent	La hauteur en pixels du caractère au-dessus de la ligne de base.												

Modification de la taille et des attributs

Lorsque vous éditez une ressource police dans l'éditeur Paint, le menu Font propose trois commandes pour modifier les images police.

- Header définit les informations d'en-tête de votre ressource police, y compris la version de police, son nom, les informations de copyright, etc. Pour plus d'informations, voir la section précédente.
- Font Size définit le jeu de caractères de cette police, ainsi que la largeur et la hauteur de chaque caractère. Pour plus d'informations, voir la section "Définition et ajout de caractères pour une police" plus haut dans ce chapitre.
- Character Width spécifie la largeur d'une image particulière, dans une ressource police de largeur variable. Cette commande ne sera disponible que si vous avez déjà défini une police proportionnelle avec Font | Font Size. Pour plus d'informations, voir la section "Création de polices proportionnelles" plus haut dans ce chapitre.

Utilisation de vos polices dans vos applications

Avec d'autres ressources, vous utilisez Resource Workshop ou le compilateur de ressources Borland (BRC), pour lier vos ressources à vos fichiers exécutables. Cependant, les polices ne peuvent être liées à un fichier exécutable. Vous devrez alors créer une DLL (ressource uniquement) spéciale avec l'extension .FON, puis la charger dans votre programme au moyen de la fonction Windows **AddFontResource**.

Vous pouvez créer un fichier .FON en utilisant soit Turbo Pascal, soit C++ et Turbo Assembler.

Création des ressources définies par l'utilisateur

Outre les types de ressources décrits dans les chapitres précédents, vous pouvez aussi définir vos propres ressources. Après avoir créé un nouveau type de ressource, vous pouvez ajouter à votre projet des *ressources définies par l'utilisateur* de ce type.

Vous pouvez définir vos propres types de ressources pour y stocker des données qui ne s'adaptent à aucun type de ressource standard. Par exemple, si vous voulez créer une ressource chaîne de caractères plus longue que la limite `STRINGTABLE` de 255 caractères, vous pouvez définir votre propre type de ressource, et y stocker vos chaînes de caractères.

Dans votre projet, vous pouvez aussi inclure des *métafichiers*, en tant que ressources définies par l'utilisateur. Un métafichier est un type de graphique par points (en format source, c'est une collection d'appels à l'interface graphique GDI). Ce type de fichier est plus facile à redimensionner et plus indépendant des périphériques que les ressources bitmap standard. Il occupe aussi moins de place de stockage.

Lorsque vous définissez une nouvelle ressource, vous pouvez stocker les données en tant que partie de la définition de la ressource, dans un projet fichier, ou en tant que fichier distinct. Comme avec toute ressource, Resource Workshop peut compiler les données et les lier à votre fichier exécutable pour que votre application puisse en disposer pendant l'exécution.



Vous pouvez aussi utiliser le type de ressource `RCDATA` pour ajouter des données à votre application. Voir la section "Utilisation du type de ressource `RCDATA`" à la fin de ce chapitre pour plus d'informations.

Reportez-vous à l'ouvrage "Programming Windows" de Charles Petzold, pour plus d'informations sur les métafichiers.

Travailler avec des ressources définies par l'utilisateur implique cinq étapes principales :

1. Créer un type de ressource définie par l'utilisateur.
2. Ajouter une ressource définie par l'utilisateur à votre projet.
3. Editer la ressource définie par l'utilisateur.
4. Tester la ressource définie par l'utilisateur.
5. Enregistrer la ressource définie par l'utilisateur.

Création d'un type de ressource

Avant d'ajouter une ressource définie par l'utilisateur à votre projet, vous devez d'abord lui créer un type, de la façon suivante :

Le chapitre 14 décrit la procédure d'ouverture d'un projet.

1. Ouvrez un projet.
2. Choisissez Resource | New.
3. Dans la boîte de dialogue New Resource, cliquez sur le bouton New Type. Resource Workshop affiche la boîte de dialogue New Resource Type.
4. Dans la boîte de dialogue New Resource Type, tapez un nom pour le type de ressource que vous créez. Par exemple, si vous créez une ressource devant contenir un grand bloc de texte, vous pouvez nommer votre nouveau type de ressource TEXT. Appuyez sur *Entrée*.
5. Lorsque le système vous demande si vous voulez créer un identificateur pour le nouveau type de ressource, cliquez sur Yes. Entrez une valeur dans la boîte de dialogue New Identifier. Cette valeur est l'identificateur que Windows et votre programme associeront à ce type d'identificateur.

Si vous utilisez un identificateur numérique, il doit être supérieur à 255, car Windows réserve les valeurs 1 à 255 aux ressources standard.

Une fois que vous avez défini un nouveau type de ressource, et lorsque vous créez une nouvelle ressource, vous verrez ce type énuméré dans la boîte de dialogue New Resource, avec tous les types de ressources standard (ACCELERATOR, BITMAP, DIALOG, etc.).

Ajout d'une ressource définie par l'utilisateur

Le chapitre 14 décrit la procédure d'ouverture d'un projet.

Après avoir créé un type de ressource, vous pouvez ajouter à votre projet une ressource de ce type, de la façon suivante :

1. Ouvrez un projet.
2. Choisissez Resource | New. Resource Workshop affiche la boîte de dialogue New Resource.
3. Dans la boîte de dialogue New Resource, sélectionnez votre type de ressource définie par l'utilisateur. Sous la rubrique "Place resource in", sélectionnez le fichier projet (probablement le fichier actif) où vous voulez stocker la ressource.
4. Lorsque vous cliquez sur OK dans la boîte de dialogue New Resource, Resource Workshop ouvre l'éditeur de texte, avec une définition vierge, pour votre ressource. Par exemple, si vous ajoutez une ressource appelée TEXT, vous verrez le code suivant dans l'éditeur de texte :

```
TEXT_1 TEXT  
(  
  
)
```

La section suivante décrit la procédure d'édition d'une ressource définie par l'utilisateur.

Edition d'une ressource définie par l'utilisateur

Le chapitre 14 décrit la procédure d'ouverture d'un projet.

Pour éditer une ressource définie par l'utilisateur, un projet doit être ouvert. Vous pouvez créer la ressource (voir la section précédente) ou encore ouvrir une ressource existante, dans la fenêtre projet.

Le nom de ressource est identifié dans la fenêtre projet par son type de ressource définie par l'utilisateur, comme toute autre ressource de type prédéfini. Par exemple, si vous créez un type de ressource TEXT et une ressource TEXT, vous verrez l'entrée ressource TEXT dans la fenêtre projet.

Pour ouvrir l'éditeur,

- Double-cliquez sur le nom de la ressource à éditer.
- Sélectionnez le nom de la ressource, puis choisissez Resource | Edit ou Resource | Edit As Text (les deux commandes lancent l'éditeur de texte).

Une fois que vous avez appelé la ressource dans l'éditeur de texte, vous pouvez lui ajouter des données, ou éditer les données qu'elle contient. Voici un exemple de ce qui apparaît lorsque vous ajoutez à votre projet une nouvelle ressource définie par l'utilisateur, du type RESTYPE :

```
RESTYPE_1 RESTYPE
BEGIN
    [définitions de données]
END
```

La première ligne indique le nom et le type de ressource. Pour une nouvelle ressource, Resource Workshop construit un nom par défaut : il adjoint au type de ressource un soulignement et un nombre entier. Par exemple, la première ressource RESTYPE ajoutée à votre projet devient RESTYPE_1, la deuxième devient RESTYPE_2, etc.

Pour ajouter des données à votre ressource, exécutez une des procédures suivantes :

- Utilisez l'éditeur de texte pour taper des données entre les parenthèses.
- Stockez les données dans un fichier distinct, et ajoutez le nom de fichier à la fin de la première ligne du script ressource, comme suit :

```
RESTYPE_1 RESTYPE MYRES.FIL
```

Vous devez aussi effacer les lignes contenant les instructions BEGIN et END.

- Utilisez la commande Add to Project, comme spécifié dans la section suivante.



Après avoir effectué les modifications dans le script ressource, vous devez recompiler la ressource pour les enregistrer. Si vous quittez sans recompiler, les modifications seront perdues.

Intégration des données ressources dans le fichier projet

Comme spécifié dans la section précédente, vous pouvez ajouter des données à votre ressource en stockant les données dans un fichier distinct. L'inconvénient de cette approche est le suivant : si un incident intervient sur le fichier, les données seront perdues. Une autre option consiste à intégrer les données externes dans le script du fichier projet.

Par exemple, si vous disposez du type de ressource définie par l'utilisateur TEXT et du fichier données ressources MY_RES.TXT, vous pouvez intégrer les données dans le projet en suivant les étapes ci-dessous :

1. Choisissez File | Add to Project. Resource Workshop affiche la boîte de dialogue Add File to Project.
2. Sélectionnez "user Resource data" dans la boîte liste File Type.
3. Changez la boîte de saisie File Name pour indiquer * .TXT.
4. Changez le répertoire (si nécessaire) jusqu'à ce que MY_RES.TXT soit visible dans la liste File. Double-cliquez sur MY_RES.TXT.
5. Resource Workshop affiche la boîte de dialogue Custom Resource Type. Double-cliquez sur l'entrée pour le type de ressource TEXT. Une nouvelle ressource TEXT apparaît dans la fenêtre projet.

Vous pouvez aussi entrer le nom de fichier et le chemin d'accès complet dans la boîte de saisie File Name.



Si vous sélectionnez la nouvelle ressource et choisissez Resource | Edit (ou Edit as Text, effet identique), vous constaterez que les ressources ont été converties au format hexadécimal. Pour cette raison, il est conseillé de conserver le fichier de données externe au cas où vous voudriez éditer le script ressource plus tard.

Comment entrer des données dans le script ressource

Lorsque vous utilisez l'éditeur de texte, les touches *Tab*, *Suppr*, *Début*, *Fin*, *PgPréc*, *PgSuiv* et *Retour arrière* opèrent comme d'habitude. Cependant, n'utilisez pas cet éditeur pour un travail de formatage important, car Resource Workshop est susceptible de réorganiser le texte lorsqu'il compile ou décompile cette ressource.

Voici quelques directives pour spécifier les données entre les paramètres BEGIN et END :

Les données peuvent inclure toute combinaison de valeurs numériques et de chaînes.

- Vous pouvez utiliser une notation hexadécimale, octale ou décimale pour représenter les valeurs numériques.
 - Utilisez 0x (un zéro suivi de la lettre x) ou \$ (un signe dollar) comme caractères de tête dans une notation hexadécimale. Cette notation n'accepte que les valeurs 16 bits. Si vous voulez utiliser un nombre impair de valeurs hexadécimales, utilisez le type de données *hexstring* (décrit ci-après).
 - Utilisez 0o (un zéro suivi de la lettre o) ou simplement 0 (zéro) comme caractères de tête en notation octale.

- Vous pouvez aussi représenter les valeurs hexadécimales en utilisant un *hexstring* (chaîne hexadécimale) de valeurs hexadécimales encadrées par des apostrophes. Le compilateur ignore les espaces insérés pour faciliter la lecture des codes hexadécimaux.

Par exemple, vous pourriez représenter les valeurs ASCII des caractères du mot *string* sous la forme '73 74 72 69 6E 67' et le numéro hexadécimal 06E07A avec '06E07A'.

- Si vous incluez des chaînes de texte dans votre ressource, encadrez les chaînes par des guillemets, comme ceci : "string".

Les chaînes ne sont pas automatiquement terminées par un zéro. Pour cela, tapez \0 (barre oblique inverse, zéro), à la fin de la chaîne.

Par exemple, votre script ressource avec des données ajoutées pourrait avoir l'aspect suivant :

```
RESTYPE_1 RESTYPE
BEGIN
    "This is a string".
    0xFFAA 0o7076 01077
    "54 68 69 73 20 69 73 0D 0A 73 6F 6D 65 20 73 61"
    "6D 70 6C 65 0D 64 61 74 61 2E"
END
```

Gestion des données stockées dans un fichier distinct

Si les données sont stockées dans un fichier distinct, vous devez utiliser un éditeur externe pour éditer le fichier. Pour ajouter une référence au fichier de données dans le script ressource :

- Sur la première ligne du script, après le nom du type de ressource, entrez le nom du chemin d'accès complet du fichier.
- Effacez les instructions BEGIN et END.

Le script ressource qui suit (pour la ressource RESTYPE_1), indique que les données sont stockées dans le fichier C:\RW\MYDATA.TXT :

```
RESTYPE_1 RESTYPE c:\rw\mydata.txt
```

Utilisation du type de ressource RCDATA

Les ressources RCDATA ont été incluses dans Resource Workshop pour des raisons de compatibilité.

Vous pouvez utiliser le type de ressource prédéfini RCDATA pour ajouter une ressource données à votre application. Il fonctionne de la même façon qu'un type de ressource définie par l'utilisateur. La différence principale réside dans la capacité d'adressage : vous pouvez préférer de nombreux types de ressources définies par l'utilisateur plutôt qu'un seul type RCDATA.

Si vous utilisez une ressource RCDATA, ajoutez-la au projet en choisissant File | New, et en ajoutant une nouvelle ressource du type RCDATA. Vous apercevez une définition RCDATA vierge dans l'éditeur de texte. Vous pouvez taper les données entre les paramètres BEGIN et END. Pour taper les données, utilisez les règles décrites pour les ressources définies par l'utilisateur.

Messages d'erreur

Cette annexe répertorie les messages d'erreur qui peuvent être générés par le compilateur, l'utilitaire MAKE, le bibliothécaire (TLIB), l'éditeur de liens (TLINK) et le compilateur d'aide pour Windows de Borland C++. Elle comporte également des messages d'erreur générés lors de l'exécution de vos programmes ou en temps réel.

Les messages sont triés dans l'ordre alphabétique en anglais (les messages débutant par un symbole précèdent ceux débutant par un chiffre, puis ceux débutant par une lettre). Les messages débutant par un symbole sont triés selon l'ordre alphabétique du premier mot suivant ce symbole. Par exemple, le message suivant peut s'afficher si vous avez mal déclaré la fonction *ma_fonct* :

```
ma_fonct must be declared with no parameters
```

Reportez-vous à "must" dans la liste pour vérifier ce message.

Types de messages

Il existe trois types de messages, à savoir, les erreurs fatales, les erreurs simples et les mises en garde.

Erreurs fatales

Une erreur fatale peut être générée par le compilateur, l'éditeur de liens ou l'utilitaire MAKE. Elle provoque l'interruption immédiate de la compilation. Vous devez impérativement résoudre le problème avant toute reprise de la compilation.

En cas d'erreur fatale générée par le compilateur ou l'utilitaire MAKE, aucun fichier exécutable .EXE n'est créé. En cas d'erreur fatale générée par l'éditeur de liens, tout fichier .EXE que l'éditeur a pu créé est supprimé avant la fin du traitement par l'éditeur.

Erreurs

Une erreur peut être générée par le compilateur, l'éditeur de liens, l'utilitaire MAKE ou le bibliothécaire. Elle peut également être générée à l'exécution de l'un de vos programmes.

Une erreur générée par le compilateur indique une erreur de syntaxe dans le programme, une erreur dans la ligne de commande ou une erreur d'accès à un disque ou à la mémoire. Ce type d'erreur ne provoque pas l'interruption de la compilation. Le compilateur exécute jusqu'au bout la phase de compilation en cours, interrompt la compilation et affiche les erreurs détectées. Le compilateur essaye de détecter toutes les erreurs possibles dans le code source au cours de chaque phase de compilation (traitement par le préprocesseur, analyse syntaxique, optimisation et génération du code).

Une erreur générée par l'éditeur de liens n'entraîne pas la suppression de fichiers .EXE ou .MAP par ce même éditeur. Il n'est pas conseillé d'exécuter un fichier .EXE dont l'édition de liens a comporté des erreurs. Une erreur générée par l'éditeur de liens est traitée comme une erreur fatale si vous compilez à partir de l'environnement EDI (Environnement de développement intégré).

L'utilitaire MAKE génère une erreur en cas d'inexactitude syntaxique ou sémantique dans le fichier MAKE source. Corrigez ce type d'erreur directement dans le fichier.

Une erreur d'exécution est en général due à une erreur logique dans le code d'un programme. Dans ce cas, corrigez l'erreur dans le code source et recompiliez le programme pour vérifier l'efficacité de cette modification.

Mises en garde

Un message de mise en garde peut être généré par le compilateur, l'éditeur de liens ou le bibliothécaire. Bien que la génération d'un tel message n'empêche pas la fin de la compilation d'un programme, il indique des conditions peu fiables même si la condition à l'origine du message est justifiée dans le langage. Le compilateur génère également des messages de mise en garde si vous utilisez des compilations spécifiques à votre machine dans vos fichiers source.

Messages du compilateur d'aide

Le compilateur d'aide affiche des messages lorsqu'il détecte des erreurs ou des mises en garde au cours de la création du fichier ressource du système d'aide. Un message généré lors du traitement du fichier projet est précédé de la lettre **P** et suivi d'un code d'erreur. Un message généré lors du traitement d'un ou de plusieurs fichier(s) d'écran d'aide RTF est précédé de la lettre **R** et suivi d'un code d'erreur.

Le compilateur d'aide affiche le numéro d'écran d'aide et/ou le nom du fichier contenant l'erreur chaque fois que cela est possible. Si les écrans d'aide ont été numérotés, le numéro affiché avec un message d'erreur désigne la position séquentielle de cet écran dans le fichier RTF (premier, second, etc.). Ce numéro peut correspondre au numéro de page indiqué par votre traitement de texte. Dans les fichiers sources du système d'aide, les écrans sont séparés par des ruptures de page forcées bien que le système d'aide ne soit pas structuré en pages à proprement parler.

Un message débutant par le mot "Error" indique une erreur fatale. Le message d'erreur s'affiche systématiquement et aucun fichier d'aide utilisable n'est généré. Un message débutant par le mot "Warning" indique une erreur de nature moins grave. Un fichier ressource d'aide est généré et peut être chargé sous Windows. Ce fichier peut cependant ne pas fonctionner normalement.

Listes de messages

Pour chaque message sont indiqués dans l'ordre, le type de message, le nom du fichier source et le numéro de ligne où l'erreur a été détectée et le texte du message.

Certains messages sont précédés d'un symbole (comme une variable, un nom de fichier ou un module) extrait de votre programme. Un symbole figure en *italique* dans la description d'un message. Par définition, un symbole peut varier dans un message.

Remarquez que le compilateur génère les messages au fur et à mesure qu'il les détecte. Comme les langages C et C++ n'imposent aucune restriction sur la position d'une instruction dans une ligne de texte, la véritable origine de l'erreur peut en fait se situer une ou plusieurs lignes avant ou après le numéro de ligne indiqué dans le message d'erreur.

Description des messages

-)' missing in macro invocation** *Erreur de MAKE*
Parenthèse gauche manquante dans l'appel d'une macro.
- (expected** *Erreur du compilateur*
Parenthèse gauche manquante. Ajoutez-la au début d'une liste de paramètres.
-) expected** *Erreur du compilateur*
Paranthèse droite manquante. Ajoutez-la à la fin d'une liste de paramètres.
- , expected** *Erreur du compilateur*
Virgule manquante. Les éléments d'une liste de déclarations, d'initialisations ou de paramètres doivent être séparés par une virgule.
- : expected after private/protected/public** *Erreur du compilateur*
Lorsque les mots réservés **private/protected/public** sont utilisés pour débiter une section de classe, ils doivent impérativement être suivis d'un signe deux-points (:).
- < expected** *Erreur du compilateur*
Le mot clé **template** doit être suivi du signe inférieur (<). Dans une déclaration de template, les paramètres formels de template doivent figurer entre les signes < et > immédiatement après le mot clé **template**.
- > expected** *Erreur du compilateur*
Signe ">" manquant dans une instruction de nouveau transtypage (par exemple, `dynamic_cast`).
- @ seen, expected a response-files name** *Erreur du bibliothécaire*
Aucun fichier de recherche n'est spécifié immédiatement après @@.
- { expected** *Erreur du compilateur*
Accolade gauche ({} manquant au début d'un bloc ou d'une initialisation.
- } expected** *Erreur du compilateur*
Accolade droite ({} manquant à la fin d'un bloc ou d'une initialisation.
- 16-bit segments not supported in module module** *Erreur de l'éditeur de liens*
Les segments 16 bits ne sont pas gérés dans les applications 32 bits. Vérifiez que vous n'avez pas compilé par erreur une application 32 bits à l'aide du compilateur 16 bits.
- 286/287 instructions not enabled** *Erreur du compilateur*
Spécifiez l'option de compilateur **-2** sur la ligne de commande ou sélectionnez les options 80286 à partir de la boîte de dialogue Options|Compiler|Code Generation|Advanced Code Generation pour activer les codes opération 286/287.
Remarquez que le code obtenu ne peut être exécuté sur une machine à microprocesseur 8086 et 8088.
- 32-bit record encountered** *Erreur de l'éditeur de liens*
Un fichier objet contenant des enregistrements 32 bits 80386 a été détecté alors que l'option **/3** n'a pas été spécifiée.
- Abnormal program termination** *Erreur d'exécution*
Le programme a fait appel à `abort` car il ne disposait plus de mémoire suffisante pour s'exécuter. Ce message peut être généré en cas de surécritures en mémoire.

Access can only be changed to public or protected*Erreur du compilateur*

Un membre d'une classe de base peut être rendu accessible en mode **public** ou **protected** à partir d'une classe dérivée. Il est impossible de lui attribuer des droits d'accès **private**.

Added file *nom_fichier* does not begin correctly, ignored*Mise en garde du bibliothécaire*

Le fichier indiqué ne correspond pas à un module objet et n'est donc ajouté à la bibliothèque. La bibliothèque est quand même créée.

Address of overloaded function *fonction* doesn't match *type**Erreur du compilateur*

Une variable ou un paramètre est affecté ou initialisé avec l'adresse d'une fonction surchargée. Le type de la variable ou du paramètre ne correspond à aucun des types des fonctions surchargées dont le nom est spécifié.

module already in LIB, not changed!*Mise en garde du bibliothécaire*

Vous avez essayé d'ajouter un objet à une bibliothèque (opération +) mais un objet portant le même nom existe déjà dans cette bibliothèque. Pour mettre à jour le module, utilisez l'action +-. La bibliothèque n'a pas été modifiée.

Ambiguity between *fonction1* and *fonction2**Erreur du compilateur*

Les fonctions surchargées indiquées peuvent toutes deux être utilisées avec les paramètres fournis. Cette ambiguïté n'est pas autorisée.

Ambiguous member name *nom**Erreur du compilateur*

Un membre de structure en assembleur en ligne doit être désigné par un seul nom. S'il est défini dans plusieurs structures, les définitions correspondantes doivent indiquer le même type et le même décalage au sein de ces structures. Le nom de membre est dans ce cas ambigu. Utilisez de préférence la syntaxe `(struct xxx).yyy`.

Ambiguous Override of Virtual Base Member *fonction1:fonction2**Erreur du compilateur*

Une fonction virtuelle dans une classe de base virtuelle a été supplantée par deux fonctions différentes ou plus selon différents chemins de la hiérarchie d'héritage.

Ambiguous operators need parentheses*Mise en garde du compilateur*

Ce message s'affiche lorsque deux opérateurs de décalage, relationnels ou booléens bit à bit sont utilisés sans parenthèses. Un opérateur d'addition ou de soustraction spécifié sans parenthèses avec un opérateur de décalage génère également ce message. Veillez à respecter la priorité de ces opérateurs.

Ambiguous override of virtual base member *fonction_base:fonction_dérivée**Erreur du compilateur*

Ce message est généré lorsqu'une fonction virtuelle définie dans une classe de base virtuelle est supplantée par différentes fonctions ayant deux classes dérivées dans la même hiérarchie d'héritage. Par exemple :

```
struct VB
{
    virtual f();
};

struct A:virtual VB
{
    virtual f();
};

struct B:virtual VB
    virtual f();
```

```
};

struct D:A,B
{
} //erreurs ICI
```

Le code ci-dessus est marqué avec les messages d'erreurs suivants :

```
Error: Ambiguous override of virtual base member VB::f():A::f()
Error: Ambiguous override of virtual base member VB::f():B::f()
```

Application load & execute error 0001

Erreur du compilateur

Application load & execute error FFEO

Erreur du compilateur

Mémoire étendue insuffisante pour le chargement de l'outil de ligne de commande en mode protégé.

Array allocated using new may not have an initializer

Erreur du compilateur

A l'initialisation d'un vecteur (tableau) de classes, vous devez impérativement utiliser le *constructeur* par défaut, c'est-à-dire le constructeur sans argument.

Array bounds missing]

Erreur du compilateur

Un tableau a été incorrectement déclaré dans le fichier source. Ajoutez le crochet droit pour limiter le tableau.

Array must have at least one element

Erreur du compilateur

En langage ANSI C et C++, un tableau doit comporter au moins un élément (il est impossible de définir un objet de taille nulle). Une astuce de programmation consiste à déclarer un élément "tableau" de taille nulle dans une structure, puis d'affecter l'espace nécessaire à l'aide de *malloc*. Vous pouvez recourir à cette astuce mais devez impérativement déclarer l'élément "tableau" avec (au moins) un élément si vous compilez strictement en mode ANSI. Naturellement, il est toujours possible de déclarer (à ne pas confondre avec définir) un tableau de taille inconnue.

Par exemple,

```
char ray[];           /* définition avec taille inconnue -- illégal */
char ray[0];         /* définition avec taille nulle -- illégal */
extern char ray[];   /* déclaration avec taille inconnue -- ok */
```

Array of references is not allowed

Erreur du compilateur

Il est impossible de déclarer un tableau de références dans la mesure où les pointeurs sur référence ne sont pas autorisés et que les noms de tableau doivent obligatoirement être désignés par des pointeurs.

Array size for 'delete' ignored

Mise en garde du compilateur

Selon les dernières spécifications de programmation en langage C++, il n'est plus nécessaire de spécifier la taille d'un tableau pour le supprimer. Pour compiler du code rédigé selon des spécifications plus anciennes, Borland C++ ne tient pas compte de cette compilation et génère cet message de mise en garde.

Array size too large

Erreur du compilateur

La taille du tableau déclaré dépasse la limite de 64 ko.

Array variable *identificateur* is near*Mise en garde du compilateur*

Si vous utilisez les options **-Ff** ou **-Fm** dans la ligne de commande ou les options EDI Options|Compiler|Advanced Code Generation...|Far Data Threshold selection pour définir la limite seuil, les variables globales dépassant ce seuil sont converties en données far par le compilateur. Par contre, si la variable est un tableau initialisé avec une taille inconnue, sa taille totale n'est pas connue lorsque le compilateur doit décider de la convertir en données near ou far. Cette variable est alors définie comme near. Si la taille totale de la variable dépasse le seuil en raison du nombre de variables d'initialisation données pour le tableau, le compilateur génère ce message. Si le fait que le compilateur définisse la variable en variable near pose problème (par exemple, si l'éditeur de liens indique un débordement de groupe dû à un nombre trop important de données globales), vous devez la définir explicitement en tant que variable far en insérant le mot clé **far** immédiatement à gauche du nom de la variable dans sa définition.

Assembler statement too long*Erreur du compilateur*

La longueur maximale d'une instruction en assembleur en ligne ne doit pas dépasser 480 octets.

Assigning type to énumération*Mise en garde du compilateur*

Affectation d'une valeur entière à un type **enum**. Bien qu'il s'agisse d'une erreur en C++, ce message a valeur de mise en garde pour que le programme puisse quand même s'exécuter.

Assignment to this not allowed, use X::operator new instead*Erreur du compilateur*

Dans des versions précédentes du C++, la seule façon de contrôler l'allocation d'une classe d'objets consistait à affecter le paramètre **this** dans un constructeur. Cette méthode n'est plus autorisée. Désormais, il est préférable et plus fiable de définir une fonction membre **operator new**.

Attempt to export non-public symbol *symbole**Erreur de l'éditeur de liens*

Ce message s'affiche en général lorsqu'un fichier .DEF spécifie un EXPORT pour un symbole que vous avez soit oublié de définir ou que vous avez mal orthographié.

Attempt to grant or reduce access to *identificateur**Erreur du compilateur*

Une classe dérivée en C++ peut modifier les droits d'accès d'un membre d'une classe de base, mais uniquement en rétablissant les droits de la classe de base. Elle ne peut pas élargir ou restreindre ces droits d'accès.

Attempting to return a reference to a local object*Erreur du compilateur*

Dans une fonction renvoyant un type référence, vous avez essayé de renvoyer une référence à un objet temporaire (peut-être le résultat d'un constructeur ou d'un appel de fonction). Dans la mesure où cet objet n'existera plus une fois la fonction exécutée, la référence sera donc illégale.

Attempting to return a reference to local variable *identificateur**Erreur du compilateur*

Cette fonction C++ renvoie un type référence ou vous essayez de renvoyer une référence à une variable locale automatique. Ceci n'est pas autorisé car la variable désignée n'existera plus à la fin du traitement de la fonction. Vous pouvez renvoyer une référence à une variable statique ou globale ou bien modifier la fonction pour qu'elle renvoie une valeur à la place.

Bad call of intrinsic function*Erreur du compilateur*

Vous avez utilisé une fonction intrinsèque sans fournir de prototype ou le prototype fourni pour une fonction intrinsèque ne correspondait pas à celui attendu par le compilateur.

Bad character in parameters*Erreur de l'éditeur de liens*

Vous avez spécifié l'un des caractères suivants (ou un caractère autre que le caractère de tabulation horizontale, de saut de ligne, de retour chariot ou Ctrl+Z) dans la ligne de commande ou un fichier de recherche :

* < = > ? [] |

Bad define directive syntax*Erreur du compilateur*

Une définition de macro débute ou se termine par l'opérateur **##** ou contient l'opérateur **#** non suivi d'un nom d'argument de macro.

Bad field list in debug information in module *module**Erreur de l'éditeur de liens*

Ce message est généralement dû à de mauvaises informations de débogage dans le fichier OBJ. Prenez-en note et contactez votre support technique Borland.

Bad file name *nom_fichier**Erreur de l'éditeur de liens*

Le nom de fichier passé en argument à l'éditeur de liens est incorrect.

Bad file name format in include directive*Erreur du compilateur*

Un nom de fichier d'inclusion doit figurer entre guillemets ("FICHIER.H") ou entre parenthèses angulaires (<FICHIER.H>). Le premier signe ouvrant est manquant au début du nom de fichier spécifié. En cas d'utilisation d'une macro, le texte développé est interprété incorrectement (puisqu'il ne figure ni entre guillemets ni entre parenthèses angulaires).

Bad filename format in include statement*Erreur de MAKE*

Un nom de fichier d'inclusion doit figurer entre guillemets ou entre parenthèses angulaires. Le premier signe ouvrant est manquant au début du nom de fichier.

Bad file name format in line directive*Erreur du compilateur*

Un nom de fichier de directives doit figurer entre guillemets ("FICHIER.H") ou entre parenthèses angulaires (<FICHIER.H>). Le premier signe ouvrant est manquant au début du nom de fichier spécifié. En cas d'utilisation d'une macro, le texte développé est interprété incorrectement (puisqu'il ne figure ni entre guillemets ni entre parenthèses angulaires).

Bad GCD type in GRPDEF, extended dictionary aborted*Mise en garde du bibliothécaire***Bad GRPDEF type encountered, extended dictionary aborted***Mise en garde du bibliothécaire*

Le bibliothécaire a détecté une entrée incorrecte dans un enregistrement de définition de groupe (GRPDEF) dans un module objet lors d'une tentative de création de dictionnaire étendu. Le bibliothécaire et l'éditeur de liens reconnaissent uniquement des enregistrements GRPDEF de type indice "index de segment". Le bibliothécaire ne peut pas créer le dictionnaire étendu s'il détecte un autre type d'enregistrement. Remarquez qu'un module objet créé à l'aide d'autres outils que ceux fournis par Borland peut créer des enregistrements GRPDEF d'autres types. Ce message peut également être généré par un module objet défectueux.

Bad header in input LIB*Erreur du bibliothécaire*

Le bibliothécaire a détecté un en-tête de bibliothèque incorrect dans la bibliothèque à laquelle des modules objets doivent être ajoutés. Recréez la bibliothèque.

Bad ifdef directive syntax*Erreur du compilateur*

Une directive **#ifdef** doit contenir un seul et unique identificateur.

Bad LF_POINTER in module *module**Erreur de l'éditeur de liens*

Ce message est généralement dû à de mauvaises informations de débogage dans le fichier OBJ. Prenez-en note et contactez votre support technique Borland.

Bad macro output translator*Erreur de MAKE*

Erreur de syntaxe de substitution dans une ou plusieurs macros. Par exemple :

```
$ (MODEL:=s) or $(MODEL:) or $(MODEL:s)
```

- Bad object file *nom_fichier* near file offset *décalage*** *Erreur de l'éditeur de liens*
L'éditeur de liens a détecté un fichier OBJ incorrect. Ce message est généralement dû à une erreur du convertisseur.
- Bad object file record in library file *nom_fichier* in module *module* near module file offset 0xXXXXXXXX**
- Bad object file record in module *nom_fichier* near module file offset 0xXXXXXXXX** *Erreur de l'éditeur de liens*
Un fichier objet mal formé a été détecté. Ce message s'affiche en général lorsqu'il a été fait appel à un fichier source ou un fichier objet qui n'a pas été complètement compilé. Il s'affiche également si la machine a été réinitialisée lors d'une compilation ou si le compilateur n'a pas supprimé le fichier objet de sortie lorsque la combinaison *Ctrl+Pause* a été actionnée.
- Bad OMF record type *type encountered* in module *module*** *Erreur du bibliothécaire*
Le bibliothécaire a détecté un enregistrement OMF (Object Module Format) incorrect à la lecture du module objet. Les enregistrements d'en-tête du *module* ayant déjà été lus et vérifiés, ce message indique en général que le module objet a été endommagé et qu'il doit être recréé.
- Bad syntax for pure function definition** *Erreur du compilateur*
Vous devez suffixer "= 0" à la déclaration d'une fonction virtuelle pure. Corrigez la définition.
- Bad undef directive syntax** *Erreur du compilateur*
Une directive **#undef** doit contenir un seul et unique identificateur.
- Bad undef statement syntax** *Erreur de MAKE*
Une instruction **!undef** doit contenir un seul et unique identificateur.
- Bad version number in parameter block** *Erreur de l'éditeur de liens*
Incohérence interne à l'environnement intégré. Quittez l'EDI et relancez-le. Cette erreur ne se produit pas dans la version en mode autonome.
- Base class *classe* contains dynamically dispatchable functions** *Erreur du compilateur*
Dans la version actuelle de Borland C++, les tables DDVT incompatibles avec l'utilisation de l'héritage multiple. Ce message s'affiche lorsqu'une classe contenant des fonctions DDVT essaie d'hériter de fonctions DDVT appartenant à plusieurs classes parents.
- Base class *classe* is inaccessible because also in *classe*** *Mise en garde du compilateur*
Il est impossible d'utiliser une classe à la fois comme classe de base directe et indirecte pour des raisons d'ambiguïté des membres. Essayez de déclarer la classe de base virtuelle aux deux endroits.
- Base class *classe* is included more than once** *Erreur du compilateur*
Une classe C++ peut être dérivée à partir d'un nombre indéfini de classes de base. Par contre, elle ne peut être dérivée directement à partir d'une classe donnée qu'une seule fois.
- Base class *classe* is initialized more than once** *Erreur du compilateur*
Dans un constructeur de classe C++, la liste des initialisations après l'en-tête de constructeur comporte plusieurs fois la classe de base *classe*.
- Base initialization without a class name is now obsolete** *Erreur du compilateur*
Dans des versions précédentes de C++, l'initialisation d'une classe de base consistait à ajouter simplement la liste des paramètres du constructeur de classe de base après l'en-tête de constructeur. Dans les versions actuelles, il est recommandé d'inclure le nom de la classe de base.

Cette nouvelle syntaxe clarifie le code et doit être utilisée en cas de classes de base multiples.

Ancienne syntaxe :

```
derived::derived(int i) : (i, 10) { ... }
```

Nouvelle syntaxe :

```
derived::derived(int i) : base(i, 10) { ... }
```

Bit field cannot be static

Erreur du compilateur

Seules les données membres de classe C++ ordinaires peuvent être déclarées **statiques**. Ce type de déclaration est impossible pour les champs de bits.

Bit field too large

Erreur du compilateur

Un champ de bits de plus de 32 bits a été spécifié.

Bit fields must be signed or unsigned int

Erreur du compilateur

En ANSI C, les champs de bits peuvent avoir uniquement les types **signed** ou **unsigned int** (les type **char** ou **long**, par exemple, ne sont pas autorisés).

Bit fields must be signed or unsigned int

Mise en garde du compilateur

En ANSI C, les champs de bits ne peuvent avoir le type **signed char** ou **unsigned char**. Si vous ne compilez pas strictement en mode ANSI, le compilateur autorise de telles compilations mais les signale à l'aide cette mise en garde.

Bit fields must contain at least one bit

Erreur du compilateur

S'il est impossible de déclarer un champ de bits nommé avec 0 bits ou moins, il est par contre possible de déclarer un champ de bits non nommé avec 0 bits. Cette convention permet de forcer le champ de bits suivant à s'aligner sur une frontière de type octet (ou de type mot si vous avez sélectionné l'alignement correspondant). En C++, les champs de bits (ainsi que les énumérations) doivent être de type entier.

Bit fields must have integral type

Erreur du compilateur

En C++, les champs de bits (ainsi que les énumérations) doivent être de type entier.

Body has already been defined for function *fonction*

Erreur du compilateur

Un corps de fonction a déjà été spécifié pour une fonction définie avec le même nom et le même type. Un corps de fonction donné ne peut être spécifié qu'une seule fois.

Both return and return with a value used

Mise en garde du compilateur

La fonction courante comporte des instructions **return** avec et sans valeurs. Cela est autorisé en C mais génère presque toujours ce message. Une instruction **return** a peut-être été omise à la fin de la fonction.

Call of nonfunction

Erreur du compilateur

Le nom appelé n'est pas déclaré en tant que fonction. Vérifiez la déclaration de la fonction ou l'orthographe de son nom.

Call to function *fonction* with no prototype

Mise en garde du compilateur

Le message de mise en garde "Prototypes required" était activé et vous avez appelé la fonction *fonction* sans préciser d'abord un prototype pour cette fonction.

Call to undefined function *fonction*

Erreur du compilateur

Alors que la déclaration de la fonction courante dans le fichier source indique le renvoi d'un type autre que **void** en C++ (ou **int** en C), le compilateur a détecté un renvoi sans aucune valeur. Il s'agit en général d'une erreur. Les fonctions **int** ne sont pas concernées en C car dans des versions précédentes du langage C, le type **void** n'était pas disponible pour définir une fonction ne renvoyant aucune valeur.

virtual can only be used with member functions

Une donnée membre a été déclarée avec le spécificateur `virtual` réservé exclusivement aux fonctions membres. Exemple :

```
class myclass
{
public:
    virtual int a;    // erreur
};
```

Cannot access an inactive scope

Erreur du compilateur

Vous avez essayé d'évaluer ou d'accéder à une variable locale à une fonction qui n'est pas en cours d'activité. Ce message d'évaluation d'expression est interne au débogueur intégré.

Cannot add or subtract relocatable symbols

Erreur du compilateur

La seule opération arithmétique qui peut être effectuée sur un symbole rélogeable dans un opérande en assembleur est l'addition ou la soustraction d'une constante. Les variables, procédures, fonctions et étiquettes sont des symboles réaffectables. En supposant que *Var* soit une variable et *Const* une constante, les instructions

```
MOV AX, Const+Const
```

et

```
MOV AX, Var+Const
```

sont correctes ; l'instruction `MOV AX, Var+Var` ne l'est pas.

Cannot allocate a reference

Erreur du compilateur

Vous avez essayé de créer une référence avec l'opérateur `new`. Cette opération n'est pas autorisée car une référence n'est pas un objet et ne peut donc être créée à l'aide de cet opérateur.

identificateur cannot be declared in an anonymous union

Erreur du compilateur

Le compilateur a détecté une déclaration de fonction membre ou de membre statique dans une union anonyme. Ce type d'union peut uniquement contenir des données membres.

fonction1 cannot be distinguished from fonction2

Erreur du compilateur

Les listes de types de paramètres dans la déclaration de ces deux fonctions sont trop similaires pour pouvoir distinguer ces fonctions. Modifiez l'ordre des paramètres ou le type d'un paramètre dans l'une des déclarations.

Cannot call main from within the program

Erreur du compilateur

Il est impossible d'appeler la fonction `main` de façon récursive en C++.

Cannot call near class member function with a pointer of type type

Erreur du compilateur

Les fonctions membres de classes de type `near` (rappel : les classes sont de type `near` par défaut dans les modèles de mémoire `TINY`, `SMALL` et `MEDIUM`) ne peuvent pas être appelées à l'aide de pointeurs de membres de type `far` ou `huge`.

Remarquez que cela s'applique également à des appels utilisant un pointeur sur membre. Redéfinissez le type du pointeur à `near` ou déclarez la classe comme `far`.

Cannot cast from type1 to type2

Erreur du compilateur

Impossible de trans typer `type1` en `type2`. En C, un pointeur peut être converti en type entier ou en un autre pointeur. Un type entier peut être converti en type entier, flottant ou pointeur. Un type flottant peut être converti en type entier ou flottant. Les types `structure` et `tableau` ne peuvent être convertis ni être le type résultant d'une conversion. Il est impossible de convertir un type `void`.

En C++, si une instruction de conversion ou un constructeur défini par l'utilisateur est détecté, les règles de priorité de types s'appliquent (sauf pour les pointeurs vers membres de classe). Parmi les types entiers, seule une constante nulle peut être transtypée en pointeur membre. Un pointeur membre peut être transtypé en entier ou en pointeur membre similaire. Un pointeur membre similaire pointe vers une donnée membre ou vers un membre de fonction selon le comportement du pointeur initial. La classe qualifiante du type résultant de la conversion doit être identique à l'originale ou être une classe de base de l'originale.

Cannot convert *type1* to *type2*

Erreur du compilateur

L'exécution d'une instruction, d'une initialisation ou d'une expression dépend de la conversion de type spécifiée mais cette conversion n'est pas autorisée.

Cannot create instance of abstract class *classe*

Erreur du compilateur

Une classe abstraite (contenant des fonctions pures de type virtuel) n'est pas utilisable directement mais uniquement de façon dérivée.

Cannot define a pointer or reference to a reference

Erreur du compilateur

Il est impossible de définir un pointeur vers une référence ou une référence à une référence.

Cannot find *classe::classe*(*classe* &) to copy a vector

Erreur du compilateur

Si une classe *classe1* contient un vecteur (tableau) de classe *classe2*, et que vous vouliez construire un objet de type *classe1* à partir d'un autre objet de type *classe1*, vous devez définir un constructeur `classe2::classe2(classe2&)` pour pouvoir construire les éléments de ce vecteur. Ce constructeur, appelé constructeur de copie, accepte une référence à sa classe comme unique paramètre.

En règle générale, le compilateur fournit automatiquement un constructeur de copie. Néanmoins, si vous avez défini un constructeur pour la classe *classe2* qui a un paramètre de type *classe2&* et d'autres paramètres à valeur par défaut, le constructeur de copie ne peut pas être généré par le compilateur. Ceci est dû au fait que `classe2::classe2(classe2&)` et `classe2::classe2(classe2&, int = 1)` ne peuvent être différenciés. Vous devez redéfinir ce constructeur afin qu'une partie seulement des paramètres aient une valeur par défaut. Vous pouvez créer un constructeur de copie ou laisser le compilateur en créer un.

Cannot find *classe::operator=(classe&)* to copy a vector

Erreur du compilateur

En C++, si une classe *classe1* contient un vecteur (tableau) de classe *classe2* et que vous vouliez copier une classe de type *classe1*, vous devez définir un opérateur d'affectation `classe2::opérateur=(classe2&)` afin que les éléments du vecteur puissent être copiés. En règle générale, le compilateur fournit cet opérateur automatiquement. Néanmoins, si vous avez défini un opérateur = pour la classe *classe2* (à condition que cet opérateur n'accepte pas de paramètre de type *classe2&*), vous devrez le spécifier explicitement car le compilateur ne le fournit pas automatiquement.

Cannot find default constructor to initialize array element of type *classe*

Erreur du compilateur

Lorsque vous déclarez un tableau d'une classe comportant des constructeurs, vous devez soit initialiser explicitement chaque élément du tableau, soit définir la classe avec un constructeur par défaut (celui-ci sera utilisé pour initialiser les éléments du tableau qui n'ont pas d'initialisateur explicite). Un constructeur par défaut est défini pour une classe par le compilateur à moins que vous n'ayez déjà défini un constructeur pour cette classe.

Cannot find default constructor to initialize base class *classe**Erreur du compilateur*

En C++, si une classe dérivée *classe2* est construite, chaque classe de base *classe1* doit être construite au préalable. Si le constructeur de *classe2* ne spécifie aucun constructeur pour *classe1* (inclus dans l'en-tête de *classe2*), vous devez définir un constructeur `classe1::classe1()` pour la classe de base. Ce constructeur sans paramètres est appelé le constructeur par défaut. Il est fourni automatiquement par le compilateur à moins que vous ayez déjà défini un constructeur pour la classe *classe1*. Dans ce cas, vous devez spécifier le constructeur par défaut explicitement car le compilateur ne le fournit pas automatiquement.

Cannot find default *constructeur* to initialize member *identificateur**Erreur du compilateur*

En C++, si une classe *classe1* contient un membre de la classe *classe2*, et que vous vouliez construire un objet de type *classe1*, vous devez définir un constructeur `classe2::classe2()` pour que le membre puisse être construit. Ce constructeur sans paramètre est appelé le constructeur par défaut. Il est fourni automatiquement par le compilateur à moins que vous ayez déjà défini un constructeur pour la classe *classe2*. Dans ce cas, vous devez spécifier le constructeur par défaut explicitement car le compilateur ne le fournit pas automatiquement.

Cannot find MAKE.EXE*Erreur de MAKE*

L'outil MAKE est introuvable. Vérifiez que le fichier MAKE.EXE figure dans le répertoire en cours ou dans un répertoire spécifié par la variable d'environnement PATH.

Cannot generate COM file: data below initial CS:IP defined*Erreur de l'éditeur de liens*

Ce message est dû à la génération de données de code en dessous de l'adresse de début (100 en général) d'un fichier .COM. Vérifiez que l'adresse de début se situe à 100 à l'aide de la commande (ORG 100H). Ce message d'erreur ne devrait pas s'afficher pour des programmes rédigés dans un langage de haut niveau. Si c'est pourtant le cas, vérifiez que le module objet de code de démarrage (COx) approprié est utilisé dans l'édition de liens.

Cannot generate COM file: invalid initial entry point address*Erreur de l'éditeur de liens*

L'option /Tdc ou /t a été utilisée mais l'adresse de début du programme ne correspond pas à l'adresse 100H (requis avec les fichiers .COM).

Cannot generate COM file: program exceeds 64K*Erreur de l'éditeur de liens*

L'option /Tdc ou /t a été utilisée mais la taille totale du programme dépasse la limite imposée aux fichiers .COM.

Cannot generate COM file: segment-relocatable items present*Erreur de l'éditeur de liens*

L'option /Tdc ou /t a été utilisée mais le programme contient des réadressages de segments non autorisés avec les fichiers .COM.

Cannot generate COM file: stack segment present*Erreur de l'éditeur de liens*

L'option /Tdc ou /t a été utilisée mais le programme déclare un segment de pile non autorisé avec les fichiers .COM.

Cannot generate *fonction* from template function *template**Erreur du compilateur*

Un appel de fonction *template* a été détecté, mais il est impossible de créer une fonction *template* à partir de la fonction appelée.

Cannot have a non-inline function in a local class*Erreur du compilateur***Cannot have a static data member in a local class***Erreur du compilateur*

Tous les membres des classes déclarées locales à une fonction doivent être entièrement définies dans la définition de classe. En d'autres termes, ce type de classes locales ne peut contenir aucune données membres statiques et les corps de toutes les fonctions membres de ces classes doivent être définies dans la définition de classe.

Cannot have multiple paths for implicit rule*Erreur de MAKE*

Chaque extension dans une règle implicite doit être désignée par un seul chemin. Des listes de plusieurs chemins sont autorisées uniquement pour des dépendants dans une règle explicite. Par exemple :

```
{path1;path2}.c.obj:    # Incorrect
{path}.c.obj           # Correct
```

Cannot have path list for target*Erreur de MAKE*

Vous pouvez spécifier une liste de chemins uniquement pour des dépendants dans une règle explicite. Par exemple :

```
{path1;path2}prog.exe: prog.obj    # Incorrect
prog.exe: {path1;path2}prog.obj    # Correct
```

Cannot initialize a class member here*Erreur du compilateur*

Les membres individuels d'une **structure**, d'une **union** ou d'une **classe** ne peuvent être définis avec des initialisateurs. Une **structure** ou une **union** peut être initialisée globalement à l'aide d'initialisateurs s'ils figurent entre accolades. Une **classe** C++ peut être initialisée uniquement à l'aide d'un constructeur.

Cannot initialize type1 with type2*Erreur du compilateur*

Impossible d'initialiser l'objet de type *type1* avec la valeur de type *type2*. Les règles d'initialisation sont pour l'essentiel identiques à celle de l'affectation.

Cannot modify a const object*Erreur du compilateur*

Opération non autorisée sur un objet déclaré comme constante (comme une affectation à cet objet).

Cannot overload 'main'*Erreur du compilateur*

Impossible de surcharger la fonction *main*.

fonction cannot return a value*Erreur du compilateur*

Une fonction définie avec le type de valeur renvoyée **void** contient une instruction **return** qui renvoie une valeur d'un autre type (par exemple une valeur de type **int**).

identificateur cannot start a parameter declaration*Erreur du compilateur*

Un *identificateur* non défini a été détecté au début d'un argument dans un déclarateur de fonction.

Ce message indique souvent une erreur d'utilisation de fichier d'en-tête ou que le nom de type a été mal orthographié ou que le type n'a pas été déclaré.

identificateur cannot start an argument declaration*Erreur du compilateur*

Un *identificateur* non défini a été détecté au début d'un argument dans un déclarateur de fonction. Ce message indique souvent une erreur d'utilisation de fichier d'en-tête ou que le nom de type a été mal orthographié ou que le type n'a pas été déclaré.

Cannot take address of main*Erreur du compilateur*

En C++, il est impossible de récupérer l'adresse de la fonction *main*.

Cannot throw type -- ambiguous base class base*Erreur du compilateur*

Il est impossible d'associer à une clause **throw** une classe qui contient plusieurs copies d'une classe de base non virtuelle.

Cannot write GRPDEF list, extended dictionary aborted*Erreur du bibliothécaire*

Impossible d'ajouter le dictionnaire étendu à la fin du fichier de bibliothèque. Ce message indique en général un espace insuffisant sur le disque.

Can't grow LE/LIDATA record buffer*Erreur du bibliothécaire*

Erreur de ligne de commande. Reportez-vous au message **Out of memory reading LE/LIDATA record from object module**.

Case bypasses initialization of a local variable*Erreur du compilateur*

En C++, il est impossible d'omettre l'initialisation d'une variable locale. Dans ce cas, une étiquette **case** peut transférer le contrôle après cette variable locale.

Case outside of switch*Erreur du compilateur*

Le compilateur a détecté une instruction **case** à l'extérieur d'une instruction **switch**. Un erreur d'accolades est souvent à l'origine de ce message.

Case statement missing :*Erreur du compilateur*

Une instruction **case** doit comporter une expression constante suivi d'un signe deux-points (:). Vérifiez que cette expression se termine par un signe deux-points ou qu'aucun autre symbole n'a été spécifié avant ce signe.

catch expected*Erreur du compilateur*

Dans un programme en C++, un bloc *try* doit être suivi par au moins un bloc *catch*.

Character constant must be one or two characters long*Erreur du compilateur*

La longueur d'une constante de type caractère est limitée à un ou deux caractères uniquement.

Circular dependency exists in makefile*Erreur de MAKE*

Le fichier MAKE indique qu'un fichier doit être mis à jour *avant* de pouvoir être construit. L'exemple suivant utilise les règles explicites

```
fichiera: fichierb
fichierb: fichierc
fichierc: fichiera
```

Selon ces règles, le fichiera dépend du fichierb, qui dépend à son tour du fichierc, lequel dépend finalement du fichiera. Ces règles sont incorrectes dans la mesure où un fichier ne peut dépendre indirectement ou directement de lui-même.

Class classe may not contain pure functions*Erreur du compilateur*

La classe déclarée ne peut être abstraite et ne peut donc pas contenir de fonctions pures.

Class member membre declared outside its class*Erreur du compilateur*

Les fonctions membres d'une classe C++ peuvent être déclarées uniquement à l'intérieur de la déclaration de cette classe. A la différence des fonctions non membres, elles ne peuvent pas être déclarées à plusieurs reprises ou en d'autres endroits d'un programme.

Code has no effect*Mise en garde du compilateur*

Le compilateur a détecté une instruction dont les opérateurs n'ont aucun effet. Par exemple, l'instruction

```
a + b;
```

n'a d'effet sur aucune des variables. L'opération est superflue et ce message indique un bug dans le fichier.

Colon expected*Erreur de MAKE*

Signe deux-points (:) manquant à la fin de la règle implicite.

```
.c.obj:      # Correct
.c.obj      # Incorrect
```

Command arguments too long*Erreur de MAKE*

La longueur de la chaîne d'arguments passés à la commande est supérieure à 511 caractères (limite imposée par DOS).

Command syntax error*Erreur de MAKE*

Ce message s'affiche dans les cas suivants :

- la première ligne de règle du fichier MAKE était précédée d'espaces blancs ;
- une règle implicite n'est pas conforme à la syntaxe `.ext.ext;` ;
- une règle explicite ne comporte pas de nom avant le caractère `;` ;
- une définition de macro ne comporte pas de nom avant le caractère `=`.

Command too long*Erreur de MAKE*

La longueur de la commande est supérieure à 512 caractères. Il peut être utile d'avoir recours à un fichier de recherche.

Common segment exceeds 64K*Erreur de l'éditeur de liens*

Le programme comporte plus de 64 ko de données de type `near` non initialisées. Essayez de déclarer certaines de ces données avec le type `far`.

Compiler could not generate copy constructor for class *classe**Erreur du compilateur*

Le compilateur ne peut pas générer de constructeur de copie pour des raisons de règles linguistiques.

Compiler could not generate default constructor for class *classe**Erreur du compilateur*

Le compilateur ne peut pas générer de constructeur par défaut pour des raisons de règles linguistiques.

Compiler could not generate operator= for class *classe**Erreur du compilateur*

Le compilateur ne peut pas générer un opérateur d'affectation requis pour des raisons de règles linguistiques.

Compiler table limit exceeded*Erreur du compilateur*

Débordement de l'une des tables internes du compilateur dû en général à un nombre trop important de corps de fonction dans le module compilé. Vous ne résoudrez pas ce problème en libérant simplement de la mémoire ne permet pas de résoudre ce problème, vous devez simplifier le fichier à compiler.

Compound statement missing }*Erreur du compilateur*

Le compilateur a atteint la fin du fichier source sans trouver d'accolade fermante. Vérifiez les différents niveaux d'accolades.

Condition is always false*Mise en garde du compilateur***Condition is always true***Mise en garde du compilateur*

Le compilateur a détecté une comparaison de valeurs dont le résultat est toujours vrai ou faux, comme dans l'exemple suivant :

```
void proc(unsigned x) {
    if (x >= 0) { /* toujours 'true' */
    }
}
```

Conflicting type modifiers*Erreur du compilateur*

Ce message signale une déclaration où un pointeur est défini en même temps avec les types `near` et `far`. Un pointeur ne peut être défini qu'avec un seul modificateur d'adressage et une fonction ne peut être définie qu'avec un seul modificateur de langage (`cdecl`, `pascal` ou `interrupt`).

symbole **conflicts with module** *module in module* *module*

Mise en garde de l'éditeur de liens

Incohérence dans la définition du *symbole* indiqué. TLINK a détecté une fonction virtuelle et une définition commune portant le même nom.

Constant expression required

Erreur du compilateur

Un tableau doit être déclaré avec une taille constante. Cette erreur est généralement due à une constante mal orthographiée dans une directive **#define**.

Constant is long

Mise en garde du compilateur

Le compilateur a détecté une constante décimale supérieure à 32 767 ou une constante octale (ou hexadécimale) supérieure à 65 535 non suivie du caractère *l* ou *L*. La constante est interprétée comme étant de type **long**.

Constant member *membre in class without constructors*

Erreur du compilateur

Une classe contenant des membres de type constante doit être définie avec au moins un constructeur défini par l'utilisateur afin de pouvoir initialiser ces membres.

Constant member *membre is not initialized*

Mise en garde du compilateur

Cette classe C++ contient un membre de type constante non initialisé. Remarquez qu'il est seulement possible d'initialiser un membre de type constante, toute affectation est impossible.

Constant out of range in comparison

Mise en garde du compilateur

Le fichier source comporte une comparaison avec une sous-expression de type constant en dehors des valeurs limites autorisées par le type de l'autre sous-expression. Par exemple, la comparaison d'une quantité **unsigned** à -1 n'a pas de sens. Pour obtenir une constante **unsigned** supérieure à 32 767 (en décimal), vous devez transtyper la constante en **unsigned** ou ajouter la lettre *u* ou *U* après la constante.

Si ce message s'affiche, le compilateur génère quand même le code pour effectuer la comparaison. Si ce code produit toujours le même résultat (comme comparer une expression de type **char** à 4000), le code effectuera toujours ce test.

Constant variable *variable must be initialized*

Erreur du compilateur

Cet objet C++ est déclaré comme constante mais n'est pas initialisé. Comme il est impossible de lui affecter une valeur, l'objet doit être initialisé lors de sa déclaration.

Constructor cannot be declared const or volatile

Erreur du compilateur

Un constructeur ne peut pas être déclaré **const** et/ou **volatile**.

Constructor cannot have a return type specification

Erreur du compilateur

Le type de la valeur de renvoi des constructeurs C++ utilisés par le compilateur est implicite. Il est donc impossible de déclarer un type de renvoi ou de renvoyer une valeur.

Conversion may lose significant digits

Mise en garde du compilateur

Pour un opérateur d'affectation ou une autre opération, vous devez effectuer une conversion de type **long** ou **unsigned long** en type **int** ou **unsigned int**. Ce type de conversion peut modifier le fonctionnement et le résultat d'un programme puisque les variables de type **int** et **long** sont de tailles différentes.

Conversion of near pointer not allowed

Erreur du compilateur

Impossible de convertir un pointeur *near* en pointeur *far* dans la boîte d'évaluation d'expression lorsque le programme n'est pas en cours d'exécution. La conversion ne dispose pas de la valeur courante de DS générée dans le programme utilisateur.

Conversion operator cannot have a return type specification

Erreur du compilateur

La fonction membre de conversion de types C++ spécifie un type de valeur de renvoi différent de son propre type. La fonction de conversion **operator** ne peut être déclarée avec un type de valeur de renvoi.

- Conversion to type will fail for members of virtual base classe** *Erreur du compilateur*
Ce message s'affiche dans les quelques cas de conversion du type d'un pointeur membre dans le type d'un autre pointeur membre, si la classe du pointeur membre contient des classes virtuelles, et uniquement si l'option **-Vv** ou les options EDI Options|Compiler|Advanced Compiler|Deep Virtual Bases ont été utilisées. Il indique que si le pointeur membre transtypé pointe vers un membre de la classe **classe** (au moment du transtypage), la conversion ne peut être effectuée et le transtypage génère un pointeur membre NULL.
- Could not allocate memory for per module data** *Erreur du bibliothécaire*
Mémoire insuffisante pour le bibliothécaire.
- Could not create list file *nom_fichier*** *Erreur du bibliothécaire*
Impossible de créer le fichier de listage pour la bibliothèque. L'espace sur le disque est peut être saturé.
- Could not find a match for *argument(s)*** *Erreur du compilateur*
Aucune fonction C++ dont les paramètres correspondent aux paramètres fournis n'a pu être trouvée.
- Could not find file *nom_fichier*** *Erreur du compilateur*
Impossible de trouver le fichier spécifié dans la ligne de commande.
- Could not get procedure address from DLL *nom_fichier*** *Erreur de l'éditeur de liens*
L'éditeur de liens n'a pas pu extraire de procédure de la bibliothèque DLL indiquée. Vérifiez que vous disposez de la version appropriée de la bibliothèque DLL.
- Could not load DLL *nom_fichier***
L'éditeur de liens n'a pas pu charger la bibliothèque DLL spécifiée. Vérifiez que cette bibliothèque est déclarée dans la variable d'environnement PATH.
- Could not write output** *Erreur du bibliothécaire*
Le bibliothécaire n'a pas pu générer le fichier de sortie.
- Couldn't alloc memory for per module data** *Erreur du bibliothécaire*
Mémoire insuffisante pour le bibliothécaire.
- nom_fichier* couldn't be created, original won't be changed** *Mise en garde du bibliothécaire*
Tentative a été faite d'extraire un objet (action **'***) mais le bibliothécaire ne peut pas créer le fichier objet pour extraire le module. L'objet existe peut-être déjà et est accessible en lecture seulement ou le disque est saturé.
- Couldn't get LE/LIDATA record buffer** *Erreur du bibliothécaire*
Erreur de ligne de commande. Reportez-vous au message **Out of memory reading LE/LIDATA record from object module**.
- Debug info switch ignored for .COM files** *Mise en garde de l'éditeur de liens*
Borland C++ n'inclut pas les informations de débogage pour les fichiers .COM.
- Debug information enabled, but no debug information found in OBJs** *Mise en garde de l'éditeur de liens*
Aucune partie de l'application n'a été compilée avec des informations de débogage bien que vous ayez activé ces informations dans l'édition de liens.
- Debug information in module *module* will be ignored** *Mise en garde de l'éditeur de liens*
Les fichiers objets compilés avec des informations de débogage ont un enregistrement de version. La version principale de cet enregistrement est plus élevée que celle supportée actuellement par TLINK. TLINK n'a pas généré les informations de débogage du module en question.

Debugging information overflow; try fewer modules with debuginfo

Erreur de l'éditeur de liens

L'édition de liens comporte trop de modules contenant des informations de débogage. Recompiliez le programme avec moins de modules marqués pour des informations de débogage.

Declaration does not specify a tag or an identifier

Erreur du compilateur

Erreur de déclaration : variable ou code de compilation conditionnelle manquant dans la déclaration d'une structure ou d'une union. En C++, cette déclaration est obligatoire.

Declaration is not allowed here

Erreur du compilateur

Impossible d'utiliser une déclaration comme instruction de contrôle d'une instruction **while**, **for**, **do**, **if** ou **switch**.

Declaration missing ;

Erreur du compilateur

Point-virgule manquant à la fin d'une déclaration.

Declaration syntax error

Erreur du compilateur

Symbole manquant ou ajouté par erreur dans une déclaration.

Declaration terminated incorrectly

Erreur du compilateur

Caractère de terminaison incorrect ou ajouté par erreur dans une déclaration (par exemple, un point-virgule ajouté à la suite du corps d'une fonction). Ce message s'affiche également si un point-virgule figure par erreur entre l'en-tête et l'accolade ouvrante gauche dans la déclaration d'une fonction membre C++ dans une classe.

Declaration was expected

Erreur du compilateur

Déclaration manquante. Erreur due généralement à l'absence d'un délimiteur (virgule, point-virgule, parenthèse droite ou accolade droite).

Declare operator delete (void*) or (void*, size_t)

Erreur du compilateur

Déclarez l'opérateur **delete** avec un seul paramètre **void*** ou un second paramètre de type **size_t**. Si vous utilisez la seconde version, elle sera utilisée de préférence à la première. L'opérateur global **delete** ne peut être déclaré qu'avec le paramètre **void***.

Declare Operator delete[] (void*) or (void*, size_t)

Erreur du compilateur

Déclarez l'opérateur **delete** avec l'une des syntaxes suivantes :

- avec un seul paramètre **void*** ;
- avec un second paramètre de type **size_t**.

Si vous utilisez la seconde version, elle sera utilisée de préférence à la première. L'opérateur global **delete** ne peut être déclaré qu'avec le paramètre **void***.

Declare type prior to use in prototype

Mise en garde du compilateur

Lorsqu'un prototype de fonction fait référence à un type de structure qui n'a pas été déclaré au préalable, une déclaration à l'intérieur du prototype est différente d'une déclaration à l'extérieur du prototype. Par exemple :

```
int func(struct s *ps);
struct s { /* ... */};
```

Dans la mesure où il n'y a pas de structure **s** à la portée du prototype de la fonction **func**, le paramètre **ps** est un pointeur vers une structure **s** non définie qui ne doit pas être confondue avec la structure **s** déclarée sur la ligne suivante. Ceci génère des messages d'erreur et de mise en garde d'incompatibilité des types. Pour résoudre le problème, déplacez la déclaration de la structure **s** avant le prototype qui s'y réfère ou ajoutez la déclaration de type incomplète `struct s;` avant le prototype qui fait référence à la structure **s**. Si le paramètre de fonction est une structure, plutôt qu'un pointeur vers une structure, la déclaration incomplète ne suffit pas. Vous devez alors spécifier la déclaration de cette structure avant le prototype.

- Default argument value redeclared** *Erreur du compilateur*
Lorsqu'un paramètre de fonction C++ est déclaré avec une valeur par défaut, cette valeur ne peut pas être modifiée, redéclarée ou omise dans une autre déclaration de la même fonction.
- Default argument value redeclared for parameter** *paramètre* *Erreur du compilateur*
Lorsqu'un paramètre de fonction C++ est déclaré avec une valeur par défaut, cette valeur ne peut pas être modifiée, redéclarée ou omise dans une autre déclaration de la même fonction.
- Default expression may not use local variables** *Erreur du compilateur*
Une expression d'argument par défaut ne peut utiliser aucune variable locale ni aucun paramètre.
- Default outside of switch** *Erreur du compilateur*
Le compilateur a détecté une instruction **default** en dehors d'une instruction **switch**. Vérifiez les niveaux d'accolades.
- Default value missing** *Erreur du compilateur*
Lorsqu'une fonction C++ déclare un paramètre avec une valeur par défaut, tous les paramètres suivants doivent également avoir une valeur par défaut. Dans cette déclaration, un paramètre avec une valeur par défaut était suivi d'un paramètre sans valeur par défaut.
- Default value missing following parameter** *paramètre* *Erreur du compilateur*
Une valeur par défaut doit être spécifiée pour tous les paramètres suivant le premier paramètre qui a été défini avec une valeur par défaut.
- Define directive needs an identifier** *Erreur du compilateur*
Le premier caractère (différent d'un espace blanc) après une instruction **#define** doit être un identificateur. Le compilateur a détecté un autre caractère.
- symbole* **defined in module** *module is duplicated in module* *module* *Erreur de l'éditeur de liens*
Conflit entre deux symboles (de type "public" ou "communal") dû à la définition du même symbole dans deux modules. Une erreur se produit si ces définitions sont détectées en même temps dans le(s) fichier(s) objet car TLINK ne peut déterminer la définition appropriée. Un message de mise en garde s'affiche si TLINK détecte l'un des symboles dans une bibliothèque et l'autre symbole dans un fichier .OBJ. Dans ce cas, TLINK utilise celui du fichier .OBJ.
- Delete array size missing]** *Erreur du compilateur*
Crochet fermant manquant à la fin du spécificateur de tableau d'un opérateur.
- Destructor cannot be declared const or volatile** *Erreur du compilateur*
Déclaration d'un destructeur de type **const** et/ou **volatile** non autorisée.
- Destructor cannot have a return type specification** *Erreur du compilateur*
Il est impossible de spécifier le type "return" pour un destructeur.
- Destructor for classe is not accessible** *Erreur du compilateur*
Le destructeur de cette classe C++ est **protégé** ou **privé** et est donc inaccessible pour détruire la classe. Si un destructeur de classe est **privé**, la classe ne peut pas être détruite et ne peut donc jamais être utilisée. Il s'agit probablement d'une erreur. Un destructeur **protégé** est accessible uniquement à partir de classes dérivées. Ceci permet de vérifier qu'aucune instance d'une base n'a été créée mais uniquement les classes qui en dérivent.

Destructor for *classe* required in conditional expression*Erreur du compilateur*

Si le compilateur doit créer une variable locale temporaire dans une expression conditionnelle, elle est mal placée pour appeler le destructeur étant donné que cette variable peut ou non avoir été initialisée. La variable temporaire peut être créée explicitement, comme avec `nom-classe (val, val)` ou implicitement à l'aide d'une autre instruction. Révissez le code pour éliminer cette valeur temporaire.

Destructor name must match the class name*Erreur du compilateur*

Dans une classe C++, le tilde (`~`) introduit une déclaration de destructeur de classe. Le nom du destructeur doit correspondre au nom de la classe. Ce message indique que le tilde précédait un autre nom.

Divide error*Erreur d'exécution*

Impossible de diviser un entier par zéro. Par exemple :

```
int n = 0;
n = 2 / n;
```

Ce message peut être intercepté à l'aide de la fonction *signal*. Autrement, Borland C++ appelle la procédure d'interruption *abort* pour mettre fin au programme.

Division by zero*Erreur du compilateur*

Un opérateur de division ou de reste a été spécifié dans une expression constante avec un diviseur nul.

Division by zero*Mise en garde du compilateur*

Une expression avec opérateur de division ou de reste comporte un diviseur égal à zéro.

Division by zero*Erreur de MAKE*

Un opérateur de division ou de reste dans une instruction **!if** spécifie un diviseur égal à zéro.

do statement must have while*Erreur du compilateur*

Mot clé **while** manquant à la fin d'une boucle de répétition **do**.

nom_fichier does not exist - don't know how to make it*Erreur de MAKE*

Nom de fichier inexistant dans la séquence de compilation. Aucune règle ne permet la création du fichier `nom_fichier`.

DOS error, ax = numéro*Erreur de l'éditeur de liens*

Message généré si un appel DOS a renvoyé une erreur inattendue. La valeur `ax` affichée correspond à un code d'erreur. Il indique une erreur interne à TLINK ou une erreur DOS. Les appels DOS effectués par TLINK qui pourraient provoquer cette erreur sont `read`, `write`, `seek` et `close`.

do-while statement missing (*Erreur du compilateur*

Parenthèse gauche manquante après le mot clé **while** dans une boucle de répétition **do**.

do-while statement missing)*Erreur du compilateur*

Parenthèse droite manquante après l'expression de test dans une boucle de répétition **do**.

do-while statement missing ;*Erreur du compilateur*

Point-virgule manquant après la parenthèse droite dans l'expression de test d'une boucle de répétition **do**.

Duplicate case*Erreur du compilateur*

Les instructions **case** d'un branchement conditionnel **switch** doivent avoir des valeurs d'expression constante toutes différentes.

Duplicate Handler for *type1*, already had *type2**Erreur du compilateur*

Impossible de spécifier deux gestionnaires pour le même type.

- Duplicate ordinal for exports:** *string (ordval1) and string (ordval2)* *Erreur de l'éditeur de liens*
 Deux exports ont été trouvés pour le même symbole mais avec des valeurs ordinales différentes. Utilisez la même valeur ordinale ou supprimez l'un des exports.
- Empty LEDATA record in module** *module* *Mise en garde de l'éditeur de liens*
 Cette mise en garde s'affiche si le convertisseur émet un enregistrement de données contenant des données. Contactez le support technique du convertisseur utilisé. Cet enregistrement ne devrait pas entraîner d'effets secondaires.
- Enum syntax error** *Erreur du compilateur*
 Liste d'identificateurs mal formée dans une déclaration **enum**.
- Error changing file buffer size** *Erreur du bibliothécaire*
 Le bibliothécaire essaye d'ajuster la taille d'un tampon utilisé à la lecture ou l'écriture d'un fichier mais la mémoire est insuffisante. Pour résoudre ce problème, vous devrez libérer de la mémoire système.
- Error directive:** *message* *Erreur du compilateur*
 Affichage du texte de la directive **#error** en cours de traitement dans le fichier source.
- Error directive:** *message* *Erreur de MAKE*
 MAKE a traité une directive **#error** dans le fichier source et le texte de cette directive est affiché dans ce message.
- Error opening** *nom_fichier* *Erreur du bibliothécaire*
 Le bibliothécaire n'a pas pu ouvrir le fichier spécifié.
- Error opening** *nom_fichier for output* *Erreur du bibliothécaire*
 Le bibliothécaire ne peut pas ouvrir le fichier spécifié pour écrire la sortie. Ceci est généralement dû à un espace disque insuffisant pour la bibliothèque cible ou un fichier de listage. Cette erreur se produit à la fermeture du fichier cible qui est cependant marqué comme un fichier accessible en lecture seulement.
- Error renaming** *fichier_a to fichier_b* *Erreur du bibliothécaire*
 Le bibliothécaire crée une bibliothèque dans un fichier temporaire, puis attribue le nom du fichier de bibliothèque cible à ce fichier temporaire. Ce message s'affiche en cas de problème (dû généralement à un espace disque insuffisant).
- Error writing output file** *Erreur du compilateur*
 Une erreur DOS empêche Borland C++ d'écrire un fichier .OBJ, .EXE ou temporaire. Vérifiez que le répertoire de sortie est valide et que l'espace disque est suffisant.
- __except or __finally expected following __try** *Erreur du compilateur*
 En C, un bloc **__try** doit être suivi d'un bloc de traitement de zones **__except** ou **__finally**.
- Exception handling variable may not be used here** *Erreur du compilateur*
 Tentative a été faite d'utiliser l'une des valeurs de traitement d'exceptions réservées à des constructions de traitement d'exceptions particulières (comme `GetExceptionCode()`).
- Exception specification not allowed here** *Erreur du compilateur*
 Une déclaration de type de pointeur de fonction ne peut pas contenir de spécifications d'exceptions.
- Explicit stacks are ignored for PE images** *Mise en garde de l'éditeur de liens*
 Les applications 32 bits sont des applications au format PE qui ne disposent pas de piles explicites. Le segment de pile est lié dans l'image mais il n'est pas utilisé comme pile de l'application. Le paramètre de taille de pile est par contre utilisé pour définir la taille de la pile et le système d'exploitation alloue une pile à cette application.

Export *symbole* is already imported

Impossible d'exporter un symbole également importé.

Mise en garde de l'éditeur de liens

Export *symbole* is duplicated

Ce message indique un conflit d'exportation de deux fonctions distinctes portant le même nom. L'éditeur de liens ne peut déterminer quelle définition doit être exportée à l'aide de **__export**. Le premier symbole est utilisé par défaut.

Mise en garde de l'éditeur de liens

Expression expected

Expression manquante mais le symbole courant ne peut pas débiter une expression. Ce message s'affiche là où l'expression contrôlante d'une clause **if** ou **while** est manquante ou à l'endroit de l'initialisation d'une variable. Ceci est généralement dû à l'insertion ou la suppression accidentelle d'un symbole dans le code source.

Erreur du compilateur

Expression of scalar type expected

Les opérateurs de négation (!), d'incrémentement (++) et de décrémentation (-@hsp(0.2)-) requièrent une expression scalaire de type **char**, **short**, **int**, **long**, **enum**, **float**, **double**, **long double** et **pointer** uniquement.

Erreur du compilateur

Expression syntax

Le compilateur a détecté une erreur grave dans une expression (deux opérateurs consécutifs, un erreur de parenthèse ou un point-virgule manquant au niveau de l'instruction précédente sont les erreurs les plus fréquentes).

Erreur du compilateur

Expression syntax error in !if statement

L'instruction !if comporte une expression mal formée (parenthèse, opérateur ou constante manquant ou ajouté par erreur).

Erreur de MAKE

raison - extended dictionary not created

Le bibliothécaire n'a pas pu créer le dictionnaire étendu. La raison est indiquée dans le message.

Mise en garde du bibliothécaire

Extended dictionary not found in library *nom_bibliothèque*, /E option ignored

Avec l'option /E de TLINK, toutes les bibliothèques incluses dans l'édition de liens doivent avoir des dictionnaires étendus. Si une bibliothèque sans dictionnaire étendu est détectée lors d'une édition de liens avec l'option /E, l'éditeur de liens abandonne le traitement des dictionnaires étendus et procède à une édition de liens par défaut.

Mise en garde de l'éditeur de liens

Extern variable cannot be initialized

La classe de stockage **extern** appliquée à une variable signifie que cette variable est déclarée mais non définie à cet endroit. Aucune mémoire de stockage ne lui est allouée. Il est donc impossible d'initialiser la variable dans la déclaration.

Erreur du compilateur

Extern *symbole* was not qualified with **__import in module *module***

Les applications Windows 32 bits qui font référence à des symboles importés doivent effectuer des opérations d'indirection pour atteindre certaines données. Au niveau des appels, ces opérations sont gérées automatiquement par l'éditeur de liens. En ce qui concerne les références aux DATA importés, le compilateur doit générer une indirection pour que l'application fonctionne normalement. Le compilateur sait comment générer cette indirection lorsque le symbole est qualifié à l'aide de **__import**. Ce message s'affiche si l'éditeur de liens détecte une référence de segment à un symbole importé, et si le symbole n'a pas été qualifié à l'aide de **__import**.

Mise en garde de l'éditeur de liens

Extra argument in template class name *template*

Trop de valeurs ont été spécifiées dans la liste des paramètres formels du nom de classe template indiqué.

Erreur du compilateur

Extra parameter in call

Trop d'arguments ont été spécifiés dans l'appel d'une fonction effectué par l'intermédiaire d'un pointeur défini avec un prototype.

Erreur du compilateur

- Extra parameter in call to fonction** *Erreur du compilateur*
Trop d'arguments ont été spécifiés dans l'appel de la fonction indiquée (définie avec un prototype).
- Failed to locate DPML server (DPML16BI.OVL)** *Erreur du compilateur*
- Failed to locate protected mode loader (DPMILOAD.EXE)** *Erreur du compilateur*
Vérifiez que les fichiers DPML16BI.OVL et DPMILOAD.EXE sont accessibles à partir de la variable d'environnement PATH ou dans le même répertoire que l'outil de ligne de commande en mode protégé que vous avez essayé d'utiliser.
- Failed read from nom_fichier** *Erreur de l'éditeur de liens*
L'éditeur de liens n'a pas pu lire de données à partir du fichier indiqué.
- Failed write to nom_fichier** *Erreur de l'éditeur de liens*
L'éditeur de liens n'a pas pu écrire de données dans le fichier indiqué.
- __far16 may only be used with __pascal or __cdecl** *Erreur du compilateur*
Si vous utilisez **__far16** pour effectuer des appels de fonction ou faire référence à des données dans une bibliothèque DLL 16 bits, ces fonctions et données ne peuvent être modifiées que par **__pascal** ou **__cdecl**.]
- File must contain at least one external declaration** *Erreur du compilateur*
L'unité de compilation était logiquement vide et ne contenait aucune déclaration externe. En ANSI C et C++, l'unité de compilation requiert au moins une déclaration.
- Filename too long** *Erreur du compilateur*
La longueur du nom de fichier indiqué dans la directive **#include** est supérieure à 260 caractères (chemin d'accès compris). Le fichier n'a pas pu être traité par le compilateur.
- File name too long** *Erreur de MAKE*
Le nom du chemin du fichier indiqué dans une directive **!include** a fait déborder le tampon interne de MAKE (limité à une taille de 512 octets).
- nom_fichier file not found** *Mise en garde du bibliothécaire*
Le bibliothécaire de ligne de commande a essayé d'ajouter un objet inexistant. La bibliothèque a quand même été créée.
- nom_fichier file not found** *Erreur du bibliothécaire*
Pour créer une bibliothèque, l'EDI supprime la bibliothèque existante avant de la recréer. Si certains objets ne sont pas disponibles ou inexistant, la bibliothèque est considérée comme incomplète et TLIB génère ce message. L'absence d'un objet peut indiquer que la compilation du module source n'a pas été effectuée ou que des erreurs se sont produites lors de la compilation. Pour résoudre le problème, exécutez une opération **Compile|Make** ou **Compile|Build** ou signalez les endroits où se sont produites les erreurs.
- nom_fichier (num_ligne): file read error** *Erreur de l'éditeur de liens*
Une erreur DOS s'est produite lorsque TLINK a lu le fichier de définition de module. Ce message indique généralement un fin de fichier prématurée.
- Fixup to zero length segment in module module** *Erreur de l'éditeur de liens*
Une référence a été spécifiée après la fin d'un segment d'image. Comme cette référence accéderait à une adresse incorrecte, elle a été signalée comme une erreur.

Fixup overflow at segment :xxxxh, target = segment xxxh in module module

Mise en garde de l'éditeur de liens

Fixup overflow at segment:xxxxh, target = symbol in module module

Mise en garde de l'éditeur de liens

Ces messages indiquent une référence à un code ou à une donnée incorrecte dans un fichier objet que TLINK doit corriger lors de l'édition de liens.

Ceci est dû à une incompatibilité de modèles de mémoire. Un appel **near** d'une fonction dans un autre segment de code est la cause la plus probable. Cette erreur se produit également si vous générez un appel **near** à une variable de donnée ou une référence de donnée à une fonction. Dans un cas comme dans l'autre, le symbole indiqué comme *target* dans le message d'erreur correspond à la variable ou la fonction désignée. La référence figure dans le module indiqué. Accédez au fichier source de ce module pour corriger la référence.

Dans un programme en langage assembleur, des débordements de correction se produisent souvent si vous déclarez une variable externe au sein d'une définition de segment. Cette variable figure en fait dans un autre segment.

Si cette technique ne permet pas de diagnostiquer la cause du problème, ou si vous utilisez le langage assembleur ou un langage de haut niveau autre que Borland C++, ce message peut avoir d'autres origines. Même en Borland C++, ce message pourrait s'afficher si vous utilisez des noms de segment ou de groupe différents des valeurs par défaut d'un modèle de mémoire donné.

Fixup to zero length segment in module nom_module

Erreur de l'éditeur de liens

Cette erreur se produit généralement si vous effectuez une référence à un segment qui ne contient aucune donnée. Si le segment n'est pas groupé avec d'autres segments, le résultat est un segment physique de longueur nulle, ce qui est impossible en soi. L'éditeur de liens ne peut donc pas y faire référence.

Floating point error: Divide by 0.

Erreur d'exécution

Floating point error: Domain.

Erreur d'exécution

Floating point error: Overflow.

Erreur d'exécution

Ces erreurs fatales sont dues à une opération en virgule flottante dont le résultat n'est pas une valeur finie.

- "Divide by 0" indique que le résultat exact est +INF ou -INF (résultat de l'opération 1.0/0.0, par exemple).
- "Domain" indique que le résultat est égal à NAN, c'est-à-dire non égal à un nombre).
- "Overflow" indique que le résultat est +INF (infini) ou -INF avec perte complète de précision (par exemple, affectation de $1e200 \times 1e200$ à une variable de type **double**).

Floating point error: Partial loss of precision.

Erreur d'exécution

Floating point error: Underflow.

Erreur d'exécution

Ces exceptions sont masquées par défaut et les messages correspondants ne s'affichent pas. Les dépassements négatifs sont convertis en une valeur nulle et les pertes de précision sont ignorées. L'appel à `_control87` permet de les démasquer.

Floating point error: Stack fault.

Erreur d'exécution

Erreur de la pile en virgule flottante. Cette erreur peut être due à un code assembleur utilisant trop de registres ou à une fonction en virgule flottante déclarée incorrectement.

Ces erreurs peuvent être évitées en masquant les exceptions pour qu'elles ne se produisent pas ou en interceptant une exception avec *signal*. Pour plus de détails, reportez-vous aux fonctions `_control87` et *signal*.

Dans les cas précédents, le programme affiche le message d'erreur et fait appel à *abort* qui génère le message suivant :

```
Abnormal program termination
```

avant d'appeler `_exit (3)`. Pour plus de détails, reportez-vous aux fonctions *abort* et *_exit*.

- for statement missing (** *Erreur du compilateur*
Parenthèse gauche manquante après le mot clé **for** dans une boucle de répétition **for**.
- for statement missing)** *Erreur du compilateur*
Parenthèse droite manquante après l'une des expressions de contrôle d'une boucle de répétition **for**.
- for statement missing ;** *Erreur du compilateur*
Point-virgule manquant après l'une des expressions de contrôle d'une boucle de répétition **for**.
- Friends must be functions or classes** *Erreur du compilateur*
Seule une fonction ou une classe peut être définie comme **amie** d'une classe C++.
- Function call missing)** *Erreur du compilateur*
Erreur de syntaxe dans la liste d'arguments d'appel d'une fonction (parenthèse droite manquante ou mal placée, par exemple).
- Function calls not supported** *Erreur du compilateur*
Les appels de fonctions (y compris les fonctions de conversion implicites, les constructeurs, les destructeurs, les opérateurs surchargés et les fonctions inline) ne sont pas supportés dans l'évaluation d'expression du débogueur intégré.
- Function defined inline after use as extern** *Erreur du compilateur*
Une fois utilisée, une fonction ne peut pas devenir une fonction inline. Déplacez la définition inline plus bas dans le fichier ou supprimez-la entièrement.
- Function definition cannot be a Typedef'ed declaration** *Erreur du compilateur*
En ANSI C, un corps de fonction ne peut être défini à l'aide d'une instruction typedef avec une fonction Type.
- Function fonction cannot be static** *Erreur du compilateur*
Seuls les fonctions membres ordinaires et le opérateurs **new** et **delete** peuvent être déclarés statiques. Les constructeurs, destructeurs et autres opérateurs ne doivent pas être statiques.
- Function fonction should have a prototype** *Erreur du compilateur*
Une fonction a été appelée alors que son prototype était hors de portée.

En C, `int toto()` ; n'est pas un prototype à la différence de `int toto(int)` ; ou de `int toto(void)` ;. En C++, `int toto()` ; est considéré comme un prototype et est équivalent à `int toto(void)` ;. En C, les prototypes sont *recommandés* pour toutes les fonctions. En C++, les prototypes sont *obligatoires* pour toutes les fonctions. Dans tous les cas, une définition de fonction (constituée d'un en-tête et d'un corps de fonction) sert de prototype si elle apparaît avant toute mention de la fonction.
- Function should return a value** *Mise en garde du compilateur*
Cette fonction a été déclarée (peut-être implicitement) pour renvoyer une valeur. Le compilateur a détecté une instruction **return** sans valeur de renvoi ou a atteint la fin de la fonction sans trouver d'instruction **return**. Renvoyez une valeur ou déclarez la fonction comme **void**.

- Functions *fonction1* and *fonction2* both use the same dispatch number** *Erreur du compilateur*
 Problème de table virtuelle distribuée dynamiquement (table DDVT). Lorsque vous prenez le contrôle d'une fonction DDVT dans une classe dérivée, utilisez le même indice de distribution. Chaque fonction d'une même hiérarchie de classe doit utiliser un indice de distribution différent.
- Functions cannot return arrays or functions** *Erreur du compilateur*
 Une fonction ne peut pas renvoyer un tableau ou une fonction. Elle peut uniquement renvoyer un pointeur ou une référence à un tableau ou à une fonction.
- Functions containing local destructors are not expanded inline in function *fonction*** *Mise en garde du compilateur*
 Borland C++ ne met pas oeuvre l'inlining pour la fonction inline que vous avez créée. Ce message n'est pas critique. La fonction est générée hors ligne.
- Functions containing *mot réservé* are not expanded inline** *Mise en garde du compilateur*
 Une fonction contenant les mots réservés **do**, **for**, **while**, **goto**, **switch**, **break**, **continue** et **case** ne peuvent être développés en mode en ligne même s'ils sont spécifiés comme **inline**. La fonction reste parfaitement correcte mais sera traitée comme une fonction statique ordinaire (et non globale).
- Functions may not be part of a struct or union** *Erreur du compilateur*
 Le champ d'une **structure** ou d'une **union** (en langage C) a été déclaré comme fonction et non comme pointeur vers une fonction. Ce type de déclaration est uniquement autorisé en C++.
- General error** *Erreur de l'éditeur de liens*
- General error in library file *nom_fichier* in module *module* near module file offset 0xyyyyyyyy** *Erreur de l'éditeur de liens*
- General error in module *module* near module file offset0x yyyyyyy** *Erreur de l'éditeur de liens*
 L'éditeur de liens fournit autant d'informations que possible sur le traitement en cours au moment de l'erreur fatale. Notez les informations sur les fichiers .OBJ ou .LIB et contactez votre support technique Borland.
- General error** *Erreur de l'éditeur de liens*
- General error in module *module*** *Erreur de l'éditeur de liens*
 L'éditeur de liens a détecté une erreur interne. Notez les circonstances de cette erreur et contactez votre support technique Borland.
- Global anonymous union not static** *Erreur du compilateur*
 En C++, une union anonyme globale au niveau du fichier doit être déclarée statique.
- Goto bypasses initialization of a local variable** *Erreur du compilateur*
 En C++, il n'est pas autorisé d'omettre l'initialisation d'une variable locale. Ce message s'affiche si une instruction **goto** essaie de transférer le contrôle une fois la variable locale passée.
- Goto Into an Exception Handler is Not Allowed**
 Il est impossible de passer à un bloc try ou un gestionnaire d'exceptions associé à un bloc try.
- Goto statement missing label** *Erreur du compilateur*
 Le mot clé **goto** doit être suivi d'un identificateur.
- Group *groupe* exceeds 64K** *Erreur de l'éditeur de liens*
 La taille totale des segments combinés d'un groupe est supérieure à 64 ko.
- Group overflowed maximum size: *groupe*** *Erreur du compilateur*
 La taille totale des segments d'un groupe (par exemple, DGROUP) est supérieure à 64 ko.

Group *groupe1* overlaps group *groupe2**Mise en garde de l'éditeur de liens*

TLINK a détecté des groupes imbriqués. Ce message s'affiche lorsque des recouvrements sont utilisés.

Handler for *type1* hidden by previous handler for *type2**Mise en garde du compilateur*

Ce message s'affiche si un gestionnaire d'un type *D* dérivé d'un type *B* est spécifié après un gestionnaire d'un type *B*. Le gestionnaire de type *D* ne sera en effet jamais invoqué.

spécificateur* has already been includedErreur du compilateur*

Plusieurs occurrences de ce type de spécificateur sont spécifiées dans cette déclaration. Supprimez ou modifiez l'une de ces occurrences.

Hexadecimal value contains more than 3 digits*Mise en garde du compilateur*

Dans les versions précédentes de C, une séquence d'échappement hexadécimale pouvait comporter 3 chiffres maximum. La norme ANSI autorise l'utilisation d'un nombre quelconque de chiffres tant que la valeur peut se coder sur un octet. Ce message s'affiche si la séquence d'échappement hexadécimale est précédée de plusieurs zéros (comme dans "x00045"). Dans les versions précédentes de C, cette chaîne serait interprétée différemment.

fonction1* hides virtual function *fonction2*Mise en garde du compilateur*

Une fonction virtuelle d'une classe de base est généralement masquée par une déclaration dans une classe dérivée. Dans ce cas, une déclaration avec le même nom mais des types d'argument différents rend les fonctions virtuelles inaccessibles aux classes dérivées créées ultérieurement.

Identifiant expected*Erreur du compilateur*

Identificateur manquant. En C, un identificateur doit être spécifié dans une liste de paramètres d'un en-tête de fonction (spécifié selon les premières spécifications du C), après les mots réservés **struct** ou **union** s'ils ne sont pas suivis d'une accolade fermante, et comme nom d'un membre dans une structure ou une union (sauf pour les champs de bits de largeur nulle).

En C++, un identificateur doit être spécifié dans une liste de classes de base à partir desquelles une autre classe est dérivée, après deux signes deux-points (::) et après le mot réservé **operator** lorsqu'aucun symbole d'opérateur n'est présent.

Identifiant *identificateur* cannot have a type qualifier*Erreur du compilateur*

Impossible d'appliquer un qualificateur *classe::identificateur*. Vous ne pouvez pas définir de qualificateur pour des noms **typedef**, des déclarations de fonction (sauf des définitions au niveau du fichier), des variables locales ou des paramètres de fonction, ou pour un membre de classe sauf si vous utilisez sa propre classe comme qualificateur (ce qui est redondant mais autorisé).

If statement missing (*Erreur du compilateur*

Parenthèse gauche manquante après le mot clé **if** dans une instruction **if**.

If statement missing)*Erreur du compilateur*

Parenthèse droite manquante après l'expression test dans une instruction **if**.

If statement too long*Erreur de MAKE***ldef statement too long***Erreur de MAKE***lndef statement too long***Erreur de MAKE*

Une instruction **if**, **ldef** ou **lndef** comporte plus de 4096 caractères.

- Ignored module, path is too long** *Mise en garde du bibliothécaire*
 La longueur du chemin d'accès à un fichier .OBJ ou .LIB est supérieure à 64 caractères. Le bibliothécaire reconnaît uniquement des chemins de longueur inférieure ou égale à 64 caractères.
- Illegal ACBP byte in SEGDEF in module module** *Erreur de l'éditeur de liens*
 Ce message indique généralement une erreur de convertisseur.
- Illegal character caractère (0xvalue)** *Erreur du compilateur*
 Le compilateur a détecté un caractère incorrect dans le fichier d'entrée. La valeur hexadécimale du caractère incorrect est affichée. Cette erreur se produit également lorsque des paramètres supplémentaires sont passés à une macro de fonction.
- Illegal character in constant expression expression** *Erreur de MAKE*
 MAKE a détecté un caractère non autorisé dans une expression constante. Si le caractère est une lettre, ce message indique probablement un identificateur mal orthographié.
- Illegal component to GRPDEF in module module** *Erreur de l'éditeur de liens*
 Ce message indique généralement une erreur de convertisseur.
- Illegal group definition: groupe in module module** *Erreur de l'éditeur de liens*
 Cette erreur est due à un enregistrement GRPDEF mal formé dans un fichier .OBJ (fichier .OBJ créé par l'utilisateur ou présence d'un bug dans le convertisseur utilisé pour générer le fichier .OBJ). Si ce message s'affiche dans un fichier créé par Borland C++, recompilez le fichier. Si le problème persiste, contactez votre support technique Borland.
- Illegal initialization** *Erreur du compilateur*
 En C, une initialisation doit être une expression constante ou l'adresse d'une variable globale **externe** ou **statique** à laquelle une constante peut être ajoutée ou soustraite. *Erreur de MAKE*
- Illegal octal digit** *Erreur du compilateur*
 Le chiffre 8 ou 9 a été détecté dans une constante octale.
- Illegal parameter to @_hsp(0.2)_emit_@hsp(0.2)_** *Erreur du compilateur*
 L'argument passé à @_hsp(0.2)_emit_@hsp(0.2)_ n'est ni une constante ni une adresse.
- Illegal pointer subtraction** *Erreur du compilateur*
 Impossible de soustraire un pointeur à un non-pointeur.
- Illegal structure operation** *Erreur du compilateur*
 En C ou C++, une structure peut être utilisée avec les opérateurs d'accès (.), d'adressage (&) ou d'affectation (=) ou passée à partir ou à destination d'une fonction en tant que paramètre. En C ou C++, une structure peut également être utilisée avec des opérateurs surchargés. Le compilateur a détecté une structure utilisée avec un opérateur non autorisé.
- Illegal to take address of bit field** *Erreur du compilateur*
 Il est impossible de récupérer l'adresse d'un champ de bits bien qu'il soit possible de récupérer l'adresse d'autres types de champs.
- Illegal type of entry point** *Erreur de l'éditeur de liens*
 Seuls les points d'entrée qui ciblent un indice de segment sont reconnus.
- Illegal use of floating point** *Erreur du compilateur*
 Les opérands en virgule flottante ne sont pas autorisés avec les opérateurs de décalage, les opérateurs booléens bit à bit, les opérateurs d'indirection (*) ou certains autres opérateurs. Le compilateur a détecté un opérande en virgule flottante avec un opérateur non autorisé.

Illegal use of member pointer*Erreur du compilateur*

Un pointeur vers un membre de classe peut uniquement être utilisé avec les opérateurs d'affectation, de comparaison et les opérateurs `*`, `->`, `?:`, `&&` et `||` ou passés comme argument à une fonction. Le compilateur a détecté un pointeur de membre utilisé avec un opérateur non autorisé.

Illegal use of pointer*Erreur du compilateur*

Un pointeur peut être utilisé avec les opérateurs d'addition, de soustraction, d'affectation, de comparaison, d'indirection (`*`) ou l'opérateur flèche (`->`). Un pointeur a été utilisé avec un opérateur non autorisé dans le fichier source.

Ill-formed pragma*Mise en garde du compilateur*

Erreur dans la directive de compilation pragma spécifiée. Le compilateur Borland C++ ne reconnaît pas cette directive.

Image base address must be a multiple of 0x10000*Erreur de l'éditeur de liens*

L'adresse de base d'une image de base doit être un multiple de 0x10000.

Implicit conversion of type1 to type2 not allowed*Erreur du compilateur*

Si une fonction membre d'une classe est appelée à l'aide d'un pointeur vers une classe dérivée, la valeur de ce pointeur doit être convertie implicitement pour pointer vers la classe de base appropriée. Ce message indique qu'une telle conversion implicite n'est pas autorisée.

Import record does not match previous definition*Mise en garde de l'éditeur de liens*

Ce message s'affiche généralement si un enregistrement `IMPDEF` apparaît dans une bibliothèque d'importation en même temps que l'import en question est importé à partir d'un fichier `.DEF`. Si le nom interne et le numéro ordinal des imports sont différents dans leur description, ce message s'affiche et la première définition est utilisée. L'affichage de ce message peut être contrôlé avec l'option `-w`.

Improper use of typedef *identificateur**Erreur du compilateur*

Un symbole `typedef` a été utilisé là où une variable aurait dû être spécifiée dans une expression. Vérifiez la déclaration du symbole et corrigez les erreurs d'orthographe éventuelles.

Include files nested too deep*Erreur du compilateur*

Lorsque le compilateur détecte plus de 1 000 niveaux d'imbrication de fichiers d'en-tête, il suppose que le fichier d'en-tête est récursif et arrête la compilation en générant ce message d'erreur fatale.

nom_fichier (num_ligne): incompatible attribute*Erreur de l'éditeur de liens*

L'éditeur de liens a détecté des attributs de segment incompatibles dans une instruction `CODE` ou `DATA`. Par exemple, `PRELOAD` et `LOADONCALL` ne peuvent pas être les attributs du même segment.

Incompatible type conversion*Erreur du compilateur*

Transtypage impossible. Vérifiez les types.

Incorrect command-line argument: *argument**Erreur de MAKE*

Spécification d'arguments incorrects dans la ligne de commande.

Incorrect command-line option: *option**Erreur du compilateur*

Le paramètre indiqué dans la ligne de commande n'a pas été reconnu comme paramètre autorisé.

Incorrect configuration file option: *option**Erreur du compilateur*

Le paramètre de fichier de configuration n'a pas été reconnu comme paramètre autorisé. Vérifiez qu'il est précédé d'un trait d'union (`SYMBOL -`).

Incorrect number format*Erreur du compilateur*

Le compilateur a détecté un séparateur décimal dans un nombre hexadécimal.

Incorrect use of default*Erreur du compilateur*

Signe deux-points (:) manquant après le mot clé **default** dans une instruction **case**.

Incorrect version of RLINK32.DLL

Une version incorrecte du fichier RLINK32.DLL a été utilisée. Vérifiez que vous disposez de la version correcte de cette bibliothèque DLL.

Initializing enumeration with type*Mise en garde du compilateur*

Tentative d'initialisation d'une variable **enum** avec un autre type. Par exemple :

```
enum count { zero, one, two } x = 2;
```

Cette initialisation génère ce message car 2 est de type **int** et non de type **enum** count. Pour un meilleur style de programmation, il est recommandé d'utiliser un identificateur **enum** au lieu d'un entier littéral lorsque vous affectez ou initialisez des types **enum**.

Bien qu'il s'agisse d'une erreur, ce message a valeur de mise en garde pour que le programme puisse quand même s'exécuter.

Inline assembly not allowed*Erreur du compilateur*

Vous essayez de compiler à partir de l'environnement intégré un fichier source contenant des instructions en assembleur en ligne. Vous devez utiliser BCC pour compiler ce type de fichier.

Inline assembly not allowed in inline and template functions*Erreur du compilateur*

Le compilateur ne peut pas traiter des instructions en assembleur en ligne dans une fonction inline ou template en C++. Vous pouvez supprimer le code en assembleur inline ou, en cas de fonction inline, la convertir en macro-instruction, ou encore supprimer la classe de stockage **inline**.

int and string types compared*Erreur de MAKE*

Vous avez essayé de comparer un opérande de type entier et un opérande de type chaîne dans une expression !if ou !elif.

Internal error - chaîne*Erreur de l'éditeur de liens*

Cette erreur se produit très rarement. Notez le texte du message et contactez votre support technique Borland.

Internal Erreur de l'éditeur de liens code_erreur*Erreur de l'éditeur de liens*

Erreur de logique interne de l'éditeur de liens TLINK. Cette erreur ne devrait pas se produire en pratique mais elle est répertoriée ici au cas où une erreur plus spécifique serait générée. Si le problème persiste, notez le numéro *code_erreur* et contactez votre support technique Borland.

Invalid combination of opcode and operands*Erreur du compilateur*

L'assembleur intégré n'autorise pas cette combinaison d'opérandes. Les causes possibles sont les suivantes :

- Trop ou pas assez d'opérandes pour ce code opération en assembleur.
- Le nombre d'opérandes est correct mais leur type ou leur ordre ne correspond pas au code opération. Par exemple, **DEC 1, MOV AX** ou **MOV 1,AX**. Essayez de faire précéder les opérandes d'instructions de remplacement de type (par exemple, **MOV AX,WORD PTR toto**).

Invalid entry point offset*Erreur de l'éditeur de liens*

Ce message s'affiche uniquement lorsque des modules comportant des enregistrements 32 bits font l'objet d'une édition de liens. Il indique que le décalage initial du point d'entrée du programme dépasse la limite des 64 ko imposée par DOS.

Invalid indirection*Erreur du compilateur*

L'opérateur d'indirection (*) requiert un opérande de type pointeur non **void**.

Invalid initial stack offset*Erreur de l'éditeur de liens*

Ce message s'affiche uniquement lorsque des modules comportant des enregistrements 32 bits font l'objet d'une édition de liens. Il indique que la valeur initiale du pointeur de pile dépasse la limite des 64 ko imposée par DOS.

Invalid macro argument separator*Erreur du compilateur*

Dans une définition de macro-instruction, les arguments doivent être séparés par une virgule. Le compilateur a détecté un séparateur différent après un nom d'argument.

Invalid page size value ignored*Mise en garde du bibliothécaire*

Taille de page incorrecte. La taille de page doit correspondre à une puissance de 2 et être comprise entre 16 et 32 768.

Invalid pointer addition*Erreur du compilateur*

Impossible d'additionner deux pointeurs.

Invalid register combination (e.g. [BP+BX])*Erreur du compilateur*

L'assembleur intégré a détecté une combinaison illégale de registres dans une instruction. Seules sont autorisées les combinaisons de registres d'indices suivantes : [BX],[BP], [SI], [DI], [BX+SI], [BX+DI], [BP+SI] et [BP+DI]. les combinaisons [AX], [BP+BX] et [SI+DX] sont illicites.

Les variables locales (variables déclarées dans des procédures et des fonctions) sont généralement allouées sur la pile et accessibles à partir d'un registre BP. L'assembleur ajoute automatiquement [BP] à ce type de variables dans des références. Par conséquent, même si une construction comme **Local**[EBX] (où **Local** est une variable locale) semble correcte, elle ne l'est en fait pas car l'opérande final équivaldrait à **Local**[BP+EBX].

Invalid segment definition in module *module**Erreur de l'éditeur de liens*

Le compilateur a généré un fichier objet défectueux. Si cela se produit dans un fichier créé avec Borland C++, recompilez le fichier. Si le problème persiste, contactez votre support technique Borland.

Invalid template argument list*Erreur du compilateur*

Dans une déclaration de template, le mot clé **template** doit être suivi d'une liste d'arguments formels spécifiée entre les délimiteurs < et >. La liste d'arguments spécifiée n'est pas autorisée.

Invalid template qualified name *template::nom**Erreur du compilateur*

Dans la définition d'un membre de classe template, les arguments du nom de classe template utilisé comme opérande gauche de l'opérateur :: doivent correspondre aux arguments formels de la classe template. Par exemple :

```
template <class T> class X
{
    void f();
};

template <class T> void X<T>::f() {}
```

La déclaration suivante serait illicite :

```
template <class T> void X<int>::f() {}
```

Invalid use of dot*Erreur du compilateur*

Un opérateur point (.) doit être immédiatement suivi d'un identificateur.

Invalid use of template *template**Erreur du compilateur*

En dehors de la définition d'un template, il est impossible d'utiliser un nom de classe template sans spécifier ses arguments. Par exemple, vous pouvez utiliser **vector<int>** mais pas **vector**.

Irreducible expression tree*Erreur du compilateur*

Indique une forme d'Erreur du compilateur. Une expression sur la ligne indiquée dans le fichier source a empêché la génération de code par le générateur de code. Corrigez l'expression. Si l'erreur se produit à nouveau, contactez votre support technique Borland.

base* is an indirect virtual base class of *classe*Erreur du compilateur*

Impossible de créer un pointeur vers un membre C++ d'une classe de base virtuelle donnée. La tentative de création de pointeur a été effectuée (soit directement soit par transtypage).

identificateur* is assigned a value that is never usedMise en garde du compilateur*

La variable apparaît dans une affectation mais n'est utilisée nulle part ailleurs dans la fonction dont le traitement vient de se terminer. Le message s'affiche uniquement lorsque le compilateur détecte l'accolade fermante.

identificateur* is declared as both external and staticMise en garde du compilateur*

L'identificateur indiqué apparaît à la fois dans une déclaration qui l'a implicitement ou explicitement marqué comme global ou externe et dans une déclaration statique. L'identificateur est finalement interprété comme statique. Revoyez toutes les déclarations de cet identificateur.

identificateur* is declared but never usedMise en garde du compilateur*

La variable spécifiée a été déclarée dans le bloc dont le traitement vient de se terminer mais n'a jamais été utilisée. Le message s'affiche lorsque le compilateur détecte l'accolade fermante d'une instruction ou d'une fonction composée. La déclaration de la variable est effectuée au début de l'instruction ou de la fonction composée.

symbole* is duplicated in module *module*Erreur de l'éditeur de liens*

Conflit entre deux symboles (déclarés comme "public" ou "communal") définis dans le même module. L'erreur se produit si les deux symboles sont détectés dans un fichier .OBJ. Un message de mise en garde s'affiche si TLINK détecte les deux symboles dans une bibliothèque. Dans ce cas, TLINK utilise la première définition.

constructeur* is not a base class of *classe*Erreur du compilateur*

Un constructeur de classe C++ **classe** tente d'appeler un constructeur de classe de base **constructeur** ou vous essayez de modifier les droits d'accès de `classe::constructeur`. **constructeur** n'est pas une classe de base de **classe**. Vérifiez les déclarations correspondantes.

identificateur* is not a member of *structure*Erreur du compilateur*

Vous essayez de faire référence à *identificateur* en tant que membre de **structure** alors qu'il ne s'agit pas d'un membre. Vérifiez les déclarations correspondantes.

identificateur* is not a non-static data member and can't be initialized hereErreur du compilateur*

Seules les données membres peuvent être initialisées dans un constructeur. Le message indique que la liste comporte un membre statique ou un membre de fonction.

identificateur* is not a parameterErreur du compilateur*

Dans la section de déclaration des paramètres d'une fonction "ancien style", l'*identificateur* indiqué est déclaré mais ne fait pas partie de la liste des paramètres. Supprimez la déclaration ou ajoutez *identificateur* en tant que paramètre.

type is not a polymorphic class type*Erreur du compilateur*

Un `dynamic_cast` a été utilisé avec un pointeur vers une classe qui a été compilée avec l'option de compilateur `-RT` désactivée.

identificateur is not a public base class of type_classe*Erreur du compilateur*

L'opérande droit d'un opérateur `.*`, `->*` ou `::` n'est pas un pointeur vers un membre d'une classe. Il doit correspondre à une classe de base accessible du type de la classe de l'opérande gauche.

nom_fichier is not a valid library*Mise en garde de l'éditeur de liens*

Cette erreur se produit si un fichier qui n'était pas une module de bibliothèque correct a été passé à l'éditeur de liens dans la section de bibliothèque.

membre is not accessible*Erreur du compilateur*

Vous essayez de faire référence au membre de classe C++ *membre* mais celui-ci est **privé** ou **protégé** et ne peut être référencé à partir de cette fonction. Ce cas peut se produire lorsque vous tentez d'appeler une fonction membre (ou un constructeur) surchargée et accessible mais que les arguments correspondent à une fonction inaccessible. La vérification de la résolution de surcharge est systématiquement effectuée avant la vérification d'accessibilité. S'il s'agit de ce problème, essayez un transtypage explicite d'un ou de plusieurs paramètres pour sélectionner la fonction accessible requise.

Last parameter of operateur must have type int*Erreur du compilateur*

Lorsqu'un **opérateur** de post-incrémentation `++` ou de post-décrémentation `--` est déclaré, le dernier paramètre doit être déclaré avec le type `int`.

Library contains COMDEF records - extended dictionary not created*Mise en garde du bibliothécaire*

Un enregistrement objet ajouté à une bibliothèque contient un enregistrement COMDEF. Ceci n'est pas compatible avec l'option de création de dictionnaire étendu.

Library too large, restart with library page size taille*Erreur du bibliothécaire*

La bibliothèque créée n'a pas pu être construite en tenant compte de la taille de page de bibliothèque en cours.

Linkage specification not allowed*Erreur du compilateur*

Les spécifications d'édition de liens comme **extern "C"** sont autorisées uniquement au niveau du fichier. Déplacez cette déclaration de fonction en dehors du niveau de fichier.

Linker name conflict for fonction*Erreur du compilateur*

Ce message est généré par le compilateur lorsque le nom d'une fonction inline ou d'une table virtuelle C++ tronquée pour dépassement de longueur correspond au nom de la fonction ou de la table virtuelle qui a été générée précédemment (ce problème se pose souvent avec des templates). Modifiez le nom de la classe ou de la fonction ou utilisez l'option de compilation `-Vs`.

Linker stack overflow*Erreur de l'éditeur de liens*

L'éditeur de liens utilise une procédure récursive pour marquer les modules à inclure dans une image exécutable à partir de bibliothèques. Cette procédure peut entraîner des débordements de pile dans des circonstances extrêmes. Si ce message s'affiche, supprimez certains modules des bibliothèques et incluez-les avec les fichiers objet dans l'édition de liens. Effectuez une nouvelle tentative.

Lvalue required*Erreur du compilateur*

La partie gauche d'un opérateur d'affectation doit correspondre à une expression adressable comme une variable numérique ou une variable pointeur, une référence de champ de structure, une indirection via pointeur ou un élément de tableau indicé.

- Macro argument syntax error** *Erreur du compilateur*
La définition d'une macro doit comporter un identificateur comme paramètre. Un caractère ne correspondant pas à un spécificateur a été détecté à la place de l'argument.
- Macro expansion too long** *Erreur du compilateur*
Le développement d'une macro-instruction ne doit pas dépasser 4096 caractères.
- Macro expansion too long** *Erreur de MAKE*
Le développement d'une macro-instruction ne doit pas dépasser 4096 caractères. Cette erreur se produit souvent si une macro-instruction se développe de façon récursive (opération illicite).
- Macro substitute text *chaîne* is too long** *Erreur de MAKE*
- Macro replace text *chaîne* is too long** *Erreur de MAKE*
Le texte *chaîne* de substitution ou de remplacement de macro a fait déborder le tampon interne de MAKE (limité à 512 octets).
- main must have a return type of int** *Erreur du compilateur*
En C++, la fonction **main** doit obligatoirement être déclarée avec un type de valeur de renvoi **int**.
- Malformed command-line** *Erreur du compilateur*
Entrée incorrecte dans la ligne de commande.
- Matching base class function for *fonction* has different dispatch number.** *Erreur du compilateur*
Si une fonction DDVT est déclarée dans une classe dérivée, la fonction de classe de base correspondante doit avoir le même indice de distribution que la fonction dérivée.
- Matching base class function for *fonction* is not dynamic** *Erreur du compilateur*
Si une fonction DDVT est déclarée dans une classe dérivée, la fonction de classe de base correspondante doit également être dynamique.
- Maximum precision used for member pointer type *type*** *Mise en garde du compilateur*
Lorsqu'un type de pointeur de membre est déclaré, sa classe n'a pas été complètement définie, et l'option **-Vmd** a été utilisée. Le compilateur doit alors utiliser la représentation la plus générale (et donc la moins efficace) de ce type de pointeur de membre. Le code généré peut s'avérer moins performant (le type de pointeur de membre étant trop grand) et certains problèmes peuvent se produire lors d'une compilation séparée.
- Member function must be called or its address taken** *Erreur du compilateur*
Lorsqu'une fonction membre est utilisée dans une expression, elle doit être appelée ou son adresse récupérée avec l'opérateur **&**. La fonction membre en question est utilisée dans un contexte non autorisé.
- Member identifier expected** *Erreur du compilateur*
Nom de structure ou de membre de classe C++ manquant. La partie droite d'un opérateur point (.) ou flèche (->) doit correspondre à un nom de membre dans la structure ou la classe spécifiée à gauche de l'opérateur.
- Member is ambiguous: *membre1* et *membre2*** *Erreur du compilateur*
Vous devez qualifier la référence de membre avec le nom de classe de base appropriée. Dans une classe *classe* C++, un member *membre* peut figurer dans plusieurs classes de base. Le message signale qu'il n'a pas été qualifié pour indiquer à quelle classe de base il appartenait réellement. Ce problème survient uniquement en cas d'héritage multiple, lorsque le nom de membre dans chaque classe de base n'est pas occulté par le même nom de membre dans une classe dérivée sur le même chemin d'accès. En C++, le test de détection d'ambiguïté doit être effectué avant la vérification des droits d'accès (**private**, **protected**, **public**). Il est donc possible que ce message s'affiche même si un seul des membres (ou aucun d'entre eux) est accessible.

Member *membre* cannot be used without an object*Erreur du compilateur*

Vous avez spécifié `classe::membre` où *membre* est de type ordinaire (non statique) mais aucune classe ne peut être associée à ce membre. Par exemple, il est possible d'écrire `obj.classe::membre` mais `classe::membre` n'est pas autorisé.

Member *membre* has the same name as its class*Erreur du compilateur*

Le nom d'une donnée membre statique, d'un énumérateur, d'un membre d'union anonyme ou d'un type imbriqué doit être différent de celui de sa classe respective. Seul une fonction membre ou un membre non statique peut avoir le même nom que sa classe.

Member *membre* is initialized more than once*Erreur du compilateur*

Dans un constructeur de classe C++, la liste des initialisations après l'en-tête du constructeur comporte plusieurs occurrences d'un nom de membre.

Member pointer required on right side of `.*` or `->*`*Erreur du compilateur*

La partie droite d'un opérateur point-étoile (`.*`) ou flèche-étoile (`->*`) doit être déclarée comme pointeur vers un membre de la classe spécifiée à gauche de l'opérateur. Ce message indique que la partie droite ne correspond pas à un pointeur de membre.

Memory full listing truncated!*Mise en garde du bibliothécaire*

Mémoire insuffisante pour le bibliothécaire qui ne peut créer qu'un fichier de listage de bibliothèque incomplet.

Memory reference expected*Erreur du compilateur*

Référence de mémoire requise par l'assembleur intégré. Vérifiez que vous avez spécifié les opérandes de registres d'indice entre crochets (par exemple, `MOV AX,[BX+SI]` et non `MOV AX,BX+SI`).

Misplaced break*Erreur du compilateur*

Le compilateur a détecté une instruction **break** en dehors d'une construction **switch** ou d'une boucle de répétition.

Misplaced continue*Erreur du compilateur*

Le compilateur a détecté une instruction **continue** en dehors d'une boucle de répétition.

Misplaced decimal point*Erreur du compilateur*

Le compilateur a détecté un séparateur décimal dans la partie exposant d'une constante en virgule flottante.

Misplaced elif directive*Erreur du compilateur*

Une directive `#elif` a été spécifiée sans directive `#if`, `#ifdef` ou `#ifndef` correspondante.

Misplaced elif statement*Erreur de MAKE*

Une directive `!elif` a été spécifiée sans directive `!if` correspondante.

Misplaced else*Erreur du compilateur*

Une instruction **else** a été spécifiée sans instruction **if** correspondante. Ce message peut indiquer la présence d'une instruction **else** ou d'un point-virgule ajouté par erreur, une erreur d'accolade ou une erreur de syntaxe dans une instruction **if** précédente.

Misplaced else directive*Erreur du compilateur*

Une directive `#else` a été spécifiée sans directive `#if`, `#ifdef` ou `#ifndef` correspondante.

Misplaced else statement*Erreur de MAKE*

Une directive `!else` a été spécifiée sans directive `!if` correspondante.

Misplaced endif directive*Erreur du compilateur*

Une directive `#endif` a été spécifiée sans directive `#if`, `#ifdef` ou `#ifndef` correspondante.

Misplaced endif statement*Erreur de MAKE*

Une directive `!endif` a été spécifiée sans directive `!if` correspondante.

nom_fichier (num_ligne): missing internal name*Erreur de l'éditeur de liens*

Dans la section `IMPORTS` du fichier de définition du module, une référence à une entrée a été spécifiée à l'aide d'un nom de module et d'un nombre ordinal. Lorsqu'une entrée est spécifiée par un numéro ordinal, un nom interne doit être affecté à la définition de cet import. Ce nom interne est utilisé par le programme pour faire référence à la définition importée. La définition du module doit avoir la syntaxe suivante :

```
<nom_interne>=<nom_module>.<ordinal>
```

Mixed common types in module module. Cannot mix COMDEFs and VIRDEFs.*Erreur de l'éditeur de liens*

Impossible de mélanger `COMDEF` et `VIRDEF`. Désactivez l'option `-Fc` pour arrêter la génération de `COMDEF` ou activez l'option `-Vs` pour arrêter la génération de `VIRDEF`.

Mixing pointers to different 'char' types*Mise en garde du compilateur*

Un pointeur de type **signed char** a été converti en pointeur **unsigned char** (ou inversement) sans instruction de `transtypage` explicite. Au sens strict, cette opération est incorrecte mais n'a pas de conséquences graves.

Multiple base classes require explicit class names*Erreur du compilateur*

Dans un constructeur de classe C++, chaque appel de constructeur de classe spécifié dans l'en-tête de constructeur doit comporter le nom de la classe de base s'il existe plusieurs classes de base immédiates.

Multiple declaration for identificateur*Erreur du compilateur*

Déclaration multiple d'un même identificateur. Cette erreur peut être due à un conflit de déclarations (par exemple, `int a; double a;`), à une fonction déclarée de deux façons différentes, une étiquette répétée dans une même fonction ou une déclaration répétée autre qu'une fonction **externe** ou une variable simple (en C).

Multiple entry points defined*Erreur de l'éditeur de liens*

Un seul point d'entrée doit être défini pour l'application.

Multiple stack segments found. The most recent one will be used.*Mise en garde de l'éditeur de liens*

Ce message s'affiche lorsque deux segments de pile de noms différents sont définis dans les modules objets. Le code de démarrage définit un segment de pile pour l'application. L'affichage de ce message peut être contrôlé à l'aide de l'option **-w**.

identificateur must be a member function*Erreur du compilateur*

La plupart des fonctions d'opérateurs C++ peuvent être des membres de classe ou des fonctions non membres ordinaires. Certaines doivent impérativement être des membres de classe comme les opérateurs `=`, `->`, `()` et les conversions de types. La fonction opérateur en question doit être une fonction membre.

identificateur must be a member function or have a parameter of class type*Erreur du compilateur*

La plupart des fonctions d'opérateur C++ doivent avoir un paramètre de type classe implicite ou explicite. Cette fonction d'opérateur a été déclarée en dehors d'une classe et n'a pas de paramètre explicite de type classe.

identificateur must be a previously defined class or struct*Erreur du compilateur*

Vous essayez de déclarer *identificateur* comme une classe de base. *identificateur* n'est pas une classe ou n'a pas été complètement défini. Corrigez le nom ou modifiez les déclarations.

- identificateur must be a previously defined enumeration tag** *Erreur du compilateur*
 Cette déclaration essaie de faire référence à *identificateur* en tant qu'indicateur de type **enum**. Corrigez le nom ou modifiez les déclarations.
- identificateur must be declared with no parameters** *Erreur du compilateur*
 Déclaration incorrecte de cette fonction d'opérateur C++ qui n'accepte aucun paramètre.
- identificateur must be declared with one parameter** *Erreur du compilateur*
 Déclaration incorrecte de cette fonction d'opérateur C++ qui accepte plusieurs paramètres.
- opérateur must be declared with one or no parameters** *Erreur du compilateur*
 Lorsque l'opérateur ++ ou -- est déclaré en tant que fonction membre, il doit être déclaré comme n'acceptant aucun paramètre (pour la version préfixée de l'opérateur) ou comme acceptant un paramètre de type **int** (pour la version suffixée de l'opérateur).
- opérateur must be declared with one or two parameters** *Erreur du compilateur*
 Lorsque l'opérateur ++ ou -- est déclaré comme une fonction non membre, il doit être déclaré comme acceptant un paramètre (pour la version préfixée de l'opérateur) ou deux paramètres (pour la version suffixée de l'opérateur).
- identificateur must be declared with two parameters** *Erreur du compilateur*
 Déclaration incorrecte de cette fonction d'opérateur C++ qui accepte uniquement deux paramètres.
- Must take address of a memory location** *Erreur du compilateur*
 Un opérateur d'adressage (&) a été utilisé avec une expression illicite (une variable registre en C, par exemple).
- Need an identifier to declare** *Erreur du compilateur*
 La déclaration doit être complétée par un identificateur dans ce contexte. Il peut s'agir d'un **typedef** sans nom ou d'un point-virgule supplémentaire au niveau du fichier. En C++, cette erreur peut être due à un nom de classe utilisé incorrectement pour un autre type d'identificateur.
- No : following the ?** *Erreur du compilateur*
 Signe deux-points (:) manquant après le point d'interrogation (?) dans l'expression ou parenthèses manquantes ou incorrectement imbriquées.
- No base class to initialize** *Erreur du compilateur*
 Ce constructeur de classe C++ essaie d'appeler implicitement un constructeur de classe de base mais la classe en question a été déclarée sans classes de base. Vérifiez les déclarations.
- No closing quote** *Erreur de MAKE*
 Apostrophe fermante manquante dans une expression chaîne d'une instruction !if ou !elif.
- No declaration for function *fonction*** *Mise en garde du compilateur*
 Une fonction a été appelée avant d'avoir été déclarée. En C, une fonction peut être déclarée sans prototype (par exemple, `int func();`). En C++, la déclaration d'une fonction correspond également à son prototype. Ainsi, l'exemple précédent équivaut à `int func(void);`. Cette déclaration peut être de style classique (sans prototype) ou moderne (avec prototype).
- No .DEF file; using defaults** *Mise en garde de l'éditeur de liens*
 Aucun fichier .DEF n'a été spécifié pour l'édition de liens. L'option **-w** permet de contrôler l'affichage de ce message.
- No file name ending** *Erreur du compilateur*
 Apostrophe ou parenthèse angulaire fermante manquante après le nom de fichier spécifié dans une directive `#include`.

- No filename ending** *Erreur de MAKE*
Apostrophe ou parenthèse angulaire fermante manquante après le nom de fichier spécifié dans une instruction #include.
- No file names given** *Erreur du compilateur*
Aucun nom de fichier spécifié dans la ligne de commande du compilateur Borland C++. Spécifiez un fichier source.
- No internal name for IMPORT in .DEF file**
Erreur sémantique dans le fichier .DEF : nom interne d'import manquant avant le nom de module. Par exemple :
- ```
IMPORTS
 _toto.1
```
- Dans cet exemple, `_toto` est supposé désigner la fonction à importer mais la syntaxe correcte est la suivante :
- ```
IMPORTS
    _toto=mydll.1
```
- No macro before =** *Erreur de MAKE*
Vous devez attribuer un nom à une macro-instruction avant de pouvoir lui affecter à une valeur.
- No match found for wildcard expression** *Erreur de MAKE*
Aucun fichier ne correspond à la spécification utilisant des caractères de remplacement et ne peut donc être traité par MAKE.
Par exemple, si vous codez
- ```
prog.exe: *.obj
```
- MAKE envoie ce message d'erreur si aucun fichier avec l'extension .OBJ ne figure dans le répertoire en cours.
- No output file specified** *Erreur de l'éditeur de liens*  
Aucun fichier .EXE ou .DLL n'a été spécifié. Dans la mesure où l'éditeur de liens utilise par défaut le premier nom avec l'extension .OBJ, cette erreur a généralement pour origine l'absence de fichiers objet.
- No program entry point** *Mise en garde de l'éditeur de liens*  
Ce message s'affiche si aucun point d'entrée d'exécution n'a été défini pour l'application. Cette erreur se produit si vous oubliez d'effectuer l'édition de liens du code de démarrage. L'affichage de ce message peut être contrôlé à l'aide de l'option **-w**.
- No program starting address defined** *Mise en garde de l'éditeur de liens*  
Ce message indique qu'aucun module n'a été défini dans l'adresse de départ initiale du programme. Ceci est probablement dû à un défaut d'édition de liens du module d'initialisation C0x.OBJ.
- No stack** *Mise en garde de l'éditeur de liens*  
Ce message s'affiche si aucun segment de pile n'a été défini dans l'un des fichiers objet ou l'une des bibliothèques incluses dans l'édition de liens. Ce message est normal pour le modèle de mémoire "tiny" en Borland C++ ou pour tout programme d'application qui doit être converti en fichier .COM. Ce message indique une erreur sauf pour les fichiers de bibliothèque DLL.  
Si un programme Borland C++ génère ce message pour un modèle de mémoire autre que le modèle "tiny", vérifiez que vous utilisez les fichiers objet de démarrage C0x appropriés.
- No stub for fixup at segment:xxxxh in module module** *Mise en garde de l'éditeur de liens*  
Cette erreur se produit lorsque la cible d'une référence se trouve dans un segment de recouvrement mais qu'aucun "stub" n'est disponible pour une cible externe. Ceci est dû au fait de restreindre l'accès à un symbole dans un recouvrement qui est référencé à partir du même module.

**No terminator specified for in-line file operator***Erreur de MAKE*

Le fichier MAKE contient les opérateurs **&&** ou **<<** pour débiter un fichier en ligne, mais la fin de ce fichier n'est pas marquée.

**Non-const function fonction called for const object***Mise en garde du compilateur*

Une fonction membre non **constante** a été appelée pour un objet de type **const**. Bien qu'il s'agisse d'une erreur, ce message a valeur de mise en garde pour que le programme puisse quand même s'exécuter.

**Non portable pointer comparison***Mise en garde du compilateur*

Le fichier source comporte une comparasion entre un pointeur vers un non-pointeur autre que la constante nulle. Utilisez un transtypage pour supprimer ce message si la comparaison est correcte.

**Nonportable pointer conversion***Erreur du compilateur*

Une conversion implicite entre un pointeur et un type entier est requise mais les types n'ont pas la même taille. Cette opération ne peut pas être effectuée sans transtypage explicite. Vérifiez la logique et le résultat de cette conversion.

**Non portable pointer conversion***Mise en garde du compilateur*

Une valeur entière non nulle est utilisée dans un contexte où un pointeur ou une valeur entière est requis. Le type entier et le pointeur ont la même taille. Utilisez un transtypage explicite pour vous assurer de la logique et du résultat de l'opération.

**Non resident Name Table is greater than 64K***Mise en garde de l'éditeur de liens*

La taille maximale d'une table de noms NonResident est de 64 ko (selon les spécifications en vigueur). L'éditeur de liens poursuit son opération mais ne tient pas compte des autres noms NonResident détectés lors de l'édition de liens.

**Nontype template argument must be of scalar type***Erreur du compilateur*

Un argument de template formel non type doit être de type scalaire (type entier, énumération ou pointeur).

**Non-ANSI Keyword Used: mot clé***Erreur du compilateur*

Un mot clé non défini par la norme ANSI (comme **\_\_fastcall**) a été utilisé alors que l'option **-A** stipulait l'utilisation d'un code en stricte conformité avec cette norme.

**Non-virtual function fonction declared pure***Erreur du compilateur*

Seule une fonction virtuelle peut être déclarée pure dans la mesure où une classe dérivée doit pouvoir la remplacer.

**Non-volatile function fonction called for volatile object***Mise en garde du compilateur*

En C++, une fonction membre de classe a été appelée pour un objet volatile de type classe, mais la fonction n'a pas été déclarée comme **volatile** après l'en-tête de fonction. Seule une fonction volatile peut être appelée pour un objet volatile.

**nom\_fichier not a MAKE***Erreur de MAKE*

Le fichier spécifié avec l'option **-f** n'est pas un fichier MAKE.

**Not an allowed type***Erreur du compilateur*

Le fichier source contient une déclaration d'un type non autorisé. Par exemple, une fonction renvoyant une fonction ou un tableau.

**Not enough memory***Erreur de MAKE*

La totalité de la mémoire de travail a été épuisée.

**Not enough memory***Erreur de l'éditeur de liens*

Mémoire insuffisante pour exécuter TLINK. Essayez de réduire la taille d'un disque RAM ou d'un cache actif. Si vous êtes en mode d'exécution réel, essayez d'utiliser l'option MAKE **-S** ou de retirer des programmes en mémoire résidentes ou des gestionnaires de réseau.

**Not enough memory for command-line buffer***Erreur du bibliothécaire*

Mémoire insuffisante pour le bibliothécaire.

**module not found in library***Mise en garde du bibliothécaire*

Tentative a été faite d'exécuter une opération ' \_ ' ou '\*' sur une bibliothèque mais l'objet spécifié ne figure pas dans la bibliothèque.

**Null pointer assignment***Erreur d'exécution*

A la fin d'un programme (modèle de mémoire "small" ou "medium"), une vérification est effectuée pour déterminer si les premiers octets du segment de données du programme ont été modifiés. Ces octets ne doivent en principe jamais être modifiés par un programme en exécution. S'ils ont été modifiés, le message CHAR 34 Null pointer assignment CHAR 34 s'affiche pour indiquer qu'une valeur a été probablement stockée dans un pointeur non initialisé. Même si le programme peut sembler fonctionner normalement, il s'agit d'un bug important qui doit être corrigé rapidement. Si vous ne rectifiez pas ce pointeur non initialisé, le programme peut réagir de façon imprévisible et "bloquer" l'ordinateur dans les modèles de mémoire "large", "compact" et "huge". Vous pouvez utiliser le débogueur intégré pour rechercher des pointeurs NULL.

**Numeric constant too large***Erreur du compilateur*

Impossible de générer une séquence d'échappement de type chaîne ou caractère plus grande que \xFF (en hexadécimal) ou \377 (en octal). Un caractère sur deux octets peut être spécifiée à l'aide d'une deuxième barre oblique inversée.

Par exemple, \x0D\x0A représente une constante sur deux octets. Un littéral numérique après une séquence d'échappement doit être décomposé comme suit :

```
printf("\x0D" "12345");
```

Cette instruction affiche un retour chariot suivi de 12345.

**Object module *nom\_fichier* is invalid***Erreur du bibliothécaire*

Le bibliothécaire n'a pas pu interprété l'enregistrement d'en-tête du module objet ajouté à la bibliothèque et a supposé que le module était incorrect.

**Objects of type *type* cannot be initialized with {}***Erreur du compilateur*

Une structure C ordinaire peut être initialisée à l'aide d'un groupe de valeurs entre accolades. Une classe C++ peut être initialisée à l'aide de constructeurs si la classe définit des constructeurs, des membres privés, des fonctions ou des classes de base virtuelles.

**Old debug information in module *module* will be ignored***Mise en garde de l'éditeur de liens*

Incompatibilité des informations de débogage avec cette version de l'éditeur de liens. Ces informations ne sont pas prises en compte.

**Only <<KEEP or <<NOKEEP***Erreur de MAKE*

Vous avez spécifié quelque chose d'autre après KEEP ou NO KEEP lors de la fermeture d'un fichier inline temporaire.

**Only member functions may be 'const' or 'volatile'***Erreur du compilateur*

Seule une fonction membre de classe peut être déclarée **const** et/ou **volatile**. Toute déclaration d'un autre élément avec ces types est illicite.

**Only one of a set of overloaded functions can be "C"***Erreur du compilateur*

Les fonctions C++ sont surchargées par défaut et le compilateur affecte un nouveau nom à chaque fonction. Pour remplacer ce nom en déclarant la fonction externe "C", vous ne pouvez effectuer cette opération que pour un seul groupe de fonctions portant le même nom. Autrement, l'éditeur de liens détecterait plusieurs fonctions globales avec le même nom.

- Operand of delete must be non-const pointer** *Erreur du compilateur*  
Impossible de supprimer une valeur de pointeur constant avec l'opérateur **delete**.
- Operator [ ] missing** *Erreur du compilateur*  
L'**opérateur [ ]** a été déclaré comme **opérateur [**. Rajoutez le signe **]** manquant ou modifiez la déclaration.
- Operator -> must return a pointer or a class** *Erreur du compilateur*  
L'**opérateur ->** doit être déclaré pour renvoyer une classe ou un pointeur vers une **classe** (ou une **structure** ou une **union**) à laquelle il est possible d'appliquer l'opérateur **->**.
- Operator delete must return void** *Erreur du compilateur*  
Déclaration incorrecte de l'opérateur surchargé **delete**. Redéclarez-le pour qu'il renvoie une valeur void.
- Operator delete[ ] Must Return Void** *Erreur du compilateur*  
Déclaration incorrecte de l'opérateur surchargé **delete**. Redéclarez-le comme suit :
- avec un paramètre unique void\* ;
  - avec un second paramètre de type size\_t.
- Si vous utilisez la deuxième syntaxe, elle sera utilisée de préférence à la première. L'opérateur global **delete** ne peut être déclaré qu'avec le paramètre **void\***.
- Operator must be declared as function** *Erreur du compilateur*  
Déclaration incorrecte d'un opérateur surchargé. Déclarez-le avec le type fonction.
- Operator new must have an initial parameter of type size\_t** *Erreur du compilateur*  
L'opérateur **new** peut être déclaré avec un nombre de paramètres arbitraire. L'un de ces paramètres doit impérativement spécifier le volume d'espace à allouer.
- Operator new must return an object of type void \*** *Erreur du compilateur*  
Déclaration incorrecte de l'opérateur surchargé **new**. Il doit impérativement renvoyer un objet de type void\*.
- Operator new[] must have an initial parameter of type size\_t** *Erreur du compilateur*  
L'opérateur **new** peut être déclaré avec un nombre de paramètres arbitraire. L'un de ces paramètres doit impérativement spécifier le volume d'espace à allouer.
- Operator new[] must return an object of type void \*** *Erreur du compilateur*  
Déclaration incorrecte de l'opérateur surchargé **new**.
- Operators may not have default argument values** *Erreur du compilateur*  
Un opérateur surchargé ne peut avoir de valeurs d'argument par défaut.
- Out of memory** *Erreur du compilateur*  
La totalité de la mémoire de travail est épuisée. Compilez le fichier sur une machine disposant de plus de mémoire.
- Out of memory** *Erreur du bibliothécaire*  
Pour un certain nombre de raisons, le bibliothécaire et Borland C++ ne disposent plus de mémoire pour créer la bibliothèque. Dans des cas plus spécifiques, un message plus détaillé est affiché. Ce message s'affiche généralement pour indiquer une mémoire insuffisante.
- Vous devez libérer de la mémoire si ce message s'affiche parce que les tables de symboles publics sont devenues trop volumineuses. Vous devrez éventuellement retirer des programmes résidents ou des pilotes de périphériques utilisant la mémoire en mode réel ou refermer certaines fenêtres. Vous pouvez également libérer de la mémoire dans l'EDI en quittant certains éditeurs.

**Out of memory***Erreur de l'éditeur de liens*

L'éditeur de liens ne dispose plus de mémoire allouée dynamiquement nécessaire au processus d'édition de liens. Ce message indique une mémoire insuffisante pour l'utilisation de TLINK due à un nombre trop élevé de modules, d'externes, de groupes ou de segments qui ont été définis par les fichiers objets liés ensemble. Essayez de réduire la taille des disques RAM et/ou des caches actifs.

**Out of memory creating extended dictionary***Erreur du bibliothécaire*

Mémoire insuffisante pour que le bibliothécaire puisse créer un dictionnaire étendu pour une bibliothèque. La bibliothèque est créée mais ne dispose pas d'un tel dictionnaire.

**Out of memory for block *bloc****Erreur de l'éditeur de liens*

Erreur rare. Notez le texte du message (ainsi que le nom du bloc) et contactez votre support technique Borland.

**Out of memory reading LE/LIDATA record from object module***Erreur du bibliothécaire*

Le bibliothécaire essaie de lire un enregistrement de données à partir du module objet, mais il ne dispose pas d'un bloc de mémoire assez volumineux. Pour résoudre ce problème, vous pouvez essayer d'ajouter le module avant les autres modules si ce module a un ou plusieurs segments de données importants. Cette méthode permet de disposer de davantage de mémoire pour maintenir plus tard les listes de symboles publics et de modules.

**Out of space allocating per module debug struct***Erreur du bibliothécaire*

Mémoire insuffisante pour que le bibliothécaire alloue de l'espace aux informations de débogage associées à un module objet donné. Pour résoudre ce problème, supprimez les informations de débogage de certains des modules à ajouter à la bibliothèque.

**Output device is full***Erreur du bibliothécaire*

Ce message indique que le disque est pratiquement saturé.

**Overlays generated and no overlay manager included***Mise en garde de l'éditeur de liens*

Ce message s'affiche en cas de création de segments de recouvrement mais le symbole `__OVRTRAP__` n'est défini dans aucun des modules objet ou des bibliothèques inclus dans l'édition de liens. Ce symbole est défini dans la bibliothèque standard OVERLAY.LIB.

**Overlays only supported in medium, large, and huge memory models***Erreur du compilateur*

Seuls les programmes utilisant les modèles de mémoire "medium", "large" ou "huge" peuvent comporter des segments de recouvrement.

**Overload is now unnecessary and obsolete***Mise en garde du compilateur*

Dans les versions précédentes de C++, le mot réservé **overload** servait à marquer les noms de fonctions surchargées. La version actuelle de C++ utilise une "édition de liens avec types fiabilisés" qui suppose qu'une fonction est surchargée par défaut (sauf indication contraire). Ce message indique qu'il n'est plus nécessaire d'utiliser **overload**.

**Overloadable operator expected***Erreur du compilateur*

Pratiquement tous les opérateurs C++ peuvent être surchargés à l'exception des opérateurs point sélection de champ (`.`), point-étoile (`.*`), double deux-points (`::`) et l'expression conditionnelle (`?:`). Les opérateurs de préprocesseur (`#` et `##`) qui ne sont pas des opérateurs en C et C++ et d'autres signes de ponctuation, comme le point-virgule, ne peuvent pas être surchargés.

**Overloaded *nom\_fonction* ambiguous in this context***Erreur du compilateur*

Un nom de fonction surchargée peut être appelé sans qu'il soit nécessaire d'appeler la fonction uniquement si une variable ou un paramètre de type approprié est initialisé ou affecté. Ce message indique qu'un nom de fonction surchargée a été utilisé dans un autre contexte.

**Overloaded prefix 'operator opérateur' used as a postfix operator***Mise en garde du compilateur*

Selon les spécifications les plus récentes de C++, il est possible de surcharger les opérateurs ++ et - - en version préfixée ou suffixée. Pour pouvoir compiler un code conforme à des spécifications précédentes, dans le cas où seul l'opérateur (en version préfixée) est surchargé, mais utilisé dans un contexte de suffixage, Borland C++ utilise l'opérateur préfixé et génère ce message.

**P1001 Unable to read file *nom\_fichier****Erreur du compilateur d'aide (projet)*

Le fichier spécifié dans le fichier projet est illisible. Ce message indique une erreur de fichier DOS.

**P1003 Invalid path specified in Root option***Erreur du compilateur d'aide (projet)*

Le fichier spécifié avec l'option Root est introuvable. Le compilateur utilise le répertoire de travail en cours.

**P1005 Path and filename exceed limit of 79 characters***Erreur du compilateur d'aide (projet)*

Le nom de chemin absolu ou du chemin relatif à partir du répertoire racine comporte plus de 79 caractères (limite imposée par DOS). Le fichier n'est pas pris en compte.

**P1007 Root path exceeds maximum limit of 66 characters***Erreur du compilateur d'aide (projet)*

Le nom de chemin du répertoire racine comporte plus de 66 caractères (limite imposée par DOS). Le nom de chemin n'est pas pris en compte et le compilateur utilise le répertoire de travail en cours.

**P1009 [FILES] section missing***Erreur du compilateur d'aide (projet)*

Section [Files] obligatoire. La compilation est abandonnée.

**P1011 Option *nom\_option* previously defined***Erreur du compilateur d'aide (projet)*

L'option spécifiée a déjà été définie. Le compilateur ne tient pas compte de cette tentative de redéfinition.

**P1013 Project file extension cannot be .HLP***Erreur du compilateur d'aide (projet)*

Extension de fichier projet .HLP incorrecte. Un fichier projet doit normalement être spécifié avec l'extension .HPJ.

**P1015 Unexpected end-of-file***Erreur du compilateur d'aide (projet)*

Erreur de fin de fichier. Il peut s'agir d'un commentaire ouvert dans un fichier projet ou un fichier inclus.

**P1017 Parameter exceeds maximum length of 128 characters***Erreur du compilateur d'aide (projet)*

Une option, un nom ou un numéro de contexte, un code de compilation conditionnelle ou un autre paramètre dans la ligne indiquée comporte plus de 128 caractères. Cette ligne n'est pas prise en compte.

**P1021 Context number already used in [MAP] section***Erreur du compilateur d'aide (projet)*

Le numéro de contexte dans la ligne indiquée dans le fichier projet a déjà été mise en correspondance avec une autre chaîne contexte. Cette ligne n'est pas prise en compte.

**P1023 Include statements nested too deeply***Erreur du compilateur d'aide (projet)*

L'instruction #include de la ligne indiquée comporte plus de 5 niveaux d'inclusion.

**P1025 Section heading *nom\_section* unrecognized***Erreur du compilateur d'aide (projet)*

Utilisation d'un en-tête de section non reconnu par le compilateur. Cette ligne n'est pas prise en compte.

**P1027 Bracket missing from section heading *nom\_section****Erreur du compilateur d'aide (projet)*

Crochet droit (}) manquant dans l'en-tête de la section spécifiée. Corrigez le code est recompiliez.

**P1029 Section heading missing***Erreur du compilateur d'aide (projet)*

En-tête de section incomplet à la ligne indiquée. Ce message s'affiche également si la première entrée dans le fichier projet ne correspond pas à un en-tête de section. Le compilateur passe à la ligne suivante.



- P1030 Section *nom\_section* previously defined** *Erreur du compilateur d'aide (projet)*  
En-tête de section défini deux fois dans le fichier projet. Les lignes suivant le second en-tête ne sont pas prises en compte et le compilateur reprend au prochain en-tête de section correct.
- P1031 Maximum number of build tags exceeded** *Erreur du compilateur d'aide (projet)*  
Vous pouvez définir un maximum de 30 codes de compilation conditionnelle. Les codes supplémentaires ne sont pas pris en compte.
- P1033 Duplicate build tag in [BUILDTAGS] section** *Erreur du compilateur d'aide (projet)*  
Duplication d'un code de compilation conditionnelle dans la section [BUILDTAGS].
- P1035 Build tag length exceeds maximum** *Erreur du compilateur d'aide (projet)*  
Le code de compilation conditionnelle à la ligne indiquée comporte plus de 32 caractères (limite imposée). Cette ligne n'est pas prise en compte.
- P1037 Build tag *nom\_code* contains invalid characters** *Erreur du compilateur d'aide (projet)*  
Un code de compilation conditionnelle peut uniquement comporter des caractères alphanumériques et le trait de soulignement (\_). Cette ligne n'est pas prise en compte.
- P1039 [BUILDTAGS] section missing** *Erreur du compilateur d'aide (projet)*  
L'option **BUILD** déclare une compilation conditionnelle mais le fichier projet ne comporte pas de section [BuildTags]. Tous les écrans d'aide sont incluse dans la compilation conditionnelle.
- P1043 Too many tags in Build expression** *Erreur du compilateur d'aide (projet)*  
L'expression Build à la ligne indiquée comporte plus de 20 codes de compilation conditionnelle (limite imposée). Cette ligne n'est pas prise en compte par le compilateur.
- P1045 [ALIAS] section found after [MAP] section** *Erreur du compilateur d'aide (projet)*  
La section [Alias] doit précéder la section [Map] dans un fichier projet. Si cet ordre n'est pas respecté, la section [Alias] n'est pas prise en compte.
- P1047 Context string *nom\_contexte* already assigned an alias** *Erreur du compilateur d'aide (projet)*  
Impossible d'enchaîner les instructions du type a=b, puis a=c<\_>. Une chaîne contexte ne peut avoir qu'un seul alias.  
  
Un alias a déjà été défini à la section [Alias] pour la chaîne contexte spécifiée. Ce message indique que la tentative de réaffectation n'est pas prise en compte.
- P1049 Alias string *aliasname* already assigned** *Erreur du compilateur d'aide (projet)*  
Impossible d'enchaîner les instructions a=b et b=c.  
  
Il est illicite d'affecter un autre alias à une chaîne d'alias.
- P1051 Context string *nom\_contexte* cannot be used as alias string** *Erreur du compilateur d'aide (projet)*  
Impossible d'enchaîner les instructions a=b et c=a.  
  
Une chaîne contexte à laquelle un alias a été affectée ne peut pas être utilisée comme alias d'une autre chaîne contexte.
- P1053 Maximum number of font ranges exceeded** *Erreur du compilateur d'aide (projet)*  
Cinq intervalles de corps de polices de caractères peuvent être spécifiés au maximum. Les intervalles supplémentaires ne sont pas pris en compte.
- P1055 Current font range overlaps previously defined range** *Erreur du compilateur d'aide (projet)*  
Chevauchements d'un intervalle de taille de police de caarctères et d'un autre intervalle déjà défini. Adaptez l'un des intervalles afin de supprimer les chevauchements. Le second intervalle n'est pas pris en compte.

- P1056 Unrecognized font name in Forcefont option** *Erreur du compilateur d'aide (projet)*  
 Détection d'un nom de police de caractères non reconnu par le compilateur. Cette police de caractères n'est pas prise en compte et la police par défaut Helvetica est utilisée.
- P1057 Font name too long** *Erreur du compilateur d'aide (projet)*  
 La longueur du nom de la police spécifiée est supérieure à 20 caractères. Cette police n'est pas prise en compte.
- P1059 Invalid multiple-key syntax** *Erreur du compilateur d'aide (projet)*  
 Syntaxe utilisée avec l'option **MULTIKEY** non reconnue.
- P1061 Character already used** *Erreur du compilateur d'aide (projet)*  
 L'identificateur de table de mots clés @char(177) spécifié est déjà utilisé. Choisissez un autre caractère.
- P1063 Characters 'K' and 'k' cannot be used** *Erreur du compilateur d'aide (projet)*  
 Ces caractères sont réservés à la table de mots clés primaire du système d'aide. Choisissez un autre caractère.
- P1065 Maximum number of keyword tables exceeded** *Erreur du compilateur d'aide (projet)*  
 Cinq tables de mots clés peuvent être spécifiées au maximum. Utilisez moins de tables. Les tables supplémentaires ne sont pas prises en compte.
- P1067 Equal sign missing** *Erreur du compilateur d'aide (projet)*  
 Signe égal (=) manquant pour l'option à la ligne indiquée. Vérifiez la syntaxe de l'option.
- P1069 Context string missing** *Erreur du compilateur d'aide (projet)*  
 Chaîne contexte manquante avant un signe égal (=) à la ligne indiquée.
- P1071 Incomplete line in nom\_section section** *Erreur du compilateur d'aide (projet)*  
 Entrée incomplète à la ligne indiquée. Cette ligne n'est pas prise en compte.
- P1073 Unrecognized option in [OPTIONS] section** *Erreur du compilateur d'aide (projet)*  
 Option spécifiée non reconnue par le compilateur. La ligne n'est pas prise en compte.
- P1075 Invalid build expression** *Erreur du compilateur d'aide (projet)*  
 Erreur(s) de syntaxe ou de logique dans l'expression de compilation conditionnelle à la ligne indiquée.
- P1077 Warning level must be 1, 2, or 3** *Erreur du compilateur d'aide (projet)*  
 Erreur de définition de l'option **WARNING**. Les valeurs possibles sont 1, 2 ou 3. Le niveau par défaut fournissant un maximum d'informations (niveau 3) est utilisé par le compilateur.
- P1079 Invalid compression option** *Erreur du compilateur d'aide (projet)*  
 L'option **COMPRESS** peut uniquement être définie à la valeur TRUE ou FALSE. La compilation se poursuit sans compilation.
- P1081 Invalid title string** *Erreur du compilateur d'aide (projet)*  
 L'option **TITLE** définit une chaîne nulle ou qui contient plus de 32 caractères. Le titre est tronqué.
- P1083 Invalid context identification number** *Erreur du compilateur d'aide (projet)*  
 Le numéro de contexte à la ligne indiquée est nul ou contient des caractères non autorisés.
- P1085 Unrecognized text** *Erreur du compilateur d'aide (projet)*  
 Le texte spécifié après le texte correct à la ligne indiquée est indéchiffrable. La ligne n'est pas prise en compte.
- P1086 Invalid font-range syntax** *Erreur du compilateur d'aide (projet)*  
 Erreur de syntaxe dans la définition d'un intervalle de corps de polices de caractères. Cette ligne n'est pas prise en compte par le compilateur. Vérifiez la syntaxe de l'option **MAPFONTSIZE**.

**P1089 Unrecognized sort ordering***Erreur du compilateur d'aide (projet)*

Ordre de tri non reconnu par le compilateur. Pour plus de détails sur la signification de ce message, contactez votre support technique Borland.

**Parameter names are used only with a function body***Erreur du compilateur*

Si vous déclarez une fonction (ou si vous ne la définissez pas avec un corps de fonction), vous devez spécifier des parenthèses vides ou un prototype de fonction. Vous ne pouvez pas déclarer une fonction uniquement avec une liste de noms de paramètre.

Voici des exemples de déclaration :

```
int func(); // déclaration sans prototype--OK
int func(int, int); // declaration avec prototype--OK
int func(int i, int j); // noms de paramètre dans prototype--OK
int func(i, j); // noms de paramètre uniquement--illicite
```

**Parameter *numéro* missing name***Erreur du compilateur*

Dans un en-tête de définition de fonction, ce paramètre comporte un spécificateur de type *numéro* sans nom de paramètre. Bien qu'illicite en C, cette syntaxe est autorisée en C++ mais il n'existe aucun moyen de faire référence à ce paramètre dans la fonction.

**Parameter *paramètre* is never used***Mise en garde du compilateur*

Le paramètre spécifié, déclaré dans la fonction, n'a jamais été utilisé dans le corps de la fonction. Cette erreur est souvent due à un paramètre mal orthographié. Ce message s'affiche également si l'identificateur est redéclaré comme une variable locale automatique dans le corps de la fonction. Le paramètre est masqué par la variable automatique et n'est toujours pas utilisé.

***chemin* - path is too long***Erreur du bibliothécaire*

La longueur du *chemin* d'accès à un fichier de bibliothèque ou un fichier module est supérieure à 64 caractères.

**Pointer to structure required on left side of -> or ->\****Erreur du compilateur*

En C et C++, la partie gauche d'un opérateur flèche (->) doit impérativement être un pointeur. En C++, l'opérateur ->\* est autorisé.

**Possible reference to undefined extern *xxxx::i* in module *module****Mise en garde de l'éditeur de liens*

Une donnée membre statique a été déclarée mais non définie dans l'application.

**Possible unresolved external *sym* referenced from module *mod****Mise en garde de l'éditeur de liens*

Ce message s'affiche uniquement si une donnée membre statique d'une classe a été déclarée mais non définie.

**Possible use of *identificateur* before definition***Mise en garde du compilateur*

La variable indiquée est utilisée dans une expression avant qu'une valeur ne soit affectée. Le compilateur effectue une simple vérification pour déterminer cette condition. Ce message s'affiche si la variable est utilisée physiquement avant toute affectation. Le flux du programme peut évidemment affecter la valeur avant que le programme ne l'utilise.

### Possibly incorrect assignment

*Mise en garde du compilateur*

Ce message s'affiche si le compilateur détecte un opérateur d'affectation utilisé comme opérateur principal d'une expression conditionnelle (dans une instruction **if**, **while** ou **do-while** par exemple). Il s'agit souvent d'une erreur typographique de l'opérateur d'égalité. Pour désactiver ce message, placez cette affectation entre parenthèses et comparez l'expression ainsi formée à zéro de façon explicite. Soit le code :

```
if (a = b) ...
```

doit être réécrit comme suit :

```
if ((a = b) != 0) ...
```

### Program entry point may not reside in an overlay

*Erreur de l'éditeur de liens*

Bien que la majeure partie d'une application puisse être recouverte, l'adresse de départ initiale ne peut se situer dans un segment de recouvrement. Ce message indique en général qu'une tentative de recouvrement du module d'initialisation COX.OBJ a été faite à l'aide de l'option /o, par exemple, avant le module de démarrage.

### Public symbole in module module1 clashes with prior module module2

*Erreur du bibliothécaire*

Un symbole public ne peut apparaître qu'une seule fois dans un fichier de bibliothèque. Un module ajouté à cette bibliothèque contient un *symbole* public qui existe déjà dans un autre module. Ce message indique le nom du second module.

### Public symbole in module nom\_fichier clashes with prior module

*Erreur du bibliothécaire*

Un symbole public ne peut apparaître qu'une seule fois dans un fichier de bibliothèque. Un module ajouté à cette bibliothèque contient un *symbole* public qui existe déjà dans un autre module.

### Public symbol symbole defined in both module mod1 and mod2

*Mise en garde de l'éditeur de liens*

Ce message indique qu'un symbole public est défini dans deux fichiers .OBJ. Dans ce cas, le premier symbole public remplace le second symbole public. L'affichage de ce message peut être contrôlé à l'aide de l'option **-wdup**.

### R2001 Unable to open bitmap file nom\_fichier

*Erreur du compilateur d'aide (RTF)*

Impossible de lire le fichier bitmap spécifié. Il s'agit d'une erreur de fichier DOS.

### R2003 Unable to include bitmap file nom\_fichier

*Erreur du compilateur d'aide (RTF)*

Impossible de trouver ou de lire le fichier bitmap spécifié. Ce message indique une erreur de fichier DOS ou une mémoire insuffisante.

### R2005 Disk full

*Erreur du compilateur d'aide (RTF)*

Impossible d'écrire le fichier ressource d'aide sur le disque. Libérez de l'espace sur l'unité de destination.

### R2009 Cannot use reserved DOS device name for file nom\_fichier

*Erreur du compilateur d'aide (RTF)*

Un fichier a été désigné comme étant COM1, LPT2, PRN, etc. Renommez ce fichier.

### R2013 Output file nom\_fichier already exists as a directory

*Erreur du compilateur d'aide (RTF)*

Un répertoire sous la racine du fichier projet d'aide porte le même nom que le fichier ressource d'aide requis. Déplacez ou renommez ce répertoire.

### R2015 Output file nom\_fichier already exists as read-only

*Erreur du compilateur d'aide (RTF)*

Le fichier spécifié est accessible en lecture uniquement et ne peut être remplacé par le fichier ressource d'aide. Renommez le fichier projet ou modifiez les droits d'accès au fichier.

### R2017 Path for file nom\_fichier exceeds limit of 79 characters

*Erreur du compilateur d'aide (RTF)*

Le nom de chemin absolu ou du chemin relatif à partir du répertoire racine comporte plus de 79 caractères (limite imposée par DOS). Le fichier n'est pas pris en compte.

- R2019 Cannot open file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
 Impossible de lire le fichier spécifié. Ce message indique une erreur de fichier DOS.
- R2021 Cannot find file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
 Impossible de trouver ou de lire le fichier spécifié. Ce message indique une erreur de fichier DOS ou une mémoire insuffisante.
- R2023 Not enough memory to build Help file** *Erreur du compilateur d'aide (RTF)*  
 Mémoire insuffisante pour compiler le fichier d'aide. Déchargez toute application, pilote de périphérique ou programme résidents non utilisé pour libérer de la mémoire.
- R2025 File environment error** *Erreur du compilateur d'aide (RTF)*  
 Nombre de descripteurs de fichier insuffisant pour la poursuite de la compilation. Augmentez les valeurs de FILES= et BUFFERS= dans le fichier CONFIG.SYS et réinitialisez le système.
- R2027 Build tag *nom\_code* not defined in [BUILDTAGS] section of project file** *Erreur du compilateur d'aide (RTF)*  
 Le code de compilation conditionnelle spécifié a été affecté à un écran d'aide mais n'a pas été déclaré dans le fichier projet. Ce code n'est pas pris en compte.
- R2033 Context string in Map section not defined in any topic** *Erreur du compilateur d'aide (RTF)*  
 Une ou plusieurs chaînes contexte ont été définies dans le fichier projet mais aucun écran d'aide ne correspond à ces chaînes.
- R2035 Build expression missing from project file** *Erreur du compilateur d'aide (RTF)*  
 Les écrans sont définis avec des codes de compilation conditionnelle mais aucune expression Build= n'est spécifiée dans le fichier projet. Tous les écrans d'aide sont inclus dans la compilation.
- R2037 File *nom\_fichier* cannot be created, due to previous error(s)** *Erreur du compilateur d'aide (RTF)*  
 Impossible de créer le fichier ressource d'aide car le compilateur ne dispose plus d'écrans à traiter. Corrigez les erreurs précédant cette erreur et recompilez.
- R2039 Unrecognized table formatting in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
 Le compilateur ignore le formatage d'un tableau lorsqu'il n'est pas supporté dans le système d'aide. Reformatez les entrées du tableau de manière linéaire, si possible.
- R2041 Jump *chaîne\_contexte* unresolved in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
 L'écran spécifié contient une chaîne contexte qui indique un écran d'aide inexistant. Vérifiez que la chaîne est correctement orthographiée et que l'écran d'aide est bien inclus dans cette compilation.
- R2043 Hotspot text cannot spread over paragraphs** *Erreur du compilateur d'aide (RTF)*  
 Un mot actif de saut figure sur deux paragraphes consécutifs. Supprimez le formatage de la marque du paragraphe.
- R2045 Maximum number of tab stops reached in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
 Le système d'aide autorise au maximum 32 positions de tabulation. Cette limite est dépassée dans l'écran d'aide spécifié. Par défaut, les autres tabulations ne sont pas prises en compte.
- R2047 File *nom\_fichier* not created** *Erreur du compilateur d'aide (RTF)*  
 Aucun écran à compiler ou l'expression de compilation est incorrecte pour tous les écrans. Le fichier ressource d'aide n'est pas créé.

- R2049 Context string text too long in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
La partie en texte caché qui correspond à la chaîne contexte ne peut dépasser 64 caractères. Cette chaîne contexte n'est pas prise en compte.
- R2051 File *nom\_fichier* is not a valid RTF topic file** *Erreur du compilateur d'aide (RTF)*  
Le fichier spécifié n'est pas un fichier RTF. Vérifiez que l'écran a été sauvegardé dans le format RTF sous votre traitement de texte.
- R2053 Font *nom\_police* in file *nom\_fichier* not in RTF font table** *Erreur du compilateur d'aide (RTF)*  
Une police non définie dans l'en-tête RTF a été utilisée pour l'écran. Le compilateur utilise la police système par défaut.
- R2055 File *nom\_fichier* is not a usable RTF topic file** *Erreur du compilateur d'aide (RTF)*  
Le fichier spécifié contient un en-tête RTF correct mais le fichier n'est pas au format RTF ou est défectueux.
- R2057 Unrecognized graphic format in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Le compilateur reconnaît uniquement les fichiers bitmap Windows. Vérifiez que les formats metafile ou Macintosh n'ont pas été utilisés. Ce graphique n'est pas pris en compte.
- R2059 Context string identifier already defined in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
L'écran spécifié est déjà défini par un identificateur de chaîne contexte. Le compilateur utilise l'identificateur défini dans la première note #.
- R2061 Context string *nom\_contexte* already used in file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
La chaîne contexte spécifiée a déjà été affectée à un autre écran. Le compilateur ne tient pas compte de la deuxième chaîne et l'écran d'aide ne peut plus être atteint par saut hypertexte.
- R2063 Invalid context-string identifier for topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
L'identificateur de chaîne contexte contient des caractères non alphanumériques ou est nul. Aucun identificateur n'est affecté à l'écran.
- R2065 Context string defined for index topic is unresolved** *Erreur du compilateur d'aide (RTF)*  
L'écran d'aide de l'index défini dans le fichier projet est introuvable. Le compilateur utilise comme index le premier écran d'aide trouvé dans la compilation.
- R2067 Footnote text too long in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Le texte des notes de bas de page est limité à mille caractères. Cette note de bas de page n'est pas prise en compte.
- R2069 Build tag footnote not at beginning of topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
L'écran d'aide indiqué contient une note de bas de page pour un code de compilation conditionnelle, mais ce code n'est pas le premier caractère de l'écran d'aide. Le code est ignoré pour cet écran d'aide.
- R2071 Footnote text missing in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
L'écran spécifié contient une note de bas de page ne comportant aucun caractère.
- R2073 Keyword string is null in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Une note de bas de page pour un mot clé a été spécifiée mais ne comporte aucun caractère.
- R2075 Keyword string too long in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Le texte de la note de bas de page pour le mot clé dans l'écran indiqué comporte plus de 255 caractères. Les caractères supplémentaires ne sont pas pris en compte.

- R2077 Keyword(s) defined without title in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Un ou plusieurs mot(s) clé ont été définis pour cet écran auquel aucun titre n'est affecté. La fonction Recherche de la fenêtre d'aide affichera Rubrique sans titre<< pour cet écran d'aide.
- R2079 Browse sequence string is null in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
La note de bas de page d'une séquence d'écrans pour l'écran spécifié ne contient aucun caractère de séquence.
- R2081 Browse sequence string too long in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
La note de bas de page d'une séquence d'écrans pour l'écran spécifié dépasse 128 caractères. La séquence n'est pas prise en compte.
- R2083 Missing sequence number in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Signe deux-points (:) ajouté par erreur à la fin d'un numéro de séquence de recherche. Supprimez ce signe ou entrez un numéro de séquence complémentaire.
- R2085 Sequence number already defined in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Note de bas de page de séquence d'écrans déjà définie pour l'écran indiqué. La deuxième séquence n'est pas prise en compte.
- R2087 Build tag too long** *Erreur du compilateur d'aide (RTF)*  
Un code de compilation conditionnelle pour l'écran indiqué comporte plus de 32 caractères. Ce code n'est pas pris en compte pour cet écran d'aide.
- R2089 Title string null in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
La note de bas de page de titre de l'écran indiqué ne comporte aucun caractère. Aucun titre n'est affecté à cet écran d'aide.
- R2091 Title too long in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Le titre de l'écran indiqué comporte plus de 128 caractères. Les caractères supplémentaires ne sont pas pris en compte.
- R2093 Title titlename in topic *num\_écran* of file *nom\_fichier* used previously** *Erreur du compilateur d'aide (RTF)*  
Le titre indiqué a déjà été affecté à un autre écran.
- R2095 Title defined more than once in topic *num\_écran* of file *nom\_fichier*** *Erreur du compilateur d'aide (RTF)*  
Plusieurs notes de bas de page de titre ont été spécifiées pour l'écran indiqué. Le compilateur utilise la première chaîne titre détectée.
- R2501 Using old key-phrase table** *Erreur du compilateur d'aide (RTF)*  
Une compression maximale peut être obtenue uniquement en supprimant le fichier .PH avant chaque recompilation des écrans d'aide.
- R2503 Out of memory during text compression** *Erreur du compilateur d'aide (RTF)*  
Mémoire insuffisante pour comprimer le fichier ressource d'aide. La compilation se poursuit sans compression. Déchargez toute application, pilote de périphérique ou programme résident non utilisé pour libérer de la mémoire.
- R2505 File environment error during text compression** *Erreur du compilateur d'aide (RTF)*  
Nombre de descripteurs de fichier insuffisant pour la compression. La compilation se poursuit sans compression du fichier ressource d'aide. Augmentez les valeurs FILES= et BUFFERS= dans le fichier CONFIG.SYS et réinitialisez le système.
- R2507 DOS file error during text compression** *Erreur du compilateur d'aide (RTF)*  
Erreur d'accès à un fichier sur disque lors de la compression. La compilation se poursuit sans compression du fichier ressource d'aide.

- R2509 Error during text compression** *Erreur du compilateur d'aide (RTF)*  
 L'une des erreurs de compression (R2503, R2505 ou R2507) s'est produite. La compilation se poursuit sans compression du fichier ressource d'aide.
- R2701 Internal error** *Erreur du compilateur d'aide (RTF)*
- R2703 Internal error** *Erreur du compilateur d'aide (RTF)*
- R2705 Internal error** *Erreur du compilateur d'aide (RTF)*
- R2707 Internal error** *Erreur du compilateur d'aide (RTF)*
- R2709 Internal error** *Erreur du compilateur d'aide (RTF)*  
 Pour plus de détails sur la signification de ces messages contactez votre support technique Borland.
- Record kind *num* found, expected theadr or lheadr in module *nom\_fichier*** *Erreur du bibliothécaire*  
 Le bibliothécaire n'a pas pu interprété l'enregistrement d'en-tête du module objet ajouté à la bibliothèque et a supposé que le module était incorrect.
- Record length *len* exceeds available buffer in module *module*** *Erreur du bibliothécaire*  
 Cette erreur se produit lorsque la longueur d'enregistrement *len* dépasse la taille du tampon disponible du module *module* et lorsque le bibliothécaire ne dispose plus de mémoire dynamique.
- Record type *type* found, expected theadr or lheadr in *module*** *Erreur du bibliothécaire*  
 Le bibliothécaire a détecté le type *type* au lieu du type d'enregistrement THEADR ou LHEADER dans le module *module*.
- Redefinition of *macro* is not identical** *Mise en garde du compilateur*  
 Le fichier source contient une rédefinition d'une macro dont le texte n'est pas exactement identique à la première définition. Le nouveau texte est pris en compte.
- Redefinition of target *nom\_fichier*** *Erreur de MAKE*  
 Le nom de fichier indiqué figure dans la partie gauche de plusieurs règles explicites.
- Reference initialized with *type1*, needs lvalue of type *type2*** *Erreur du compilateur*  
 Une variable ou un paramètre référence qui n'est pas déclaré comme constante doit être initialisé avec une lvalue de type approprié. Ce message indique que l'initialisateur n'était pas une lvalue ou que son type ne correspondait pas à la référence initialisée.
- Reference member *membre* in class without constructors** *Erreur du compilateur*  
 Une classe contenant des références membres doit comporter au moins un constructeur défini par l'utilisateur pour pouvoir initialiser ces membres.
- Reference member *membre* initialized with a non-reference parameter** *Erreur du compilateur*  
 Une tentative de lien entre une référence membre à un paramètre a été faite dans un constructeur. L'objet paramètre perdant toute existence lorsque le constructeur renvoie une valeur, la référence membre ferait alors référence à un objet non défini. Cette erreur assez fréquente peut se traduire par un blocage du système ou des résultats de programme inattendus.
- Reference member *membre* is not initialized** *Erreur du compilateur*  
 Une référence doit toujours être initialisée. Un membre de classe de type référence doit avoir un initialisateur dans tous les constructeurs de cette classe. Vous ne devez donc pas vous attendre à ce que le compilateur génère des constructeurs pour cette classe car il n'a aucun moyen de savoir comment initialiser ces références.



**Reference member** *membre needs a temporary for initialization**Erreur du compilateur*

La valeur initiale spécifiée pour un type référence ne correspondait pas à une lvalue du type référencé. Le compilateur doit créer une variable temporaire pour cette initialisation. Dans la mesure où il n'existe aucun endroit évident pour stocker cette variable, l'initialisation est illicite.

**Reference variable** *variable must be initialized**Erreur du compilateur*

Cet objet C++ a été déclaré comme une référence mais n'a pas été initialisée. Une référence doit impérativement être initialisée au moment de sa déclaration.

**Register allocation failure***Erreur du compilateur*

Indication d'Erreur du compilateur. Une expression trop complexe dans la fonction indiquée a empêché la génération du code correspondant par le générateur de code. Essayez de simplifier cette fonction. Si le problème persiste, notez-le et contactez votre support technique Borland.

**Relocation item exceeds 1MB DOS limit***Erreur de l'éditeur de liens*

Le format de fichier exécutable DOS ne gère pas les éléments de réaffectation de plus de 1 Mo. Bien que DOS ne puisse pas charger une image aussi volumineuse, les programmes d'accès à la mémoire étendue DOS le permettent, et TLINK supporte la génération d'images plus volumineuses que celles gérables par DOS. Même si l'image est chargée à l'aide d'un tel programme, le format de fichier exécutable DOS est limité à la description d'éléments de réaffectation dans le premier méga-octet de l'image.

**Relocation offset overflow***Erreur de l'éditeur de liens*

Cette erreur se produit uniquement pour des modules objet 32 bits et indique un décalage d'élément de réaffectation (référence de segment) supérieur à 64 ko (limite imposée par DOS).

**Relocation table overflow***Erreur de l'éditeur de liens*

Cette erreur se produit uniquement pour des modules objet 32 bits. Le fichier lié contient plus de références de base que la table de réaffectation DOS standard ne peut contenir (les références de base sont créées en général par des appels à des fonctions far).

**Resident Name Table is greater than 64K***Mise en garde de l'éditeur de liens*

La taille maximale d'une table de noms Resident est de 64 ko (conformément aux normes de l'industrie en termes de spécifications de fichiers exécutables). L'édition de liens se poursuit sans tenir compte d'autres noms Resident détectés par la suite.

**Restarting compile using assembly***Mise en garde du compilateur*

Le compilateur a détecté une instruction **ASM** accompagnée d'une option **-B** ou d'une instruction en ligne **#pragma**. Le compilateur reprend en utilisant les fonctionnalités du langage assembleur.

**Results are safe in file** *nom\_fichier**Mise en garde du bibliothécaire*

Le bibliothécaire a compilé la bibliothèque dans un fichier temporaire mais ne peut attribuer le nom de la bibliothèque requis à ce fichier. Le fichier temporaire n'est pas supprimé (la bibliothèque est ainsi conservée).

**Rule line too long***Erreur de MAKE*

Une règle implicite ou explicite comporte plus de 4 096 caractères.

**Segment** *segment exceeds 64K**Erreur de l'éditeur de liens*

Ce message indique qu'un nombre trop important de données a été défini pour un segment de données ou de code une fois les segments portant le même nom combinés à partir de différents fichiers sources par TLINK.

**Segment *segment* is in two groups: *groupe1* and *groupe2****Mise en garde de l'éditeur de liens*

Conflit entre deux groupes détecté par l'éditeur de liens. Cette erreur se produit généralement dans des programmes en langage assembleur et indique que deux modules ont affecté le même segment à deux groupes différents.

**Self relative fixup overflowed in module *module****Mise en garde de l'éditeur de liens*

Ce message s'affiche si une référence auto-relative (un appel en général) est effectuée à partir d'un segment physique sur un autre segment. Ceci se produit en général dans un programme en langage assembleur et également si vous utilisez les options de spécification de segment avec le compilateur. Si la référence est effectuée d'un segment de code à un autre, aucun problème ne devrait se produire. S'il s'agit d'une référence d'un segment de code à un segment de données, le message indique une erreur probable dans le code assembleur. L'affichage de ce message peut être contrôlé à l'aide de l'option **-w**.

**Size of *identificateur* is unknown or zero***Erreur du compilateur*

Cette identificateur est utilisé dans un contexte où il est nécessaire de spécifier sa taille. Par exemple, un modèle de **structure** a peut être été déclaré (sans que la **structure** n'ait été définie) ou un tableau **extern** déclaré sans taille. Si c'est le cas, il est impossible d'avoir des références à ce type d'élément (comme **sizeof**) ou de donner la correspondance d'un pointeur vers ce type. Modifiez votre déclaration pour que la taille de *identificateur* soit disponible.

**sizeof may not be applied to a bit field***Erreur du compilateur*

**sizeof** renvoie la taille en octet d'un objet de données (ce qui ne s'applique pas à un champ de bits).

**sizeof may not be applied to a function***Erreur du compilateur*

**sizeof** est applicable uniquement à des objets de données et non à des fonctions. Vous pouvez déterminer la taille d'un pointeur vers une fonction.

**Size of the type is unknown or zero***Erreur du compilateur*

Ce type est utilisé dans un contexte où il est nécessaire de spécifier sa taille. Par exemple, un modèle de **structure** a peut être été déclaré (sans que la **structure** n'ait encore été définie). Si c'est le cas, il est impossible d'avoir des références à ce type d'élément (comme **sizeof**) ou de donner la correspondance d'un pointeur vers ce type. Modifiez votre déclaration pour que la taille de ce type soit disponible.

***identificateur* specifies multiple or duplicate access***Erreur du compilateur*

Une classe de base peut être déclarée **publique** ou **privée** mais pas simultanément avec les deux statuts. Un spécificateur d'accès doit apparaître une seule fois pour une classe de base donnée.

**Stack overflow***Erreur d'exécution*

La taille de pile par défaut d'un programme en Borland C++ est de 5 120 octets. Bien que cette taille soit suffisante pour la plupart des programmes, certains programmes exécutant des fonctions récursives ou stockant de nombreuses données locales peuvent entraîner un débordement de pile. Ce message s'affiche uniquement si l'option de vérification de pile est activée. Si ce message s'affiche, essayez d'augmenter la taille de la pile ou de minimiser l'utilisation de la pile dans votre programme. Pour modifier la taille de la pile, redéfinissez la variable `_stklen`. Essayez de passer à un modèle de mémoire plus grand pour contenir la pile agrandie.

L'exemple ci-dessus illustre comment réduire le volume de données locales utilisées par une fonction. La variable `buffer` est déclarée statique et ne consomme aucun espace de pile (à la différence de `list`).

```
void anyfunction(void) {
 static int buffer[2000]; /* réside dans le segment de données */
 int list[2000]; /* réside sur la pile */
}
```

Les inconvénients d'une déclaration de variable locale statique sont les suivants :

1. Elle occupe un espace mémoire en permanence non disponible pour les variables globales ni pour le tas (ceci n'est qu'un inconvénient mineur) ;
2. La fonction ne peut plus être réentrante. Si la fonction est appelée de façon récursive ou asynchrone et que chaque appel de la fonction doit disposer de sa propre copie de la variable, vous ne pouvez pas dans ce cas la déclarer comme statique. En effet, à chaque appel de la fonction, celle-ci utilise le même espace alloué en mémoire pour cette variable au lieu d'allouer un espace différent à chaque appel. Si la fonction essaie de s'appeler elle-même (de façon récursive) ou au même moment où il est fait appel à cette fonction (de façon asynchrone), elle risque de provoquer un problème de partage. Ceci n'est pas un problème pour la plupart des programmes DOS.

**Statement missing ;** *Erreur du compilateur*

Point-virgule manquant à la fin d'une instruction.

**Storage class *classe\_stockage* is not allowed here** *Erreur du compilateur*

Impossible de spécifier la classe de stockage indiquée. Vous avez probablement spécifier deux classes de stockage alors qu'une seule est autorisée.

**String type not allowed with this operand** *Erreur de MAKE*

L'opérande spécifié ne peut pas être utilisé pour comparer des types chaîne. Utilisez les opérandes ==, !=, <, >, <=, ou >=.

**Structure passed by value** *Mise en garde du compilateur*

Une structure a été passée par valeur en tant qu'argument à une fonction sans prototype. Cette erreur qui consiste à omettre l'opérateur d'adressage (&) devant une structure lorsqu'elle doit être passée en argument est fréquente en programmation ; elle reste cependant acceptable car une structure peut effectivement être passée par valeur. Ce message s'affiche à titre de mise en garde.

**Structure required on left side of . or .\*** *Erreur du compilateur*

La partie gauche d'un opérateur point (.) ou point-étoile en C++ doit correspondre à un type structure.

**Structure size too large** *Erreur du compilateur*

La taille d'une structure est supérieure à 64 ko.

**Style of function definition is now obsolete** *Mise en garde du compilateur*

En C++, la définition "en C ancienne manière" de la fonction suivante n'est pas autorisée :

```
int func(p1, p2)
int p1, p2;
{...}
```

**Subscripting missing ]** *Erreur du compilateur*

Crochet fermant manquant à la fin d'une expression d'indilage. Vérifiez les opérateurs spécifiés ou les parenthèses.

**Superfluous & with function** *Mise en garde du compilateur*

Opérateur d'adressage (&) inutile avec le nom de la fonction (tout opérateur de ce type détecté est supprimé).

**Suspicious pointer conversion** *Mise en garde du compilateur*

Le compilateur a détecté une conversion de pointeur qui a fait pointer le pointeur vers un type différent. Utilisez un opérateur de transtypage pour supprimer ce message si la conversion est correcte.

**Switch selection expression must be of integral type***Erreur du compilateur*

L'expression de sélection entre parenthèses dans une instruction **switch** doit correspondre à un type entier (**char**, **short**, **int**, **long**, **enum**). Vous pourriez utiliser un opérateur de transtypage pour respecter cette condition.

**Switch statement missing (***Erreur du compilateur*

Parenthèse gauche manquante après le mot clé **switch** dans une instruction **switch**.

**Switch statement missing )***Erreur du compilateur*

Parenthèse droite manquante après l'expression test dans une instruction **switch**.

**nom\_fichier (num\_ligne): Syntax error***Erreur de l'éditeur de liens*

Erreur de syntaxe dans le fichier de définition de modules et à la ligne indiqués.

**Table limit exceeded***Erreur de l'éditeur de liens*

Débordement de l'une des tables internes de l'éditeur de liens. Ce message indique que le programme comporte un nombre trop important de symboles publics, de symboles externes ou de définitions de segments logiques gérables par l'éditeur de liens. Notez que chaque instance d'un nom de segment distinct dans un fichier objet compte pour un segment logique. Si ce segment est défini par deux fichiers objets, il en résulte deux segments logiques.

**Target index of FIXUP is 0 in module module***Erreur de l'éditeur de liens*

Erreur du convertisseur.

**Template argument must be a constant expression***Erreur du compilateur*

Un argument de classe template doit correspondre à une expression constante (ou être de type approprié), soit une expression de type entier constant ou une adresse d'objet ou de fonction avec membres ou édition de liens externe.

**Template class nesting too deep: 'class'***Erreur du compilateur*

Le compilateur impose une certaine limite des niveaux d'imbrication de classes templates. Cette limite est en général dépassée par une dépendance de classe template récursive. Ce message d'erreur est affiché pour toutes les classes template imbriquées (il est ainsi facile de détecter la récursivité). Le message **Out of memory** s'affiche ensuite systématiquement.

A titre d'exemple, analysez le groupe de classes template suivant :

```
template<class T> class A
{
 friend class B<T*>;
};
```

```
template<class T> class B
{
 friend class A<T>;
};
```

```
A<int> x;
```

Ce code génère les messages d'erreur suivants :

```
Error: Template class nesting too deep: 'B<int * * * * *>'
Error: Template class nesting too deep: 'A<int * * * *>'
Error: Template class nesting too deep: 'B<int * * * *>'
Error: Template class nesting too deep: 'A<int * * *>'
Error: Template class nesting too deep: 'B<int * * *>'
Error: Template class nesting too deep: 'A<int * *>'
Error: Template class nesting too deep: 'B<int * *>'
Error: Template class nesting too deep: 'A<int *>'
Error: Template class nesting too deep: 'B<int *>'
Error: Template class nesting too deep: 'A<int*>'
Fatal: Out of memory
```

**Template function argument *argument not used in argument types***

*Erreur du compilateur*

L'argument indiqué n'a pas été utilisé dans la liste d'arguments de la fonction. La liste d'arguments d'une fonction template doit utiliser tous les arguments formels de template. A cette seule condition, une instance de la fonction template peut être générée à partir des types d'argument spécifiés.

**Template functions may only have type-arguments**

*Erreur du compilateur*

Une fonction template a été déclarée avec un argument non type. Cette déclaration n'est pas autorisée car il est alors impossible de spécifier la valeur lors de l'appel de la fonction template.

**Templates can only be declared at file level**

*Erreur du compilateur*

Un template ne peut pas être déclaré dans une classe ou une fonction. Il est possible de le déclarer uniquement au niveau global (au niveau du fichier).

**Templates must be classes or functions**

*Erreur du compilateur*

La déclaration d'un template doit spécifier un type classe ou une fonction.

**Temporary used to initialize *identificateur***

*Mise en garde du compilateur*

**Temporary used for parameter *numéro* in call to *fonction***

*Mise en garde du compilateur*

**Temporary used for parameter *paramètre* in call to *fonction***

*Mise en garde du compilateur*

**Temporary used for parameter *numéro***

*Mise en garde du compilateur*

**Temporary used for parameter *paramètre***

*Mise en garde du compilateur*

En C++, une variable ou un paramètre de type référence n'accepte pour affectation qu'une référence à un objet du même type. Si les types sont incompatibles, la valeur effective est affectée à une valeur temporaire de type approprié. L'adresse de cette variable temporaire est alors affectée à la variable ou au paramètre de type référence. Ce message indique que la variable ou le paramètre de type référence ne font pas référence à la variable attendue mais à une variable temporaire qui n'est autrement pas utilisée.

Dans l'exemple suivant, la fonction *f* nécessite une référence à un entier (**int**) et *c* est un caractère (**char**) :

```
f(int&);
char c;
f(c);
```

Au lieu d'appeler la fonction *f* avec l'adresse de *c*, le compilateur génère un code équivalent au code source C++ :

```
int X = c, f(X);
```

**Terminated by user**

Vous avez mis fin explicitement à l'édition de liens.

*Erreur de l'éditeur de liens*

**The '...' handler must be last**

Le gestionnaire '...' doit figurer à la fin d'une liste de gestionnaires catch.

*Erreur du compilateur*

**The combinations '+\*' or '\*+' are not allowed**

Impossible d'ajouter ou d'extraire un module objet d'une bibliothèque en une seule opération. Vous avez probablement indiqué l'opération '+-'.

*Erreur du bibliothécaire*

**The constructor *constructeur* is not allowed**

Les constructeurs de la forme **X::X** ne sont pas autorisés. La syntaxe correcte d'un constructeur de copie est la suivante : **X::X(const X&)**.

*Erreur du compilateur*

**The value for *identificateur* is not within the range of an int**

La valeur d'un énumérateur doit être représentée sous la forme d'un entier. La valeur que vous avez essayé d'affecter n'est pas comprise dans l'intervalle des valeurs entières. En C++, vous devez utiliser un entier de type **constant**.

*Erreur du compilateur*

**'this' can be used only within a member function**

En C++, **this** est un mot réservé à utiliser uniquement dans une fonction de membre de classe.

*Erreur du compilateur*

**This initialization is only partly bracketed**

Erreur d'accolade dans une initialisation après avoir sélectionné EDI Options|Compiler|Messages|ANSI violations. Chaque membre d'une structure peut être initialisé en étant spécifié entre accolades. Si un membre est un tableau ou une structure, il est possible d'imbriquer plusieurs niveaux d'accolades. Cette syntaxe garantit que le compilateur affecte les valeurs aux membres exactement comme vous l'avez spécifié. Ce message s'affiche si vous oubliez de spécifier certaines des accolades facultatives.

*Mise en garde du compilateur*

**Too few arguments in template class name *template***

Certaines valeurs des paramètres formels d'un nom de classe template n'ont pas été spécifiées.

*Erreur du compilateur*

**Too few parameters in call**

Les arguments d'un appel de fonction avec prototype (via un pointeur de fonction) n'ont pas tous été spécifiés. Tous les paramètres doivent être spécifiés.

*Erreur du compilateur*

**Too few parameters in call to *fonction***

Les arguments d'un appel de la fonction indiquée (déclarée avec un prototype) n'ont pas tous été spécifiés.

*Erreur du compilateur*

**Too many commas on command-line**

Entrée incorrecte dans la ligne de commande.

*Erreur de l'éditeur de liens*

**Too many decimal points**

Plusieurs virgules décimales ont été détectées dans une constante en virgule flottante.

*Erreur du compilateur*

|                                                                                                                                                                                                                                      |                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| <b>Too many default cases</b>                                                                                                                                                                                                        | <i>Erreur du compilateur</i>        |
| Plusieurs expressions <b>default</b> ont été détectées dans une instruction <b>switch</b> .                                                                                                                                          |                                     |
| <b>Too many default libraries</b>                                                                                                                                                                                                    | <i>Erreur du compilateur</i>        |
| L'éditeur de liens ne peut gérer qu'un maximum de 128 bibliothèques.                                                                                                                                                                 |                                     |
| <b>Too many error or warning messages</b>                                                                                                                                                                                            | <i>Erreur du compilateur</i>        |
| Un maximum de 255 messages d'erreur et de mise en garde peut être défini avant que le compilateur ne s'arrête.                                                                                                                       |                                     |
| <b>Too many error or warning messages</b>                                                                                                                                                                                            | <i>Erreur de l'éditeur de liens</i> |
| Le nombre de messages affichés par le compilateur est supérieur à 255. Ce message indique que TLINK a atteint cette limite.                                                                                                          |                                     |
| <b>Too many errors</b>                                                                                                                                                                                                               | <i>Erreur de l'éditeur de liens</i> |
| Le compilateur a détecté un nombre d'erreurs plus important que celui autorisé par l'option <b>-E</b> .                                                                                                                              |                                     |
| <b>Too many exponents</b>                                                                                                                                                                                                            | <i>Erreur du compilateur</i>        |
| Plusieurs exposants ont été détectés dans une constante en virgule flottante.                                                                                                                                                        |                                     |
| <b>Too many initializers</b>                                                                                                                                                                                                         | <i>Erreur du compilateur</i>        |
| Un nombre d'initialisateurs plus important que celui autorisé par la déclaration initialisée a été détecté.                                                                                                                          |                                     |
| <b>Too many LNAMEs</b>                                                                                                                                                                                                               | <i>Erreur de l'éditeur de liens</i> |
| Un fichier .OBJ ne doit pas comporter plus de 256 LNAME (limite imposée pr TLINK).                                                                                                                                                   |                                     |
| <b>Too many segments</b>                                                                                                                                                                                                             | <i>Erreur de l'éditeur de liens</i> |
| Un fichier .OBJ ne doit pas comporter plus de 256 SEGDEF (limite imposée par TLINK).                                                                                                                                                 |                                     |
| <b>Too many storage classes in declaration</b>                                                                                                                                                                                       | <i>Erreur du compilateur</i>        |
| Une déclaration ne peut pas avoir plusieurs classes de stockage.                                                                                                                                                                     |                                     |
| <b>Too many suffixes in .SUFFIXES list</b>                                                                                                                                                                                           | <i>Erreur de MAKE</i>               |
| La liste de suffixes peut comporter un maximum de 255 suffixes autorisés.                                                                                                                                                            |                                     |
| <b>Too many types in declaration</b>                                                                                                                                                                                                 | <i>Erreur du compilateur</i>        |
| Une déclaration ne peut avoir plusieurs types de base ( <b>char</b> , <b>int</b> , <b>float</b> , <b>double</b> , <b>struct</b> , <b>union</b> , <b>enum</b> ou <b>typedef-nom</b> ).                                                |                                     |
| <b>Too much global data defined in file</b>                                                                                                                                                                                          | <i>Erreur du compilateur</i>        |
| La somme des déclarations de données globales dépasse 64 ko. Vérifiez la déclaration d'un tableau qui pourrait être trop grand, modifiez le programme ou utilisez des variables <b>far</b> si toutes les déclarations sont requises. |                                     |
| <b>Trying to derive a far class from the huge base base</b>                                                                                                                                                                          | <i>Erreur du compilateur</i>        |
| Si une classe de base est déclarée <b>huge</b> (ou prend cette valeur par défaut), toutes les classes dérivées doivent également être déclarées <b>huge</b> .                                                                        |                                     |
| <b>Trying to derive a far class from the near base base</b>                                                                                                                                                                          | <i>Erreur du compilateur</i>        |
| Si une classe de base est déclarée <b>near</b> (ou prend cette valeur par défaut), toutes les classes dérivées doivent également être déclarées <b>near</b> .                                                                        |                                     |
| <b>Trying to derive a huge class from the far base base</b>                                                                                                                                                                          | <i>Erreur du compilateur</i>        |
| Si une classe de base est déclarée <b>far</b> (ou prend cette valeur par défaut), toutes les classes dérivées doivent également être déclarées <b>far</b> .                                                                          |                                     |

**Trying to derive a huge class from the near base base***Erreur du compilateur*

Si une classe de base est déclarée **near** (ou prend cette valeur par défaut), toutes les classes dérivées doivent également être déclarées **near**.

**Trying to derive a near class from the far base base***Erreur du compilateur*

Si une classe de base est déclarée **far** (ou prend cette valeur par défaut), toutes les classes dérivées doivent également être déclarées **far**.

**Trying to derive a near class from the huge base base***Erreur du compilateur*

Si une classe de base est déclarée **huge** (ou prend cette valeur par défaut), toutes les classes dérivées doivent également être déclarées **huge**.

**Two consecutive dots***Erreur du compilateur*

Deux points consécutifs n'ont aucune signification dans un programme en C. Vous avez peut-être voulu spécifier des points de suspension (...), un séparateur décimal ou un opérateur de sélection de membre (.).

**Two operands must evaluate to the same type***Erreur du compilateur*

Les types des expressions de part et d'autre du signe deux-points de l'opérateur d'expression conditionnelle (?:) doivent être identiques sauf pour les conversions usuelles (**char** en **int**, **float** en **double** ou **void\*** en pointeur particulier). Ce message indique que les types sont différents et ne sont pas convertis automatiquement. Corrigez ou convertissez explicitement les types en question.

**Type type is not a defined class with virtual functions***Erreur du compilateur*

Un `dynamic_cast` a été utilisé avec un pointeur vers un type de classe qui n'est pas défini ou qui n'a pas de fonctions membres virtuelles.

*Remarque* sur les erreurs de compatibilité de types :

Lors de la compilation d'un programme C++, les messages d'erreur suivants qui indiquent une incompatibilité de type sont toujours précédés d'un autre message qui explique la raison exacte de cette incompatibilité. En général, il s'agit du message "Cannot convert *type1* to *type2*" mais d'autres raisons sont possibles.

**Type mismatch in default argument value***Erreur du compilateur***Type mismatch in default value for parameter paramètre***Erreur du compilateur*

La valeur du paramètre par défaut indiquée n'a pas pu être convertie dans le type du paramètre. Le premier message s'affiche si aucun nom n'a été affecté aux paramètres. Reportez-vous à la remarque sur les messages d'erreur indiquant une incompatibilité de types.

**Type mismatch in parameter numéro***Erreur du compilateur*

La fonction appelée, par l'intermédiaire d'un pointeur de fonction, a été déclarée avec un prototype. Le paramètre *numéro* indiqué (les paramètres sont comptés de gauche à droite dans l'ordre croissant) n'a pas pu être converti dans le type du paramètre déclaré. Reportez-vous à la remarque sur les messages d'erreur indiquant une incompatibilité de types.

**Type mismatch in parameter numéro in call to fonction***Erreur du compilateur*

La fonction indiquée est déclarée avec un prototype dans le fichier source et le paramètre *numéro* (les paramètres sont comptés de gauche à droite dans l'ordre croissant) n'a pas pu être converti dans le type du paramètre déclaré. Reportez-vous à la remarque sur les messages d'erreur indiquant une incompatibilité de types.



- Type mismatch in parameter** *paramètre* *Erreur du compilateur*  
 La fonction appelée via un pointeur de fonction est déclarée avec un prototype dans le fichier source et le paramètre spécifié n'a pas pu être converti dans le type du paramètre déclaré. Reportez-vous à la remarque sur les messages d'erreur indiquant une incompatibilité de types.
- Type mismatch in parameter** *paramètre in call to fonction* *Erreur du compilateur*  
 La fonction appelée via un pointeur de fonction est déclarée avec un prototype dans le fichier source et le paramètre spécifié n'a pas pu être converti dans le type du paramètre déclaré. Reportez-vous au message **Type mismatch in parameter** *paramètre* et à la remarque sur les messages d'erreur indiquant une incompatibilité de types.
- Type mismatch in parameter** *paramètre in template class name* *template* *Erreur du compilateur*
- Type mismatch in parameter** *numéro in template class name* *template* *Erreur du compilateur*  
 La valeur d'argument de template fournie pour le paramètre spécifié ne correspondait pas exactement au type du paramètre formel de template. Reportez-vous à la remarque sur les messages d'erreur indiquant une incompatibilité de types.
- Type mismatch in redeclaration of** *identificateur* *Erreur du compilateur*  
 Redéclaration de l'identificateur indiqué avec un type autre que le type initial. Ce message s'affiche par exemple lorsqu'une fonction est appelée et déclarée ensuite pour renvoyer une valeur d'un type autre qu'un entier. Vous devez dans ce cas déclarer la fonction avant de l'appeler pour la première fois. Reportez-vous à la remarque sur les messages d'erreur indiquant une incompatibilité de types.
- Type name expected** *Erreur du compilateur*  
 L'une des erreurs suivantes s'est produite :
- Aucun nom de type ou aucune classe de stockage n'a été spécifié lors de la déclaration d'une variable de niveau de fichier ou d'un champ de **structure**.
  - Aucun type de nom n'a été spécifié dans une déclaration **typedef**.
  - Le nom du destructeur spécifié dans la déclaration d'un destructeur d'une classe C++ ne correspondait pas à un nom de type (il doit correspondre au nom de sa classe).
  - Le nom spécifié pour une classe de base C++ ne correspondait pas à un nom de classe.
- Type qualifier** *identificateur must be a struct or class name* *Erreur du compilateur*  
 Le qualificateur d'une construction `qual::identificateur` ne correspond pas au nom d'une **structure** ou d'une **classe**.
- Unable to create output file** *nom\_fichier* *Erreur du compilateur*  
 Disque de travail saturé ou protégé contre l'écriture ou répertoire de sortie inexistant. Si le disque est saturé, essayez de supprimer des fichiers non utilisés et recompilez. Si le disque est protégé contre l'écriture, déplacez les fichiers sources sur un disque accessible en écriture et recompilez.
- Unable to create turboc.\$ln** *Erreur du compilateur*  
 Le compilateur ne peut pas créer le fichier temporaire TURBOC.\$LN car le disque est saturé ou protégé contre l'écriture.
- Unable to execute command:** *commande* *Erreur de MAKE*  
 La commande indiquée n'a pas pu être exécutée. L'une des erreurs suivantes s'est produite :
- le fichier de la commande n'a pas été trouvé ;
  - le nom du fichier de commande est incorrectement spécifié ;

- le répertoire de permutation indiqué est saturé ou inexistant ;
- le fichier de la commande est défectueux (raison moins probable).

**Unable to execute command** *commande* *Erreur du compilateur*  
 TLINK ou TASM introuvable ou disque défectueux.

**Unable to open file** *nom\_fichier* *Erreur de MAKE*

**Unable to open** *nom\_fichier* *Erreur de l'éditeur de liens*  
 Nom du fichier indiqué inexistant ou incorrectement spécifié.

**Unable to open** *nom\_fichier* **for output** *Erreur du bibliothécaire*  
 Le bibliothécaire ne peut pas ouvrir le fichier indiqué pour la sortie. Ce message indique un espace insuffisant pour la bibliothèque cible ou un fichier de listage. Il s'affiche également si le fichier de bibliothèque cible existe mais n'est accessible qu'en lecture seulement.

**Unable to open include file** *nom\_fichier* *Erreur du compilateur*  
 Le compilateur n'a pas pu trouver le fichier indiqué. Ce message peut être dû à une auto-inclusion du fichier #include ou au fait que la variable FILES n'est pas définie dans le fichier CONFIG.SYS de votre répertoire racine (spécifiez FILES=20). Vérifiez que le fichier indiqué existe.

**Unable to open include file** *nom\_fichier* *Erreur de MAKE*  
 Le compilateur n'a pas pu trouver le fichier indiqué. Ce message peut être dû à une auto-inclusion du fichier !include ou au fait que la variable FILES n'est pas définie dans le fichier CONFIG.SYS de votre répertoire racine (spécifiez FILES=20). Vérifiez que le fichier indiqué existe.

**Unable to open input file** *nom\_fichier* *Erreur du compilateur*  
 Fichier source introuvable. Vérifiez que le nom du fichier est correctement spécifié et que le fichier figure sur le disque ou dans le répertoire approprié.

**Unable to open makefile** *Erreur de MAKE*  
 Le répertoire en cours ne contient pas le fichier MAKE spécifié (fichier MAKEFILE.MAK par défaut) ou le fichier que vous avez spécifié avec -f.

**Unable to redirect input or output** *Erreur de MAKE*  
 MAKE n'a pas pu ouvrir les fichiers temporaires nécessaires pour rediriger les entrées ou les sorties. Si vous êtes connecté en réseau, vérifiez que vous disposez des droits d'accès sur le répertoire en cours.

**Unable to rename** *nom\_fichier* **to** *nom\_fichier* *Erreur du bibliothécaire*  
 Le bibliothécaire compile une bibliothèque dans un fichier temporaire et réaffecte le nom du fichier de bibliothèque cible à ce fichier temporaire. Ce message s'affiche en cas d'erreur (il indique en général un espace insuffisant sur disque).

**Undefined label** *identificateur* *Erreur du compilateur*  
 L'étiquette indiquée comporte une instruction **goto** mais n'a pas été définie.

**Undefined structure** *identificateur* *Mise en garde du compilateur*  
 La structure indiquée a probablement été spécifiée au niveau d'un pointeur vers une structure. Cette structure n'a cependant pas été définie dans le fichier source. Vérifiez que la structure a effectivement été déclarée et correctement spécifiée.

- Undefined structure** *structure* *Erreur du compilateur*  
 La structure indiquée a été spécifiée sur une ligne précédant l'endroit de l'erreur (probablement au niveau d'un pointeur vers une structure) mais aucune déclaration de cette structure n'a été effectuée. Vérifiez que la structure a effectivement été déclarée et correctement spécifiée.
- Undefined symbol** *identificateur* *Erreur du compilateur*  
 L'identificateur indiqué n'a pas été déclaré. Vérifiez qu'il a été correctement spécifié à cet endroit ou au niveau de la déclaration ou que la déclaration est correcte.
- Undefined symbol** *symbole in module module* *Erreur de l'éditeur de liens*  
 Il est fait référence au symbole dans le module spécifié mais le symbole n'est défini dans aucun des fichiers objet et bibliothèques inclus dans l'édition de liens. Vérifiez que le nom du symbole a été correctement spécifié.
- TLINK génère ce message si des incompatibilités de déclarations d'un symbole de type **pascal** et **cdecl** dans différents fichiers sources ont été détectées ou si vous avez omis le nom d'un fichier .OBJ requis par votre programme. Si vous liez du code C++ avec des modules C, vous avez peut-être oublié de spécifier des déclarations externes C dans `extern "C" { . . . }`. Vérifiez que les deux symboles débutent bien soit par une majuscule soit par une minuscule. Reportez-vous aux options `/C` et `/c`.
- Unexpected }** *Erreur du compilateur*  
 Une accolade fermante a été ajoutée par erreur. Vérifiez s'il ne manque pas une accolade ouvrante ({}).
- Unexpected char X in command line** *Erreur du bibliothécaire*  
 Erreur de syntaxe détectée dans la ligne de commande.
- Unexpected end of file** *Erreur de MAKE*  
 La fin du fichier MAKE a été atteinte alors qu'un fichier inline temporaire n'a pas été fermé.
- Unexpected end of file in comment started on** *num\_ligne* *Erreur de MAKE*  
 La fin d'un fichier source a été atteinte au milieu d'un commentaire. Vérifiez que le commentaire est correctement terminé par `*/`.
- Unexpected end of file in conditional started on line** *num\_ligne* *Erreur du compilateur*  
 La fin du fichier source a été atteinte avant que le compilateur (ou MAKE) ne détecte une instruction `!endif`. Vérifiez que `!endif` a effectivement et correctement spécifiée.
- Union cannot be a base type** *Erreur du compilateur*  
 Une union ne peut être utilisée comme type de base d'une autre classe.
- Union cannot have a base type** *Erreur du compilateur*  
 A union ne peut être dérivée d'une autre classe.
- Union member** *membre is of type class with constructor* *Erreur du compilateur*
- Union member** *membre is of type class with destructor* *Erreur du compilateur*
- Union member** *membre is of type class with operator=* *Erreur du compilateur*  
 Une union ne peut comporter de membres de type **classe** avec des constructeurs, destructeurs ou l'opérateur = définis par l'utilisateur.
- Unions cannot have virtual member functions** *Erreur du compilateur*  
 Une union ne peut avoir un fonction virtuelle comme membre.

**Unknown assembler instruction***Mise en garde du compilateur*

Le compilateur a détecté une instruction en assembleur en ligne.

**Unknown command line switch X ignored***Mise en garde du bibliothécaire*

La ligne de commande ou un fichier de recherche comporte un caractère de barre oblique (/) sans qu'il soit suivi d'une des options autorisées.

**Unknown language, must be C or C++***Erreur du compilateur*

Dans la construction C++

```
extern "nom" type func(/*...*/);
```

"nom" doit correspondre à "C" ou "C++" uniquement. Tout autre nom de langage n'est pas reconnu.

Vous pouvez déclarer une fonction externe en Pascal sans que le compilateur la renomme, comme suit :

```
extern "C" int pascal func(/*...*/);
```

Une fonction C++ (éventuellement surchargée) peut être déclarée Pascal et permettre au compilateur de la renommer (c'est-à-dire de la surcharger), comme suit :

```
extern int pascal func(/*...*/);
```

**Unknown option***Erreur de l'éditeur de liens*

La ligne de commande ou un fichier de recherche comporte un caractère de barre oblique @notct[()], un trait d'union (SYMBOL NOTCT -) ou un caractère d'option DOS sans qu'il soit suivi d'une des options autorisées. Vérifiez que l'option est correctement spécifiée en majuscules ou en minuscules.

**Unknown preprocessor directive: identificateur***Erreur du compilateur*

Le compilateur a détecté un caractère # au début d'une ligne suivi d'un nom de directive non autorisé ou la spécification du reste de la directive est incorrecte.

**Unknown preprocessor statement***Erreur de MAKE*

Un caractère ! a été détecté au début d'une ligne suivi d'un nom d'instruction non autorisé. Vous devez spécifier **error**, **undef**, **if**, **elif**, **include**, **else** ou **endif**.

**Unreachable code***Mise en garde du compilateur*

Une instruction **break**, **continue**, **goto** ou **return** doit être suivie d'une étiquette ou de l'instruction de fin d'une boucle ou d'une fonction. Le compilateur vérifie les boucles **while**, **do** et **for** avec une condition de test constante et tente de reconnaître les boucles qui ne s'interrompent pas.

**Unsupported option chaîne***Erreur de l'éditeur de liens*

Option d'édition de liens incorrecte.

**Unterminated string or character constant***Erreur du compilateur*

Apostrophe manquante à la fin d'une chaîne ou d'un caractère de type constant.

**Use '> >' for nested templates instead of '>>'***Mise en garde du compilateur*

Espace manquant entre deux caractères ">" à la fin d'un nom de template imbriqué. Le compilateur tolère cette erreur très fréquente et la signale comme une mise en garde.

**Use . or -> to call fonction***Erreur du compilateur*

Vous avez essayé d'appeler une fonction membre sans spécifier d'objet.

**Use . or -> to call *membre*, or & to take its address***Erreur du compilateur*

Le compilateur a détecté une référence à un membre de classe non statique sans objet. Ce membre doit être utilisé avec un objet ou son adresse récupérée à l'aide de l'opérateur **&**.

**Use :: to take the address of a member function***Erreur du compilateur*

Si **f** est une fonction membre d'une classe **c**, l'adresse de cette fonction peut être obtenue avec la syntaxe **&c::f**. Notez l'utilisation d'un nom de type classe plutôt que le nom d'un objet et l'opérateur **::** entre le nom de la classe et le nom de la fonction. Un pointeur de fonction membre n'est pas un vrai type de pointeur et ne fait référence à aucune instance particulière d'une classe.

**Use /e with TLINK to obtain debug information from library***Mise en garde du bibliothécaire*

La bibliothèque a été compilée avec un dictionnaire étendu et inclut des informations de débogage. TLINK ne peut extraire des informations de débogage si l'édition de liens inclut un dictionnaire étendu. Par conséquent, pour obtenir ces informations dans un exécutable à partir de cette bibliothèque, vous devez spécifier l'option **/e** pour que l'éditeur de liens ne tienne pas compte du dictionnaire étendu. L'éditeur de liens de l'environnement intégré Borland C++ ne gère pas les dictionnaires étendus, il est donc inutile de reconfigurer l'environnement.

**Use of : and :: dependents for target *cible****Erreur de MAKE*

Vous avez tenté d'utiliser un nom de cible simultanément dans des blocs descripteurs simples et multiples (en utilisant successivement les opérateurs **:** et **::**). Voici un exemple :

```
fichier.a: fichierb
fichier.a:: fichierc
```

**Use qualified name to access nested type *type****Mise en garde du compilateur*

Dans des versions précédentes de C++, les noms de typedef et de modèles de structure déclarés dans des classes étaient directement accessibles à un niveau global. Dans la version actuelle de C++, ces noms doivent être avoir un qualificateur **classe::** comme préfixe s'ils doivent être utilisés en dehors de la portée de leur classe. Pour pouvoir compiler un code écrit selon les spécifications précédentes, Borland C++ permet d'utiliser sans préfixe un tel nom définissant une seule classe mais génère ce message de mise en garde.

**User break***Erreur du compilateur*

Vous avez mis fin explicitement à la compilation ou à l'édition de liens dans l'environnement intégré en appuyant sur **Ctrl+Pause**. Ce message n'est pas une erreur mais une simple confirmation.

**Using based linking for DLLs may cause the DLL to malfunction***Mise en garde de l'éditeur de liens*

Ce message s'affiche si vous utilisez l'option **/B** lors de l'édition de liens d'une bibliothèque DLL. Dans pratiquement tous les cas, cette erreur empêche l'exécution normale de l'application. L'affichage de ce message peut être contrôlé à l'aide de l'option **-w**.

**Value of type void is not allowed***Erreur du compilateur*

Une valeur de type **void** ne correspond en fait à aucune valeur tangible et ne peut donc apparaître dans un contexte où une valeur est requise (par exemple, partie droite d'une affectation, argument d'une fonction, expression de contrôle d'une instruction **if**, **for** ou **while**).

**VIRDEF Name Conflict for *fonction***

Un nom dont la longueur dépasse la limite imposée par l'éditeur de liens est systématiquement tronqué le compilateur. Dans de très rares cas, deux noms tronqués peuvent être exactement identiques. Si ces noms sont des noms de **VIRDEF**, le compilateur génère ce message d'erreur. Pour résoudre ce problème, modifiez le nom de la fonction indiquée.

- Variable *identificateur* is initialized more than once** *Erreur du compilateur*  
Cette variable a été initialisée plusieurs fois. Il est possible de déclarer plusieurs fois une variable de niveau de fichier mais cette variable ne peut être initialisée qu'une seule fois.
- 'virtual' can only be used with member functions** *Erreur du compilateur*  
Une donnée membre a été déclarée avec le spécificateur **virtual** réservé uniquement à des fonctions membres.
- Virtual function *fonction1* conflicts with base class *base*** *Erreur du compilateur*  
Une fonction virtuelle a été déclarée avec les mêmes types d'argument que celle d'une classe de base. Cette déclaration n'est pas autorisée.
- Virtual specified more than once** *Erreur du compilateur*  
En C++, le mot réservé **virtual** ne peut être spécifié qu'une seule fois dans la déclaration d'une fonction membre.
- void & is not a valid type** *Erreur du compilateur*  
Un référence fait systématiquement référence à un objet. Cet objet ne peut pas avoir le type void.
- Void functions may not return a value** *Mise en garde du compilateur*  
La fonction en cours est déclarée **void** mais le compilateur a détecté une instruction return avec une valeur. Cette valeur n'est pas prise en compte.
- fonction* was previously declared with the language *langage*** *Erreur du compilateur*  
La déclaration **extern** de la fonction indiquée autorise l'utilisation d'un seul langage. Cette fonction a été déclarée avec plusieurs langages à différents endroits du même module.
- While statement missing (** *Erreur du compilateur*  
Parenthèse gauche manquante après le mot clé **while** dans une boucle de répétition **while**.
- While statement missing )** *Erreur du compilateur*  
Parenthèse droite manquante après l'expression test dans une boucle de répétition **while**.  
Ce message s'affiche si TLINK n'a pas pu écrire toutes les données en raison d'un disque probablement saturé.
- Write error on file *nom\_fichier*** *Erreur de MAKE*  
MAKE n'a pas pu accéder en lecture ou en écriture au fichier spécifié dans le fichier MAKE. Vérifiez que le disque n'est pas saturé et que vous y avez accès en écriture.
- Wrong number of arguments in call of macro *macro*** *Erreur du compilateur*  
La macro-instruction indiquée a été appelée dans le fichier source avec un nombre d'arguments incorrects.

# Contrôles personnalisés Windows de Borland

Avant de lire cette annexe, familiarisez-vous avec les commandes des boîtes de dialogue et l'éditeur Dialog. Vous trouverez au chapitre 15 des renseignements sur ces questions.

La bibliothèque de contrôles personnalisés Windows de Borland (BWCC) contient une classe de dialogues personnalisés et un ensemble de contrôles personnalisés (boutons, cases à cocher, groupes ombrés, etc.). Ces contrôles renforcent l'impact visuel de vos boîtes de dialogue et en optimisent les fonctions.

Deux des fichiers en ligne fournis avec Resource Workshop donnent des renseignements complémentaires sur les contrôles personnalisés Windows de Borland :

- BWCCAPI.RW donne des renseignements techniques sur l'interface du programme d'application des contrôles personnalisés Windows de Borland.
- BWCCSTYL.RW propose quelques styles pour créer des boîtes de dialogue de style Borland.

## Utilisation de la classe Borland de dialogues personnalisés

---

La classe de dialogues personnalisés BORDLG a une action à la fois visuelle et fonctionnelle :

- Elle améliore l'apparence de votre fenêtre de dialogue en peignant le fond avec un pinceau qui varie en fonction de l'écran de l'application. Pour les écrans VGA et de résolution supérieure, le fond est une fine grille de lignes blanches perpendiculaires produisant un effet de métal ciselé. Pour les écrans EGA et monochromes, le fond est blanc.

- Elle optimise le traçage des boîtes de dialogue en appelant directement les routines de dessin des contrôles personnalisés, au lieu d'attendre que Windows peigne les contrôles, ce qui accélère la réalisation des boîtes de dialogue, autrement assez laborieuse.

Pour utiliser la classe de dialogues personnalisés,

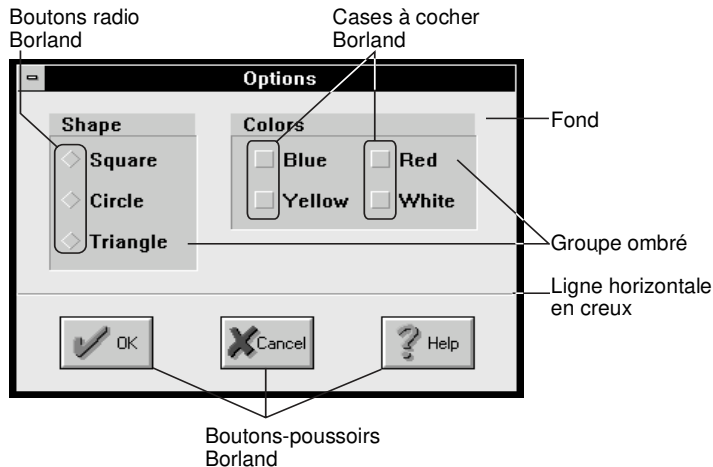
1. Ouvrez la ressource de dialogue que vous voulez convertir.
2. Double-cliquez sur la barre de titre du dialogue pour afficher la boîte de dialogue Window Style.
3. Indiquez "bordlg" comme classe et cliquez sur OK.

## Utilisation des contrôles Borland

Les contrôles Borland ajoutent à vos boîtes de dialogue un effet de relief qui renforce leur impact visuel. Pour l'utilisateur final, elles fonctionnent comme les contrôles Windows standard, mais en intégrant plusieurs améliorations techniques (décrites plus loin).

La figure ci-dessous montre une boîte de dialogue convertie en BWCC qui utilise plusieurs contrôles Borland.

Figure B.1  
Boîte de dialogue  
avec contrôles  
Borland





Vous trouverez ci-après un rapide descriptif de chaque commande, accompagné d'une icône de l'outil correspondant. Comme pour les contrôles Windows standard, vous pouvez insérer des contrôles Borland dans vos dialogues en allant les chercher dans la palette d'outils de l'éditeur Dialog.

La classe de chaque contrôle est précisée dans le descriptif. Pour voir la classe et les autres paramètres de l'un quelconque de ces contrôles, affichez la boîte de dialogue Generic Control Style en appuyant sur la touche *Ctrl* et en double-cliquant simultanément sur le contrôle.



**Groupe ombré** Boîte rectangulaire ombrée qui regroupe visuellement d'autres contrôles. Peut apparaître comme gravé dans la boîte de dialogue ou en relief. Appartient à la classe *BorShade*.



**Ligne horizontale en creux** Ligne horizontale divisant la boîte de dialogue et donnant l'impression d'être gravée dans la boîte de dialogue. (Peut être transformée en ligne "en relief"). Appartient à la classe *BorShade*.



**Ligne verticale en creux** Même chose que la Ligne horizontale en creux, mais dans le sens vertical. Appartient à la classe *BorShade*.



**Bouton-poussoir Borland** Famille de boutons-poussoirs avec des symboles ayant un grand impact visuel, plus une option "owner-draw". Les boutons-poussoirs Borland sont plus grands que la plupart des boutons-poussoirs Windows standard. Ils appartiennent à la classe *BorBtn*.



**Bouton radio Borland** Bouton en relief en forme de losange. Quand on clique sur ce bouton, un losange noir apparaît au centre et l'ombrage s'inverse, donnant l'impression que le bouton a été enfoncé. Il y a également une option de "owner-draw". Appartient à la classe *BorRadio*.



**Case à cocher Borland** Case à cocher en relief affichant une coche au lieu d'un "X." Il y a également une option Owner-draw. Appartient à la classe *BorCheck*.



**Texte statique Borland** Chaîne de texte fixe utilisée principalement pour étiqueter certaines parties de la boîte de dialogue. Appartient à la classe *BorStatic*.

---

**Perfectionnements  
des boutons et  
cases à cocher**

Les boutons-poussoirs, boutons radio et cases à cocher Borland ont, sur les commandes Windows classiques, les avantages fonctionnels suivants :

- Un niveau supplémentaire de notification aux fenêtres mères et de contrôle de la fenêtre de premier plan Windows et des tabulations. Si vous sélectionnez l'option Parent Notify dans la boîte de dialogue de style du contrôle, celui-ci envoie les messages appropriés, sélectionnés dans la liste suivante, au moment de l'exécution :
  - BBN\_SETFOCUS indique à la fenêtre mère que le bouton-poussoir, le bouton radio ou la case à cocher est la cible de toutes les entrées clavier autrement que par cliquage de la souris.
  - BBN\_SETFOCUSMOUSE indique à la fenêtre mère que le bouton-poussoir, le bouton radio ou la case à cocher est la cible de toutes les entrées clavier par cliquage de la souris.
  - BBN\_GOTATAB indique à la fenêtre mère que l'utilisateur a appuyé sur la touche Tab alors que le bouton-poussoir, le bouton radio ou la case à cocher est la cible de toutes les entrées clavier. La fenêtre mère peut intervenir dans le traitement de la frappe en retournant une valeur différente de zéro.
  - BBN\_GOTABTAB indique à la fenêtre mère que l'utilisateur a appuyé sur *Maj+Tab* (espace arrière) alors que le bouton-poussoir, le bouton radio ou la case à cocher est la cible de toutes les entrées clavier. La fenêtre mère peut intervenir dans le traitement de la frappe en retournant une valeur différente de zéro.
- Une option "owner-draw" qui permet à la fenêtre mère de dessiner le bouton-poussoir, le bouton radio ou la case à cocher. Dans la mesure où c'est votre application qui se charge du dessin, ils ne ressembleront pas nécessairement à des commandes Borland, mais auront toutefois le comportement type de leur classe de commande.

---

**Utilisation des  
boîtes de dialogue  
de style BWCC**

Quatre boîtes de dialogue permettent de définir le style des contrôles BWCC :

- Borland Button Style
- Borland Radio Button Style
- Borland Check Box Style
- Borland Shade Style

Pour afficher l'une de ces boîtes de dialogue, double-cliquez sur le contrôle dont vous voulez définir le style.

Chacun possède une fenêtre permettant de saisir une légende et un ID de contrôle. Les boîtes de dialogue Button style, Radio button style et Check box style ont des options d'attributs (Attributes Options) pour Tab Stop, Disabled, Group, Visible, et Border, ainsi que Parent Notify et Owner Draw (décrits plus haut).

Les quatre sections qui suivent décrivent les fonctions spécifiques à chaque boîte de dialogue de style.

**Boîte de dialogue  
Borland Button  
Style**

Cette boîte de dialogue permet de choisir parmi les trois types de bouton : Pushbutton, Defpushbutton et Bitmap.

**Pushbutton et Defpushbutton**

Pushbutton est l'option par défaut. Ce type de bouton est entouré d'une bordure en gras pour indiquer à l'utilisateur qu'il s'agit du *bouton par défaut*, exécuté quand l'utilisateur appuie sur la touche Entrée. (La seule exception est quand la fenêtre de premier plan Windows est dans une commande Edit Text pour laquelle l'indicateur Want Return a été activé. Reportez-vous au tableau 15.11 pour une description de l'indicateur Want Return.)

La première fois que vous insérez un bouton Borland dans une boîte de dialogue, son texte est Button et il prend le premier ID disponible dans l'ordre. Pour remplacer le bouton par l'un des boutons Borland standard, remplacez l'ID de contrôle par l'une des valeurs prédéfinies du tableau suivant :

Tableau B.1  
Boutons de  
contrôles BWCC  
prédéfinis

| Nom de l'ID | Valeur de l'ID | Type       | Représentation                        |
|-------------|----------------|------------|---------------------------------------|
| IDOK        | 1              | OK         | Coche verte                           |
| IDCANCEL    | 2              | Annuler    | X rouge                               |
| IDABORT     | 3              | Abandonner | Touche SOS                            |
| IDRETRY     | 4              | Réessayer  | Machine à sous                        |
| IDIGNORE    | 5              | Ignorer    | Panneau de limite de vitesse à 55 mph |
| IDYES       | 6              | Oui        | Coche verte                           |
| IDNO        | 7              | Non        | Cercle rouge barré                    |
| IDHELP      | 998            | Aide       | Point d'interrogation bleu            |

## Bitmap

L'option Bitmap vous permet d'insérer une image bitmap (appelée par son ID de contrôle) dans le bouton. Pour lire dans une image bitmap :

1. Utilisez le contrôle Bouton pour insérer le bouton générique BWCC dans votre boîte de dialogue. Notez son ID de contrôle.
  2. Appelez l'éditeur Bitmap et créez une image bitmap. (Voir page 234 pour plus d'informations sur la création de bitmaps.)
  3. Dans l'éditeur Bitmap, sélectionnez Resource | Rename pour afficher la boîte de dialogue Rename Resource puis exécutez l'une des deux actions suivantes :
    - Dans la boîte de texte New Name entrez un entier égal à l'ID de contrôle du bouton auquel vous aurez ajouté la valeur de décalage indiquée dans le tableau B.2.
    - Renommez le bitmap et affectez-lui un identificateur dont la valeur sera égale à l'ID de contrôle du bouton auquel vous ajouterez le décalage indiqué dans le tableau B.2. (La création des identificateurs est expliquée au chapitre 14.)
1. Fermez l'éditeur Bitmap.
  2. Revenez à l'éditeur Dialog. Si le bitmap n'apparaît pas immédiatement à l'intérieur du bouton BWCC, redimensionnez le bouton. Le bitmap doit alors apparaître.

Le bitmap ne s'affiche pas dans l'éditeur Dialog tant que l'éditeur Bitmap n'est pas fermé.

Tableau B.2  
Décalages Bitmap

| Etat du bouton | Décalage pour<br>VGA et résolutions<br>supérieures | Décalage pour<br>EGA/monochrome |
|----------------|----------------------------------------------------|---------------------------------|
| Standard       | 1000                                               | 2000                            |
| Appuyé         | 3000                                               | 4000                            |
| Fenêtre active | 5000                                               | 6000                            |

Par exemple, pour afficher le bitmap de fenêtre active d'un bouton dont l'ID de contrôle est 276, entrez 5276 pour un écran VGA, 6276 pour un écran EGA.

---

**Boîte de dialogue  
Borland Radio  
Button Style**

Cette boîte de dialogue propose deux styles de bouton :

- *Bouton radio*. La mise en surbrillance et la désélection ne sont pas automatiques. L'application doit appeler la fonction **CheckRadioButton** pour envoyer un message BM\_SETCHECK qui mettra en surbrillance le bouton sélectionné et désélectionnera les autres boutons.
- *Auto radio button*. BWCC et Windows s'associent pour prendre en charge la mise en surbrillance du bouton sélectionné et la désélection des autres boutons. C'est l'option par défaut.

---

**Boîte de dialogue  
Borland Check Box  
Style**

Cette boîte de dialogue propose quatre styles de cases à cocher :

- *Check box*. La case n'est pas cochée automatiquement. L'application doit appeler la fonction **CheckDlgButton** pour envoyer un message BM\_SETCHECK qui cochera la case sélectionnée.
- *Auto check box*. BWCC et Windows s'associent pour cocher la case sélectionnée. C'est l'option par défaut.
- *3-state*. La case n'est pas cochée automatiquement. L'application doit appeler la fonction **CheckDlgButton** pour envoyer un message BM\_SETCHECK qui cochera la case sélectionnée.

Les trois états du bouton sont actif, inactif, et "indéterminé," représenté par un motif à damier. L'application détermine ce qui est sous-entendu par "indéterminé".

- *Auto 3-state*. BWCC et Windows s'associent pour cocher la case sélectionnée.

---

**Boîte de dialogue  
Borland Shade  
Style**

Cette boîte de dialogue vous propose trois outils pour définir le style des contrôles que vous avez ajoutés ; ces outils sont : groupe ombré, ligne horizontale en creux et ligne verticale en creux. Les boutons radio de Shade Style vous permettent d'effectuer les conversions suivantes :

- *Group Shade to Raised Shade*. Les groupes ombrés et les ombrages en relief servent à regrouper ensemble des contrôles dont les fonctions sont liées – par exemple des boutons radio et des cases à cocher. Les groupes ombrés semblent gravés dans la surface de la boîte de dialogue, les ombrages en relief apparaissent en relief par rapport à cette surface.
- *Horizontal Dip to Horizontal Bump, Vertical Dip to Vertical Bump*. Les lignes en creux ont un rôle de séparateurs dans les boîtes de dialogue ou les ombrages en relief ; les lignes en relief ont un rôle de séparateurs dans les boîtes à fond grisé gravé.

---

**Boîte de dialogue  
Borland Static Text  
Style**

Cette boîte de dialogue permet de saisir du texte, de définir les attributs et de choisir le style du texte statique Borland.

Outre les attributs standard (Disabled, Group et Visible), le texte statique possède deux attributs supplémentaires :

- Quand l'option Border est cochée, le texte statique est entouré d'un cadre Windows standard qui utilise la couleur active de la fenêtre principale de Windows (voir l'Écran de contrôle Windows) ;
- Quand l'option No Underline est cochée, un signe perluète (&) apparaît comme caractère constant, au lieu de souligner le caractère suivant de la légende.

Les options de Control Style sont décrites dans le tableau 15.12. Il y a cependant une différence majeure : *tout* texte statique BWCC, y compris Simple Text, utilise le fond standard BWCC gris.

---

## Modifier des applications existantes avec BWCC

---

Resource Workshop vous permet de modifier des applications Windows existantes en introduisant des contrôles personnalisés de style Borland (boutons 3D, boîtes de dialogue à aspect "métal ciselé", etc.). Cette procédure s'effectue en deux temps :

Les paragraphes qui suivent décrivent ces opérations plus en détail.

1. Modifiez votre fichier WIN.INI pour charger la bibliothèque Borland Windows Custom Control (BWCC) chaque fois que vous lancez Windows.
2. Modifiez l'application dans Resource Workshop pour changer de fonctions d'interfaces utilisateur, par exemple les boîtes de dialogue, les menus, les icônes, etc.

---

**Chargement de  
BWCC pour activer  
les contrôles  
Borland**

La bibliothèque BWCC, qui permet la prise en charge des contrôles personnalisés de style Borland, doit être chargée pour qu'une application puisse utiliser les fonctions BWCC.

Modifiez le fichier WIN.INI (situé dans le répertoire principal de Windows) pour que Windows charge le fichier LOADBWCC.EXE dans la mémoire au démarrage. (Le programme d'installation met LOADBWCC.EXE dans le répertoire des fichiers exécutables du compilateur et ajoute ce répertoire dans votre CHEMIN.)

Ajoutez LOADBWCC.EXE au début de la liste de fichiers qui apparaît à la suite de l'instruction "LOAD=". Il doit d'abord apparaître dans l'instruction, afin que BWCC soit chargé en mémoire avant d'exécuter toute application modifiée.

Par exemple, si l'instruction LOAD de votre fichier WIN.INI est :

```
LOAD=NWPOPOPUP.EXE AD.EXE
```

vous devez modifier l'instruction comme suit :

```
LOAD=loadbwcc.exe NWPOPOPUP.EXE AD.EXE
```

---

### Utilisation de BWCC dans des programmes C et C++

Si vous utilisez l'EDI Borland C++, cochez BWCC quand vous créez une application (voir le chapitre 2 pour toute information sur les projets de l'EDI). Vous devez aussi ajouter `BWCCGetVersion()` ; dans `WinMain`.

Si vous n'utilisez pas l'EDI, la procédure est la suivante :

- Ajoutez une **#include** pour BWCC.H dans votre fichier .C ou .CPP.
- Ajoutez BWCC.LIB dans votre projet EDI C ou C++, ou insérez-le dans le champ TLINK Library avant IMPORT.LIB.
- Ajoutez `BWCCGetVersion()` ; dans `WinMain`.

---

### Astuces sur les modifications de ressources

Cette section évoque ce que vous ne devez pas perdre de vue quand vous modifiez les ressources d'applications existantes.

---

### Accélérateurs

Si vous ajoutez un accélérateur, assurez-vous qu'il retourne le même ID que la commande de menu correspondante. Sinon, soit l'accélérateur exécutera une mauvaise commande, soit il ne fera rien.

---

### Bitmaps, curseurs et icônes

Vous pouvez modifier les bitmaps, curseurs et icônes existants. N'effacez pas de bitmaps, de curseurs ni d'icônes, n'essayez pas d'en ajouter. Dans la plupart des cas, l'application ne sera pas capable de les utiliser.

---

### Boîtes de dialogue

Vous pouvez déplacer des éléments à l'intérieur d'une boîte de dialogue et convertir les contrôles en contrôles personnalisés Borland équivalents. Lors de la modification, veillez à ne pas changer le type de contrôle associé à chaque ID de contrôle. Par exemple, si l'ID de contrôle 100 est une case à cocher, ne le transformez pas en bouton radio, car l'application continuera à le considérer comme une case à cocher.

Dans la plupart des cas, vous pouvez supprimer des contrôles qui ne sont pas directement liés aux fonctions de l'application. Vous pouvez par exemple supprimer une légende, un texte statique qui n'a aucun effet sur le fonctionnement de l'application. Ne supprimez pas de contrôle d'édition : cela *aura* une incidence sur le fonctionnement de l'application.

N'ajoutez pas de nouveaux contrôles ; l'application ne sera pas capable de les utiliser.

---

**Menus**

Avec la plupart des applications vous pouvez sans risque déplacer des commandes à l'intérieur d'un menu. Mais ne déplacez pas des commandes d'un menu à un autre. (Par exemple ne déplacez pas la commande Open du menu File au menu Edit.) Sinon, l'application risque de ne pas être capable d'afficher l'aide contextuelle, ni de cocher ou décocher les commandes des menus. Ne changez jamais l'ordre des menus dans la barre des menus. Par exemple, si File est le premier menu, ne le mettez pas en deuxième place.

---

**Tables de chaînes**

Faites attention quand vous modifiez des tables de chaînes existantes. Certains programmes chargent les chaînes dans des tampons de taille fixe et le fait d'ajouter du texte à une chaîne existante peut entraîner un dépassement de la capacité du tampon. N'ajoutez pas de nouvelles chaînes : l'application ne sera pas capable de les utiliser.



# En-têtes précompilés

Borland C++ peut générer et par conséquent utiliser des en-têtes précompilés pour vos projets. Les en-têtes précompilés permettent d'accélérer considérablement la procédure de compilation.

## Principe

---

Pour compiler des programmes C et C++ volumineux, le compilateur peut passer la moitié de son temps à analyser la syntaxe des fichiers d'en-têtes. Pendant cette opération, il insère des déclarations et des définitions dans sa table de symboles. Or, si dix de vos fichiers source ont le même fichier d'en-têtes, le compilateur l'analysera dix fois en produisant à chaque fois la même table de symboles.

Les fichiers d'en-têtes précompilés permettent un gain de temps considérable. Au cours de la compilation, BCC.EXE mémorise une image de la table des symboles dans un fichier intitulé par défaut BCDEF.CSM (BC32DEF.CSM pour BCC32.EXE). BCDEF.CSM est enregistré dans le même répertoire que le compilateur. Par la suite, quand le même fichier source (ou un autre fichier source contenant les mêmes fichiers d'en-têtes) est recompilé, le compilateur recharge BCDEF.CSM à partir du disque au lieu de recommencer l'analyse de syntaxe de tous les fichiers d'en-têtes. Le rechargement direct de la table des symboles à partir du disque est dix fois plus rapide que l'analyse de la syntaxe du texte des fichiers d'en-têtes.

Les en-têtes précompilés ne sont utilisés que si la seconde compilation utilise un ou plusieurs fichiers d'en-têtes, options de compilateur, macros, etc. de la première compilation.

Si, en cours de compilation d'un fichier source, Borland C++ s'aperçoit que les premiers **#include** sont identiques à ceux d'une compilation antérieure (de la même source ou d'une source différente), il charge l'image binaire de ces **#include** et analyse la syntaxe des **#include** restants.

Pour un module donné, soit la totalité des en-têtes précompilés est utilisée, soit aucun : si la compilation d'un fichier d'en-tête quelconque échoue, le fichier d'en-tête précompilé n'est pas mis à jour pour ce module.

---

### Inconvénients

Quand vous utilisez des en-têtes précompilés, BCDEF.CSM peut devenir très volumineux du fait qu'il stocke les images des tables de symboles de tous les ensembles de directives **#include** rencontrés dans vos sources. Si vous n'avez pas assez d'espace sur votre disque, un message vous prévient que l'écriture a échoué à cause des en-têtes précompilés. Pour régler ce problème, faites de la place sur votre disque et relancez la compilation. Pour de plus amples renseignements sur la manière de réduire la taille du fichier BCDEF.CSM, reportez-vous à la section "Optimisation des en-têtes précompilés", plus haut dans cette annexe.

Si dans un fichier Make vous utilisez des macros importantes en plus des en-têtes précompilés, la taille des macros est limitée : 4 ko pour les applications 16 bits et 16 ko pour les applications 32 bits.

Si un en-tête comporte un code quelconque, il ne peut pas être compilé. Ainsi, bien qu'il puisse y avoir des définitions de classes C++ dans des fichiers d'en-têtes, assurez-vous que seules des fonctions membres en ligne sont définies dans l'en-tête et qu'elles tiennent compte de mises en garde du type "Functions containing for are not expanded inline".

---

## Comment utiliser des en-têtes précompilés

---

Vous pouvez commander l'utilisation des en-têtes précompilés de différentes manières :

- A partir de l'EDI, grâce à la boîte de dialogue des Project Options. L'EDI crée le nom du fichier d'en-têtes précompilé à partir du nom du projet, soit *PROJECT.CSM*
- A partir de la ligne de commande, à l'aide des options **-H**, **-H=nom\_fichier**, **-Hc**, **-H"nom\_fichier"** et **-Hu** (voir la section "Compiler | Precompiled headers" du chapitre 3)
- A partir de votre code, à l'aide des directives pragma **hdrfile** et **hdrstop** (voir le chapitre 5 du *Guide du programmeur*)

---

## Création des noms de fichiers

Le compilateur utilise un seul fichier pour stocker toutes les en-têtes précompilés. Le nom par défaut de ce fichier est BCDEF.CSM. Mais vous pouvez lui donner le nom de votre choix à l'aide de l'option de ligne de commande **-H=nom\_fichier** ou de la directive #pragma **hdrfile** .

Attention !

Si vous avez l'impression que votre fichier .CSM est plus petit qu'il ne devrait, il se peut que le compilateur ait manqué d'espace disque pour enregistrer les en-têtes dans le fichier .CSM. Dans ce cas, le compilateur efface le fichier .CSM pour faire de la place au fichier .OBJ, puis crée un nouveau fichier .CSM (par conséquent moins important). Si vous vous trouvez dans cette situation, faites de la place sur le disque avant de procéder à la compilation.

---

## Contrôle d'identité

Pour qu'un en-tête précompilé préalablement généré puisse être chargé pour une autre compilation, les conditions suivantes doivent être remplies.

Le deuxième ou le dernier fichier source doit :

- Avoir le même groupe de fichiers à inclure, classés dans le même ordre
- Avoir les mêmes macros définies pour des valeurs identiques
- Utiliser le même langage (C ou C++)
- Utiliser des fichiers d'en-têtes de même date ; ils pourront être inclus soit directement, soit indirectement

Vous devrez aussi, lors de la compilation du nouveau fichier source, conserver les mêmes valeurs pour les options énumérées ci-après (par exemple, si vous avez compilé le premier fichier avec le modèle SMALL, vous devez compiler le second fichier avec ce même modèle) :

- Modèle mémoire, y compris SS != DS (**-m $x$** )
- Préfixe de soulignement pour les extréma (**-u**) "
- Taille maximum des identificateurs (**-iL**)
- Destination DOS (par défaut) ou Windows (**-W** ou **-W $x$** )
- Code de recouvrement compatible DOS (**-Y**)
- Commande des tables virtuelles (**-V $x$**  et **-Vm $x$** )
- Expansion en ligne des fonctions intrinsèques (**-Oi**)
- Modèles (**-J $x$** )
- Constantes de type chaîne dans le segment code (**-dc**, 16 bits seulement)
- Informations de débogage (**-v**, **-vi**, et **-R**)
- Variables far (**-F $x$** )

- Génération de l'alignement sur mot (-a)
- Conventions d'appel Pascal (-p)
- Traitement des **enum** comme des entiers (-b)
- **char** non signés par défaut (-K)
- Conformité du langage (-A)
- Compilation C++ (-P)

---

### Optimisation des en-têtes précompilés

Pour une compilation optimale avec des en-têtes précompilés, observez les règles suivantes :

- Enregistrez vos fichiers d'en-têtes dans le même ordre dans tous vos fichiers source.
- Mettez en premier les plus gros fichiers d'en-têtes.
- Dans BCDEF.CSM, mettez en premier les séquences de fichiers d'en-têtes les plus souvent utilisées.
- Utilisez `#pragma hdrstop` pour délimiter la liste des fichiers d'en-têtes au bon endroit. Vous aurez ainsi dans les différentes sources une liste de fichiers d'en-têtes similaire à celle du compilateur. `#pragma hdrstop` est décrite plus en détail dans le chapitre 5 du *Guide du programmeur*.

Par exemple, dans le cas de deux fichiers source ASOURCE.C et BSOURCE.C, contenant tous deux windows.h et myhdr.h,

```
ASOURCE.C #include <windows.h>
 #include "myhdr.h"
 #include "xxx.h"
 <...>
```

```
BSOURCE.C #include "zz.h"
 #include <string.h>
 #include "myhdr.h"
 #include <windows.h>
 <...>
```

il est conseillé de réorganiser le début de BSOURCE.C de la manière suivante :

```
BSOURCE.C modifié #include <windows.h>
 #include "myhdr.h"
 #include "zz.h"
 #include <string.h>
 <...>
```

Notez que `windows.h` et `myhdr.h` sont dans le même ordre dans `BSOURCE.C` et dans `ASOURCE.C`. Vous pouvez aussi créer une nouvelle source intitulée `PREFIX.C` contenant uniquement les fichiers d'en-têtes, à savoir :

`PREFIX.C`

```
#include <windows.h>
#include "myhdr.h"
```

Si vous compilez `PREFIX.C` en premier (ou si vous insérez une directive `#pragma hdrstop` dans `ASOURCE.C` et dans `BSOURCE.C` à la suite de la directive `#include "myhdr.h"`) après la compilation initiale de `PREFIX.C`, `ASOURCE.C` et `BSOURCE.C` seront en mesure de charger la table de symbole produite par `PREFIX.C`. Le compilateur n'aura plus qu'à analyser la syntaxe de `xxx.h` pour `ASOURCE.C`, celle de `zz.h` et de `string.h` pour `BSOURCE.C`.



# Utilisation d'EasyWin

EasyWin est une fonction de Borland C++ qui vous permet de compiler des applications DOS standard utilisant les entrées et sorties de type TTY traditionnelles pour les faire tourner comme de véritables programmes Windows. Avec EasyWin, vous n'avez aucune modification à apporter aux programmes DOS pour leur permettre de tourner sous Windows.

## Convertir des applications DOS en applications Windows : un jeu d'enfant !

---

Pour convertir celles de vos applications machines qui utilisent des fichiers ou fonctions `iostream` standard, vérifiez la valeur de EasyWin Target Type dans TargetExpert dans EDI. Si vous travaillez avec le compilateur de ligne de commandes, utilisez le commutateur **-W**. Borland C++ note que votre programme ne contient pas la fonction `WinMain` (en principe nécessaire pour les applications Windows) et édite un lien avec la bibliothèque d'EasyWin. Quand vous faites tourner votre programme sous Windows, une fenêtre standard est créée et votre programme reçoit des entrées et produit des sorties dans cette fenêtre exactement comme s'il s'agissait de l'écran standard.

Exemple de programme en C :

```
#include <stdio.h>
main()
{
 printf("Hello, world\n");
 return 0;
}
```

Exemple de programme C++ :

```
#include <iostream.h>
main()
{
 cout << "Hello, world\n";
 return 0;
}
```

La fenêtre d'EasyWin peut être utilisée chaque fois qu'un périphérique TTY demande une entrée ou une sortie. Cela signifie que `stderr`, `stdaux`, et `cerr` sont tous, au même titre que `stdin` et `stdout`, connectés à cette fenêtre.

## **InitEasyWin( )**

---

L'objet d'EasyWin est de vous permettre de convertir rapidement et facilement des applications DOS en programmes Windows. Mais vous pouvez de temps à autre avoir besoin d'utiliser EasyWin à partir d'un programme Windows proprement dit, par exemple ajouter des fonctions *printf* au code de votre programme pour déboguer votre programme Windows.

Pour utiliser EasyWin à partir d'un programme Windows, appelez la fonction *\_InitEasyWin* avant d'effectuer toute entrée ou sortie standard.

Par exemple :

```
#include <windows.h>
#include <stdio.h>

#pragma argsused
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
 LPSTR lpszCmdLine, int cmdShow)
{
 _InitEasyWin();
 /* Normal windows setup */
 printf("Hello, world\n");
 return 0;
}
```

Vous trouverez le prototype d'*\_InitEasyWin* dans `stdio.h` et `iostream.h`.



---

## Fonctions supplémentaires

Reportez-vous au paragraphe LR, chapitre RTL, pour la description des fonctions disponibles avec les programmes EasyWin.

Pour plus de commodité, EasyWin possède cinq fonctions supplémentaires qui vous permettent d'indiquer les coordonnées X et Y de la fenêtre pour les entrées et les sorties, d'effacer la fenêtre ou d'effacer la fin de la ligne en cours :

|               |           |
|---------------|-----------|
| <i>clreol</i> | (conio.h) |
| <i>clrscr</i> | (conio.h) |
| <i>gotoxy</i> | (conio.h) |
| <i>wherex</i> | (conio.h) |
| <i>wherey</i> | (conio.h) |

Ces fonctions ont le même nom (et rôle) que les fonctions du fichier d'en-tête conio.h. Les classes de `constrea.h` offrent des fonctions conio pour les flux C++. Reportez-vous au chapitre 6 du *Guide du programmeur* pour une présentation complète de `constreams` et `iostreams`.

Les programmes suivants peuvent être adaptés aux programmes d'EasyWin mais ne sont pas disponibles au format Windows 16 bits. Ils sont fournis pour faciliter le portage du code existant dans une application Windows 16 bits.

|                 |           |
|-----------------|-----------|
| <i>fgetchar</i> | (stdio.h) |
| <i>getch</i>    | (stdio.h) |
| <i>getchar</i>  | (stdio.h) |
| <i>getche</i>   | (stdio.h) |
| <i>gets</i>     | (stdio.h) |
| <i>kbhit</i>    | (conio.h) |
| <i>perror</i>   | (errno.h) |
| <i>printf</i>   | (stdio.h) |
| <i>putch</i>    | (conio.h) |
| <i>putchar</i>  | (stdio.h) |
| <i>puts</i>     | (stdio.h) |
| <i>scanf</i>    | (stdio.h) |
| <i>vprintf</i>  | (stdio.h) |
| <i>vscanf</i>   | (stdio.h) |



# Index

"Mémoire insuffisante" (erreur)  
  et outils 32 bits 21  
#defines 253  
  afficher 242  
\$\* (macro MAKE)  
  compatibilité avec NMAKE 211  
\$d (macro MAKE)  
  compatibilité avec NMAKE 212  
&, dans le texte d'une commande menu 314  
\* (symbole d'action TLIB) 206  
+, sujets AppExpert 97  
+ (symbole d'action TLIB) 206  
-, sujets AppExpert 97  
- (symbole d'action TLIB) 206  
-\* et \*- (symboles d'action TLIB) 206  
+ et +- (symboles d'action TLIB) 206  
-? (option MAKE) 210  
-d (option MAKE) 210  
-m (option MAKE) 210  
.HLP (fichiers)  
  création 96  
/0, option TLIB (enregistrement de commentaire)  
205  
-2 (option BCC)  
  instructions 80286 68  
-3 (option BCC)  
  instructions 80386 68, 76  
32 bits, ressources dans Resource Workshop 261  
-4 (option BCC)  
  instructions 80486 68, 76  
-5 (option BCC)  
  instructions Pentium 76  
~, symbole de fichier de sauvegarde 259  
\a, caractère d'alignement à droite dans les menus  
315  
\t, caractère de tabulation dans les menus 315

## A

-A (option BCC.EXE) 175 - 9-176  
  alignement d'entier 68, 76

  mots-clés ANSI 66  
Absolute (options de grille) 277  
Absolute Align (style), boîtes de dialogue 267  
ACBP (champ) 179  
accélérateurs  
  astuces et réserves 477  
Accelerator (éditeur) 323  
  boîte texte Command 329  
  lancer 325  
  mode Key Value 327  
  mode Manual 330  
  utiliser avec l'éditeur Menu 326  
  volet Attribute 326  
  volet Outline 326  
activer les commandes de menu 313  
Add to Project (commande) 245  
adresse de base 176  
Advanced (bouton)  
  TargetExpert et 28  
Aérographe 350  
  forme de pinceau, choisir 363  
  options de couleur 350  
  motifs 363  
affichage  
  des déclarations de symboles 18  
  des détails d'un objet 18  
  des avertissements, options 119  
affichage (options), dans l'éditeur Dialog 300  
Aide  
  activation 10  
  scripts ressource 244  
Airbrush Shape (commande) 363  
ajouter (symbole d'action TLIB) 206  
ajouter  
  des caractères, ressource police 390  
  des identificateurs 254  
  un libellé 266  
  un suivi à la fenêtre Watch 127  
Align Controls (boîte de dialogue) 282  
alignement (options), boîte de dialogue Edit Text  
Style 293 - 15-294

- alignement à droite, dans les menus 315
- alignement de mot et d'entier 68, 76
- aligner
  - contrôles 277
  - zone sélectionnée, éditeur Bitmap 365
- annuler 258, 287
- ANSI
  - code compatible 66
  - conformité de langage (Borland C++, spécifique à l'implémentation), option 66
  - règles de conversion de calculs flottants 64
- AppExpert
  - + et - avec les sujets 97
  - classes, ajout 109
  - classes, visualisation 107
  - description 95
  - fichiers créés 96
  - gestionnaires, ajout 112
  - gestionnaires, suppression 112
  - et ObjectWindows 95
  - variables d'instance
    - ajout 113
    - suppression 113
- AppExpert Application Generation Options, boîte de dialogue 97
- AppExpert, Resource Workshop et 253
- applications
  - création avec AppExpert 95 - 4-96
  - par défaut dans AppExpert 98
  - Microsoft Windows
    - segments de code 179
- arborescence AppExpert
  - description 97
- arborescence de fenêtres 143
- arguments
  - transmettre au programme à déboguer 121
  - visualiser les arguments transmis 125
- arguments de classes
  - transmis par valeur 78
- arrêter un programme 126
- ASCII (touches de raccourci) 324
- .ASM (fichiers)
  - BCC 48
- assembleur
  - déboguer 124
  - par défaut 91
- AT (option BCC.EXE)
  - mots-clés Borland C++ 66
- attribut big 179
- Attribute (volet)
  - éditeur Menu 309
  - éditeur Accelerator) 326
- attributs 179
  - mise en surbrillance de la syntaxe et 15
- AU (option BCC.EXE)
  - conformité UNIX 66
- Auto 3-state (contrôle bouton 288
- Auto check box (BWCC) 475
- Auto Check Box (contrôle) 287
- Auto radio button (BWCC) 475
- Auto Radio Button (contrôle) 288
- AUTOEXEC.BAT
  - et la mémoire 21
- Automatic Scroll (options), boîte de dialogue Edit Text Style 293
- avertissements
  - activation et désactivation 87
  - affichage 119
  - fenêtre de messages, recherche 15
  - nombre limité 120
  - sélection 119

## B

- b (option BCC.EXE) 176, 210
  - (attribue un mot complet pour enums) 63
  - traitement de code assembleur 90
- barre d'état
  - texte affiché 13
- barre d'icônes
  - boutons estompés 9
  - Browser 16
  - modification 11
- barre de défilement 289
  - ajouter aux contrôles 278
  - ajouter dans le cadre d'une boîte de dialogue 15-267
  - options 290
    - boîte de dialogue Edit Text Style 293
    - vertical (contrôle boîte de dialogue) 272
- barre menu, rupture de ligne 313

- BBN\_GOTABTAB 472
- BBN\_GOTATAB 472
- BBN\_SETFOCUS 472
- BBN\_SETFOCUSMOUSE 472
- BCC
  - macros, définition 62
- BCC.EXE
  - aide en ligne de commande 48
  - BCC32.EXE 47
  - options
    - dans les fichiers 49
    - désactiver 48
    - en ligne de commande 48
    - priorité 48 - 3-49
  - règles de priorité des options 49
  - syntaxe 47 - 3-48
  - TLINK et 171
  - utilisation 47
- BCC32.EXE
  - BCC.EXE 47
  - options, désactiver 48
  - TLINK et 171
- BC32DEF.CSM 68
- BCCONFIG.BCW 13
- BCDEF.CSM 68, 479 - C-480
- BCWDEF.DSW 13
- BG dans la palette Colors 346
- bibliothèques 171 - 9-172
  - applications DPMI et 172
  - applications Win32 et 172
  - déboguer 138
  - fichiers 60
  - mathématiques 172
  - options de compilateur en ligne de commande 61
  - polices 237
  - recherche 61
  - recréation 65
  - répertoires 60
  - répertoires, option en ligne de commande 60
  - TLINK et 171
  - TLINK, ignorer 178
  - TLINK32 et 172
  - virgule flottante 64
  - Windows 3.x applications et 171
- bitmaps
  - accessoires pour boutons BWCC 474
  - astuces et réserves 477
  - Black Frame (type de contrôle statique) 296
  - Black Rectangle (type de contrôle statique) 296
  - BM\_SETCHECK (BWCC) 475
  - BN\_CLICKED 288
  - boîte à options combinées (contrôle boîte de dialogue) 272
  - Borland Button Style (boîte de dialogue) 473
  - Borland Check Box Style (boîte de dialogue) 475
  - Borland Radio Button Style (boîte de dialogue) 475
  - Borland Shade Style (boîte de dialogue) 475
  - Borland Static Text Style (boîte de dialogue) 476
  - Boîte de peinture 350
    - options de couleur 350
    - problèmes de remplissage 351
  - boîtes à options combinées 297
  - boîtes de dialogue 263
    - option de grille 277
    - affectation de classes personnalisées 469
    - affecter des classes personnalisées 269
    - astuces et réserves 477
    - Borland Check Box Style 475
    - Borland Shade Style 475
    - Borland Static Text Style 476
    - comparer 303
    - composants (illustration) 263
    - contrôles 269
      - barre de défilement vertical 272
      - boîte à options combinées 272
      - boîte groupe 272
      - boîte liste 272
      - bouton radio 271
      - bouton-poussoir 271
      - case à cocher 272
      - édition texte 272
      - personnalisés 273
      - rectangle noir 273
      - statique texte 272
      - statiques en forme d'icône 272
      - catégorie 269
    - coordonnées (illustration) 277
    - créer 263
    - définition 233, 263
    - déplacer 265

- éditer 264
- enregistrer 302
- menus 268
- modification d'une classe de contrôle 279
- non-modales 268
- option de grille Relative 277
- option de style Caption 278
- option de style Control ID 278
- option de style Scroll Bar 278
- polices 268
- position définie par Windows 269
- protéger les soeurs des fenêtres 267
- redimensionner 265
- regrouper des contrôles 281
- Select Font 268
- sélectionner 265
- spécifier les contrôles comme des taquets de tabulation 280
- styles de dialogue 266
- Style (contrôles) 277
- styles
  - Absolute Align 267
  - Clip Children 267
  - Clip Siblings 267
  - Horizontal Scroll 267
  - Local Edit 267
  - Maximize Box 267
  - Minimize Box 267
  - Modal Frame 267
  - No Idle Messages 267
  - System Menu 266
  - System Modal 267
  - Thick Frame 267
  - Vertical Scroll 267
  - Visible 268
- styles de cadres 266
  - Border 266
  - Caption 266
  - Dialog Frame 266
  - No Border 266
- styles de fenêtre
  - Child 266
  - Overlapped 266
  - Popup 266
- tester 301
  - texte, entrée utilisateur 292
  - visualiser deux boîtes en même temps 303
  - zone client 267
- boîtes de dialogue (illustration)
  - New Project 239
  - Size Controls 285
  - Window Style 265
- boîtes groupe
  - options 287
  - contrôle boîte de dialogue 272
- boîtes liste 290
  - contrôle boîte de dialogue 272
  - dans des boîtes à options combinées 297
  - trier les éléments 299
- Border (attribut de contrôle) 279
- Border (style de cadre), boîtes de dialogue 266
- Borland C++ 3.1
  - utilisation 21
- bouton droit de la souris
  - éditeur Bitmap 346
- Bouton radio Borland 471
- Bouton-poussoir Borland 471
- boutons
  - Browser 16
  - modification dans les barres d'icônes 11
  - personnalisés 288
- boutons de contrôle BWCC 473
- boutons radio 271, 288
  - BWCC 475
  - options 287
- boutons-poussoirs 271, 287
  - BWCC 473
  - ID de contrôle prédéfinis (BWCC) 473
  - identificateurs de contrôle prédéfinis (Windows) 288
    - options 287
    - par défaut (BWCC) 473
- BRC
  - appel 195
  - outil de ressource 191, 195
- BRCC (compilateur de ressources Borland 191 - 10-192
  - appel 192
- Break Before (options), menus 313
- Breakpoint (commande) 133

Breakpoint Properties (boîte de dialogue) 134

Breakpoints (fenêtre) 133

Brief

raccourcis clavier de l'éditeur 13

Browse

nouveau projet et 27

Browser (EDI Windows)

barre d'icônes 16

objets 17

et définition de classe 16

filtres 17

activation 16

bugs 119

BUILTINS.MAK

description 208

texte 208

Button Style (boîte de dialogue) 287

BWCC

Auto check box 475

Auto radio button 475

boîtes de dialogue

Borland Button Style 473

Borland Check Box Style 475

Borland Radio Button Style 475

Borland Shade Style 475

boutons de contrôle 473

cases à cocher

3-state 475

Auto 3-state 475

classe BORDLG 469

contrôles 470

messages

BBN\_GOTABTAB 472

BBN\_GOTATAB 472

BBN\_SETFOCUS 472

BBN\_SETFOCUSMOUSE 472

boutons et cases à cocher 472

Option owner-draw 472

programmes en langage C, utilisation dans les 477

BWCC, bitmaps

accessoires pour boutons 474

décalages 474

BWCC, contrôles

bouton radio Borland 471

bouton-poussoir Borland 471

case à cocher Borland 471

groupe ombré 471, 475

ligne horizontale en creux 471

ligne verticale en creux 471

lignes en creux, conversion 475

lignes en relief, conversion 475

texte statique 471

BWCCAPI.RW 469

BWCCSTYL.RW 469

## C

.C (fichiers) 176, 202 - 11-203

commentaires imbriqués 65

compilation 48

sans édition de liens 90

C++

code non optimal 89

code obsolète 88

erreurs 88 - 3-89

gestion d'exceptions 80

cadre de pile standard

génération 66

cadre de sélection 265, 304

cadre noir (contrôle boîte de dialogue) 272

cadres

boîtes de dialogue 266

dessiner 352

calculs flottants

règles de conversion ANSI 64

Call Stack

commande de menu 125

fenêtre 125

Caption (fenêtre) 279

Caption (option Style) 278

Caption (style de cadre), boîtes de dialogue 266

caractère souligné dans les menus 314

caractère Tab

texte statique 295

caractères

ajouter à une ressource police 390

définir la largeur dans une ressource police 391

caractères de soulignement 65

génération automatique 65

Case (options), boîte de dialogue Edit Text Style

293

- case 3-state (BWCC) 475
- Case à cocher Borland 471
- case à cocher 3-State 288
  - BWCC 475
- cases à cocher 272, 287
  - BWCC 475
  - options 287
- CBS\_DISABLENOSCROLL (style) 299
- \_\_cdecl 69, 77
  - option en ligne de commande 69, 77
- CGA Resolution (commande), éditeur Icon 372
- chaînes
  - littérale, fusion 63
  - ressources définies par l'utilisateur 399
  - terminées par un zéro 400
  - transfert de constantes dans le segment de code 71
- Character (options), polices 391
- Check Dup Keys (commande), éditeur Accelerator 330
- Check Duplicates (commande), éditeur Menu 317
- Checked (option), menus 313
- CheckDlgButton (BWCC) 475
- CheckMenuItem (fonction) 313
- CheckRadioButton (BWCC) 475
- Child
  - style de fenêtre 266
- ciseaux (outil) 348
- .CKB (fichier)
  - description 13
- classe BORDLG 469
- classe de contrôle (boîtes de dialogue), changer 279
- classes
  - ajout (ClassExpert) 109
  - ClassExpert et 107
  - et DLL 199
  - partage d'objets 72
  - visualisation 107
- Classes (volet de ClassExpert)
  - description 108
- classes personnalisées 269, 469
- ClassExpert
  - accès au code source 114
  - classes
    - ajout 109
    - visualisation 107
  - coches 112
  - description 107
  - gestionnaires
    - ajout 112
    - suppression 112
  - parties 107
  - source, visualisation 114
  - variables d'instance
    - ajout 113
    - suppression 113
  - volets 107
    - Resource Workshop et 114
- clavier dans les boîtes de dialogue 280
- clic (technique de sélection) 273
- clignotement des éléments de menu 330
- Clip Children (style), boîtes de dialogue 267
- Clip Siblings (style), boîtes de dialogue 267
- coches
  - événements et (ClassExpert) 112
- code
  - non optimal (mises en garde) 89
  - optimisation 81
- combinaison d'attributs 179
- Combo box (contrôle boîte de dialogue) 272
- Combo Box Style (boîte de dialogue 297
- COMDEF
  - génération 65
- commandes
  - ajouter aux menus 312
  - ligne d'état de l'éditeur Bitmap 355
  - raccourcis 314, 323
  - ressource menu
    - activer 313
    - désactivées 313
    - éditer 312
    - en grisé 313
    - identificateurs 315
    - raccourcis 315
    - raccourcis alignés à droite 315
    - supprimer 316
    - tabulations 315
  - surgissantes 307
    - ajouter aux menus 312
- commandes MAKE



- règles 217
- commentaires
  - dans les scripts ressource 244
  - imbriqués 65
  - suppression d'une bibliothèque 205
- compilateur
  - compilation par défaut des fichiers 48
  - extensions de fichiers et compilation 48
  - fichiers de configuration, priorité 48
  - optimisations 81
  - options
    - par défaut 48
    - désactiver 48
  - règles de priorité des options 49
  - table d'options 50 - 3-60
- compilateur 16bits
  - description 47
- compilateur 32 bits
  - description 47
- compilateur de ressource Microsoft
  - Resource Workshop, incompatibilité 279
- compilateur en ligne
  - description de BCC et BCC32 47
  - macros, définition 62
  - options,
    - f (émulation 80x87) 64
    - émulation 80x87 (-f) 64
- compilateur en ligne de commande
  - description 47
  - TLINK et 171
- compilation par commande assembleur 90
- compilation
  - erreurs 119
  - règles 47
  - table d'options 50 - 3-60
- Compile Now (commande) 318
- compiler les ressources 192
- conditionnel
  - point d'arrêt 136
- configuration
  - de l'EDI 10
  - de l'éditeur de l'EDI 13
  - options 258
  - raccourcis clavier 13
- consigner les expressions 136
- constantes
  - visualiser 242
- CONTROL (instructions), générer 305
- Control (menu), boîtes de dialogue 266
- Control Type (options), boîte de dialogue Static Style 295 - 15-296
- contrôles de boîte de dialogue
  - Ligne verticale en creux 471
  - Bouton radio Borland 471
  - Bouton-poussoir Borland 471
  - Case à cocher Borland 471
  - Groupe ombré 471
  - Ligne horizontale en creux 471
  - Texte statique 471
- contrôle de mise en garde, TLINK et 181
- contrôle de nom de segment
  - options de compilateur en ligne de commande 72
- contrôles 263, 269
  - affecter un identificateur de contrôle 278
  - afficher la boîte de dialogue Style 277
  - ajouter 274
    - des barres de défilement 278
    - des libellés 279
  - aligner 282
    - avec une grille 277
  - attributs communs 278
  - barres de défilement 289
    - vertical 272
  - boîte de saisie 292
  - boîtes à options combinées 272, 297
  - boîtes groupe 272
  - boîtes liste 272, 290
  - bordure 279
  - boutons radio 271
  - boutons-poussoirs 271
  - BWCC 270, 470
  - case à cocher 272
  - catégories 269
  - changer la taille 276
  - coordonnées, définir 276
  - définir l'ordre 281
  - définition 269
  - déplacer 276
  - disposer en colonnes et rangées 284
  - éditer 277

- édition texte 272
- en grisé 279
- espacés régulièrement 283
- grouper 279
- hauteur, spécifier 276
- largeur, spécifier 276
- modifier 277
- options communes aux boîtes Style 278
- options de style bouton 287
- palette Tools 270
- personnalisés 270, 273, 279, 299
- propriétés 275
- rectangle noir 273
- redimensionner 285
- sélectionner 273
  - avec Tab 274
- statique texte 272
- statiques 294
- supplémentaires, sélectionner 274
- supprimer d'un groupe sélectionné 274
- tester 301
- Windows 270
- contrôles boîte de saisie
  - allouer au segment local 267
  - conversion texte OEM 299
  - dans des boîtes à options combinées 297
- contrôles boutons 287
  - Auto Check Box 287
  - Auto Radio Button 288
  - case à cocher 3-State 288
  - case à cocher Auto 3-State 288
  - Check Box 287
  - Default Push Button 287
  - Group Box 288
  - justification du texte 288
  - Owner Draw 288
  - Push Button 287
  - Radio Button 288
  - User Button 288
  - valeurs d'identificateur de contrôle prédéfinies 289
- contrôles BWCC
  - bouton radio Borland 471
  - bouton-poussoir Borland 471
  - case à cocher Borland 471
  - groupe ombré 471, 475
  - ligne horizontale en creux 471
  - ligne verticale en creux 471
  - texte statique 471
- contrôles personnalisés
  - options de dessin 305
- contrôles statiques 272
  - identificateurs de contrôle 278
- contrôles texte, distinction minuscule/majuscule 293
- conventions d'appel 69, 77
  - \_cdecl 69, 77
  - \_fastcall 69, 77
  - \_pascal 69, 77
  - C 69, 77
  - de registre 69
  - fastcall 63
  - Pascal 69, 77
  - Register 63, 69, 77
  - standard 77
- conventions typographiques (documentation) 2
- conversions
  - calculs flottants, règles ANSI 64
  - pointeurs, suspects 87
- Convert OEM (option), boîte de dialogue Edit Text Style 294
- Convert OEM option (Edit Text Style dialog box) 299
- coordonnées en pixels
  - curseurs 382
  - ligne d'état de l'éditeur Bitmap 355
- coordonnées, spécifier, contrôles 276
- copier une ressource entre projets 247
- copies (noeuds)
  - police et 36
- copies de sauvegarde 259
- coprocesseurs numériques
  - émulation 64
  - générations de code 64
- Copy (commande) 247
- couleurs
  - curseurs 381
  - de premier plan 346
  - du texte dans l'éditeur 14
  - éditeur Bitmap 356

- éditer 360
- fond 346, 357
- graphique par points 356
- icônes 356
- index de palette 360
- inversées 382
- premier plan 346, 357
- transparentes 382
- icônes 373
- mise en surbrillance de la syntaxe et 14
- couleurs inversées
  - curseurs 357, 382
  - icônes 357, 372
- couleurs transparentes 357
  - curseurs 357, 382
  - icônes 357, 372
- .CPP, fichiers
  - compilation 48
- créer des fichiers identificateurs 252
- créer une boîte de dialogue 263
- .CSM, fichiers 479 - C-480
  - espace disque 481
  - nom par défaut 480
  - plus petit que prévu 481
- .CUR, fichiers 381
- curseur
  - définition 234
  - position affichée dans la barre d'état 13
- Curseurs 379
  - astuces et réserves 477
  - couleurs inversées 357, 382
  - couleurs transparentes 357, 382
  - créer 380
  - Edit Background Color (commande) 358
  - Edit Foreground Color (commande) 358
  - éditer 381
  - fichier .CUR autonome 380
  - format binaire 380
  - palette Colors 381
  - point actif 382
  - programmes Windows 379
  - type de fichier .CUR 236
- CUSTCNTL.RW 299
- Custom (commande) 300

## D

- D (option BCC.EXE) 210
  - définitions de macros 62
  - fusion de chaînes littérales 63
- Data Inspector (fenêtre) 126, 131
- date et heure
  - modification 209
- dc (option BCC.EXE)
  - transfert des constantes chaînes dans le segment de code 71
- débogage
  - Voir aussi déboguer
  - informations 121
    - fichiers .EXE ou .OBJ 77
    - option de compilateur en ligne 67
    - TLINK 177
  - numéros de ligne, .OBJ (fichiers) 67
  - options de compilateur 67
  - TLINK et 181
- déboguer 119
  - assembleur 124
  - code externe 124
  - DLL 138
  - fonctions membres 124
  - recommencer au début 126
  - ressources menu 317
  - tables de raccourcis 330
- débogueur intégré 119
- Debug Information in OBJS 121
- Debug/Browse | Test stack overflow 67
- décalages bitmap (BWCC) 474
- Décompiler les ressources 236
- Default Push Button (contrôle) 287
- défaut
  - options, désactiver 48
- #define
  - options de compilateur en ligne 62
  - options de compilateur en ligne de commande, séparation 62
- Defines (EDI) 62
- Delete All Breakpoints (commande) 134
- Delete Breakpoint (commande) 134
- démarrage de TLINK (éditeur de liens) 167
- dépendances automatiques
  - information, désactivation 65

- déplacement (browsing) dans l'EDI Windows
  - symboles globaux 18
- déplacer une boîte de dialogue 265
- déroulant (menu) 308
- désactiver les commandes de menu 313
- Device Info (bouton-poussoir), boîte de dialogue
- Icon Image Attributes 374
- DFA avec WinSpector 162
- DFA.OUT 162
- Dialog (éditeur)
  - composants de fenêtre 264
  - définir les préférences 304
  - fenêtre Caption 264
  - lancer 263
  - ligne d'état 264
    - unités de mesure 304
  - modes 270
  - options d'affichage 300, 304
  - options Selection 305
  - options Selection Border 304
  - palette Alignment 264
  - palette Tools 264
  - personnaliser 304
- Dialog Frame (style de cadre), boîtes de dialogue 266
- directives MAKE 223
  - conditionnelles 225
- Directories (boîte liste, boîte de dialogue Open Project) 240
- Disabled (attribut de contrôle) 279
- Disabled (options), menus 313
- diviser la fenêtre de l'éditeur Bitmap 354
  - annuler 354
- DLG Resource Script (entrée) 302
- DLL
  - bibliothèques d'import 197, 200
  - classes 199
  - déboguer 138
  - fonctions d'export cachées 200
  - noms non significatifs 200
  - segments de code, stockage de 179
- documentation
  - conventions typographiques 2
- Don't Redraw (option de style List Box) 291
- données

- ajouter à une ressource définie par l'utilisateur 398
- DOS
  - écrasement et éditeur de liens 178
- DPMIMEM 21
- Draft (option d'affichage) 304
- Draw Custom Controls as Frames (préférences de l'éditeur Dialog) 305
- Draw on Both Images (option) 366
- Drawing Type (options) 304 - 15-305
- Duplicate (commande), éditeur Dialog 275
- Duplicate (outil) 275

## E

- e (option BCC.EXE) 177, 210
  - assembleur à utiliser 91
  - nom de programme EXE 91
- EasyWin 485
- écrasement
  - l'éditeur de liens et 178
- EDI
  - configuration 10
  - groupement de plusieurs répertoires 61
  - lancement 8
  - macros, définition 62
  - menus 9
  - utilisation 5
- Edit (commande) 243
- Edit (volet de ClassExpert)
  - description 108
- Edit as Text (commande) 243
- Edit Background Color (commande) 358
- Edit Foreground Color (commande) 358
- Edit Icon (bouton), boîte de dialogue Static Style
  - lancer l'éditeur Bitmap 296
- Edit Image (commande) 373
- Edit Source (commande) 135
- Edit Text Style (boîte de dialogue) 293 - 15-294
- éditer
  - boîtes de dialogue 264
  - identificateurs 256
  - menus 311
  - ressources définies par l'utilisateur 397
  - tables de chaînes 340
  - tables de raccourcis 328

- éditeur
  - ClassExpert et 108
  - couleur du texte 14
  - mise en surbrillance de la syntaxe 14
  - mise en surbrillance du texte 14
  - position du curseur sur les erreurs 15
  - volet Edit et (ClassExpert) 109
- éditeur Bitmap 345
  - aligner la zone sélectionnée 365
  - BG dans la palette Colors 346
  - coordonnées en pixels
    - curseurs 382
    - ligne d'état 355
  - couleurs 356
    - bitmaps 356
    - icônes 356
  - dilater les zones sélectionnées 365
  - explications de commandes (ligne d'état) 355
  - FB dans la palette Colors 346
  - fenêtres 354
  - FG dans la palette Colors 346
  - forme de pinceau 363
  - index de palette 355
  - informations d'outil (ligne d'état) 355
  - informations de couleurs (ligne d'état) 355
  - lancer 296, 345
  - ligne d'état 355
  - motifs 363
  - options globales 366
  - outils 346
    - Aérographe 350
    - Boîte de peinture 350
    - Ciseaux 348
    - Ellipse, cadre plein 353
    - Ellipse, cadre vide 352
    - Gomme 349
    - Ligne 351
    - Main 353
    - Pinceau 350
    - Rectangle de sélection 348
    - Rectangle, cadre plein 353
    - Rectangle, cadre vide 352
    - Stylo 349
    - Texte 352
    - Zoom 348
  - palette Colors, index 360
  - palette Tools 346
  - Rectangle arrondi, cadre plein 353
  - Rectangle arrondi, cadre vide 352
  - redimensionner les zones sélectionnées 365
  - sélectionner les outils 346
  - sélections de style 354
  - styles de ligne 364
  - texte 352, 361
  - types de ressources édités 345
  - valeurs RGB 359
    - définir 360
    - ligne d'état 355
  - volet de fenêtre actif 354
  - vues multiples 354
- éditeur Brief
  - émulation 13
- éditeur de texte
  - ressources définies par l'utilisateur 397
  - touches 399
- éditeur de texte externe 241
  - sélectionner 244
  - utiliser 244
- éditeur de texte interne 243
  - commande Compile Now 318
- éditeur Dialog
  - option Parent Notify 472
- éditeur Epsilon
  - émulation 13
- éditeur Notepad (Windows) 244
- éditeur String 335
  - activer 336
  - champs 340
  - illustration 337
  - lancer 336
- éditeurs
  - sélectionner 244
- éditeurs de ressources 235
  - Accelerator 323
  - Bitmap 345
  - Cursor 379
  - Font 385
  - Icon 369
  - String 335
  - lancer (boîte de dialogue Identifiers) 257

- Menu 307
- ressources définies par l'utilisateur 395
- édition de liens
  - compilateur de lignes de commande et 171
  - modules non-C et 178
  - options
    - à partir du compilateur en ligne 91
    - de compilateur en ligne de commande 91
    - tableau 173
  - table complète 91
  - table de liaison, création 91
- édition texte (contrôle boîte de dialogue) 272
- Editor Options (commande), éditeur Bitmap) 366
- effacer un point d'arrêt 133
- EGA/VGA Resolution (commande), éditeur Icon 372
- éléments
  - mise en surbrillance de la syntaxe 14
- éléments de données
  - inspecter 131
  - modifier la valeur 131
- éléments de menu 308
  - ajouter 312
  - sélectionner 311
- !elif (directive MAKE) 223, 225
- Ellipse
  - cadre plein 353
  - cadre vide 352
- !else (directive MAKE) 223, 225
- EM\_SETHANDLE/EM\_GETHANDLE (messages) 267
- émulation 80x87 64
- en-têtes, fichiers 252
- en-têtes (polices) 392
- en-têtes précompilés
  - contrôle 480
  - fonctions membres inline 480
  - inconvenients 480
  - optimisation 482
  - options en ligne de commande 68
  - règles 481
- Enabled (option), menus 313
- EnableMenuItem (fonction) 313
- !endif (directive MAKE) 223, 225
- enfant
  - de la fenêtre active 266
- enregistrement
  - fenêtres de l'EDI et 12
- enregistrer
  - ressources 258
- entiers 68
  - alignés sur une limite de mot 68
- énumérations (enum)
  - attribution d'entiers 87
  - traitées en tant qu'entiers 63
- Environment Options (boîte de dialogue)
  - préférences 12
- Epsilon
  - raccourcis clavier de l'éditeur 13
- erreur de protection générale 137
- erreurs 119
  - à la compilation (syntaxe) 119
  - compilation (syntaxe), causes les plus fréquentes 120
  - dans le code 15
  - à l'exécution (sémantique), causes 120
  - fenêtre de messages, recherche 15
  - limiter l'affichage 120
  - localiser 119
  - logique 120
  - option de rapport du compilateur en ligne de commande 87
  - types 119
  - violations ANSI 87
- erreurs à la compilation
  - causes les plus fréquentes 120
  - dans le code 15
  - définition 119
- erreurs de logique
  - définition 120
- !error (directive MAKE) 223
  - description 224
- ES\_READONLY (style) 294
- ES\_WANTRETURN (style) 294
- espace disque saturé 481
- estompés
  - boutons de la barre d'icônes 9
- Evaluate Expression (boîte de dialogue) 126
- Evaluate/Modify (boîte de dialogue) 129
- évaluer une expression 129

- événements
  - énumération dans ClassExpert 108
- Event Log (fenêtre) 136
- Events (volet de ClassExpert)
  - description 108
- .EXE (fichiers), TLINK et 181
- exécution
  - en mode pas à pas 122
  - erreurs, définition 120
  - erreurs, causes 120
- expressions 126
  - consigner 136
  - définition 126
  - évaluer 129
  - formatage 129
  - suivi 127
  - suivre 129
- expressions booléennes
  - condition de point d'arrêt 136
- expressions de suivi
  - activer 129
  - désactiver 129
  - modifier le format d'affichage 127
  - modifier les propriétés 129
  - supprimer 129
- Extend Select (option de style List Box) 292
- extensions
  - compilateur 48
- extensions de fichier
  - fournies par TLINK 169
- extensions pour mise en surbrillance de la syntaxe
  - ajout 14
- extraire et supprimer (action TLIB) 206

**F**

- f (option BCC) 177, 210
  - émulation 80x87 64
- F1 (touche) 10
- \_\_fastcall 77
  - convention d'appel 69
  - option en ligne de commande 69, 77
- fastthis
  - convention d'appel 63
  - option en ligne de commande 63
- FB dans la palette Colors 346
- Fc (option BCC)
  - génération de COMDEF 65
- fenêtre de messages
  - recherche automatique d'erreur 15
  - utilisation 15
- fenêtre projet 241
  - contenu 241
  - options d'affichage 242
  - ressources, sélectionner 243
- fenêtres
  - dans l'EDI, limite imposée par la mémoire disponible 13
  - éditeur Bitmap 354
- fenêtres Browser
  - utilisation 16
- fenêtres d'édition
  - configuration 13
- fenêtres de l'EDI
  - modification 13
- fenêtres enfant
  - protéger la zone client 267
- fenêtres superposées
  - boîtes de dialogue 266
- fenêtres surgissantes
  - boîtes de dialogue 266
- ff (option BCC.EXE)
  - calculs flottants rapides 64
  - variables globales far 71
- FG dans la palette Colors 346
- fichier 226
  - .DRV 236
- fichier DLL 236
- fichier exécutable 236
- fichier MAKE
  - désignation 210
- fichier projet .FNT 388
- fichier projet .ICO 371
- fichier ressources
  - définition 235
- fichiers
  - .BMP 236
  - .CUR 236
  - .DLG 236, 302
  - .FNT 237
  - .FON 237

- .H 252
- .ICO 236
- .RES, enregistrer les ressources 259
- .RES, identificateurs 251
- changement de la date et de l'heure 209
- compilation 92
  - C ou C++ 48
- copies de sauvegarde 259
- créés par AppExpert 96
- en-tête 252
- extensions 169
  - et compilateur 48
- identificateur
  - ajouter aux projets 252
  - langage C 252
- mise en surbrillance de la syntaxe 14
- recherche 169
- visualisation à partir de ClassExpert 108
- fichiers .COM, TLINK et 180
- fichiers include utilisateur 60
- fichiers mappe 91
- fichiers .RSP
  - définition 49
- fichiers d'en-tête
  - recherche 61
- fichiers de configuration
  - compilateur en ligne de commande 48
  - compilateur en ligne, priorité 49
  - compilateur en ligne, règles de priorité 49
  - priorité 48
  - TURBOC.CFG 48
- fichiers de définition de modules
  - description 183
  - différence majuscules/minuscules et 176
  - NAME absent 188
  - TLINK et 183
  - /Tw option TLINK et 181
- fichiers de mappe
  - débogage 178
  - générés par TLINK 178
- fichiers de recherche
  - définis pour BCC.EXE 49
  - définis pour TLINK 170
  - et MAKE 219
  - priorité 49
- TLINK et 170
- fichiers DLL
  - éditer les ressources 260
  - installer 300
- fichiers en ligne
  - BWCCAPI.RW 469
  - BWCCSTYL.RW 469
  - CUSTCNTL.RW 299
- fichiers exécutables
  - éditer les ressources 259
  - enregistrer lesressources 258
  - identificateurs 251
- fichiers include
  - options de compilateur en ligne de commande 61
  - recherche 61
  - répertoires 60
- fichiers MAKE
  - commandes 215
  - continuation de ligne 215
  - débogage 219
  - option KEEP 211
  - option NOKEEP 212
  - règles implicites 215
- filtres
  - Browser de Windows 17
  - fixe, largeur de police 389
- Fixed (option Owner Drawing 290
- Floating Point (option) 128
- Fm (option BCC)
  - activation des options -F 91
- .FNT, fichiers
  - créer de nouvelles polices 388
- .FON, fichiers
  - créer 394
- fonctions
  - conventions d'appel 69, 77
  - énumération dans ClassExpert 108
  - inline avec les en-têtes précompilés C++] 480
  - le Browser sous Windows 18
  - recherche d'une fonction 125
  - vide, production d'une valeur vide 87
- fonctions membres
  - déboguer 124
- fonctions virtuelles



- membres cachés dans les classes dérivées avec pointeurs 79
- fond (couleurs) 346, 357
- Font (commande) 362
- Font Size (commande) 389
- Font Version (option), en-têtes de police 393
- format binaire
  - curseurs 380
  - icônes 370
  - polices 237, 387
- format source
  - icônes 370
  - polices 387
- formatage
  - expressions de suivi 127
  - expressions 129
- Fs (option BCC)
  - DS = SS supposé 71

## G

- /Gx option TLINK 177
- generate, bouton 97
- génération de code
  - options de compilateur en ligne de commande 63
  - résultats 16 bits 47
  - résultats 32 bits 47
- gestion d'exceptions
  - option de compilateur 80
- gestionnaire de fichiers
  - utilisation dans un projet (noeuds) 34
- gestionnaire de projet
  - installation de traducteurs 42
  - noeuds dépendants et 28
  - noeuds
    - "tirer et lâcher" 34
    - ajout 34
- gestionnaires
  - ajout (ClassExpert) 112
  - suppression (ClassExpert) 112
- gestionnaires de visualisation
  - Resource Workshop et 114
- gn (option BCC.EXE)
  - interruption sur n avertissements 87
- Gomme 349
  - affectations de couleurs 357

- affectations des couleurs 346
  - options de couleur 349
- Groupe ombré 471
- GP
  - type d'erreur 137
- graphiques par points
  - définition 234
  - options de couleurs 356
  - Voir aussi* Bitmaps
- Gray Frame (type de contrôle statique) 296
- Gray Rectangle (type de contrôle statique) 296
- Grayed (option), menus 313
- Grid (commande) 277
- Grid on zoomed windows (option) 367
- grille
  - afficher 277
  - aligner les contrôles 277
- grisé (contrôles 279
- griser les commandes de menu 313
- Group (attribut de contrôle) 279
- Group Box (contrôle) 288
- grouper des contrôles 279
- groupes (contrôles boîte de dialogue) 281
  - et sélections multiples 281

## H

- H (option BCC.EXE)
  - en-têtes précompilés 68
  - pointeurs fast huge 72
- hard (mode de débogage) 139
- Has Strings (option de style List Box) 292
- hdrfile, pragma 480 - C-481
- hdrstop, pragma 480, 483
- Header (commande) 392
- Help Break (option) 313
- Hide Palette (commande) 359
- Home Budget (icône) 375
- Horizontal Scroll (style), boîtes de dialogue 267
- .HPJ (fichiers)
  - création 96

## I

- I (option BCC.EXE) 177, 210
  - longueur d'identificateur 65

- répertoire de fichiers include 60
- Icon Image Attributes (boîte de dialogue) 374
- .ICO, fichiers
  - créer de nouvelles icônes 371
  - à trois dimensions 376
- icônes 369
  - afficher des informations périphériques 374
  - astuces et réserves 477
  - CGA Resolution (commande) 372
  - changer la résolution 374
  - changer les attributs 374
  - contrôles statiques dans les boîtes de dialogue 297
  - couleurs inversées 357, 372
  - couleurs transparentes 357, 372
  - créer 369, 375
    - trois dimensions 376
  - définition 234
  - dessiner, calculatrice 375
  - Edit Background Color (commande) 358
  - Edit Foreground Color (commande) 358
  - éditer 371
  - effacer 376
  - EGA/VGA Resolution (commande) 372
  - fenêtres réduites 369
  - fichier .ICO autonome 371
  - format binaire 370
  - format source 370
  - images multiples, ajouter 372
  - images, ajouter 372
  - options d'affichage 373
  - options de couleur 356, 373
  - projet exemple 375
  - type de fichier .ICO 236
  - utilisées dans la documentation 2
- identificateur d'élément (menus) 313, 315, 317
- identificateur de contrôle
  - option Style 278
  - valeurs Windows prédéfinies, boutons-poussoirs 288
  - valeurs BWCC prédéfinies de WORD 473
- identificateurs 250
  - ajout automatique 253
  - ajouter 254
  - annulation 93
  - caractères de soulignement 65
  - commandes menu 313
  - composants 250
  - contrôles statiques icône 297
  - création automatique 253
  - de chaînes 338
  - de contrôle 278
  - de ressource 250
  - déplacer 256
  - éditer 256
    - et raccourcis 329
  - fichiers 252
  - fichiers exécutables et ressources compilées 251
  - lancer un éditeur ressource 257
  - liste 257
  - longueur significative 63, 65
  - macro MAKEINTRESOURCE 251
  - menus 315
  - modifier la valeur 251
  - raccourcis 326
  - renommer 256
  - ressources définies par l'utilisateur 396
  - stocker 252
  - suppression automatique 253
  - supprimer 256
  - tables de chaînes 337 - 18-338
  - touches virtuelles 324
  - transtyper 251
  - type MakeIntResource 251
  - unique dans un type de ressource 251
  - vérifier les doublons, menus 317
  - vérifier les doublons, raccourcis 330
  - visualiser 242
- Identifiers (boîte de dialogue)
  - ajouter des identificateurs 255
- !if (directive MAKE) 223, 225
- !ifdef (directive MAKE) 223, 225
- !ifndef (directive MAKE) 223, 225
- .ignore (directive MAKE) 223
- images bitmap
  - ajouter du texte 352
  - couleurs
    - fond 346
    - premier plan 346
  - effacer 349
  - pixels 346

- remplir de couleur 350
- sélectionner des zones irrégulières 348
- sélectionner des zones rectangulaires 348
- sélections de style 354
- texte 361
- voir, vues multiples 354
- zoomer 348
- images zoomées 348
  - outil Aérographe 350
  - outil Pinceau 350
- #include (directive) 223
  - crochets angulaires () 61
  - description 226
  - guillemets 61
- index de palette 355
- indicateur Want Return (BWCC) 473
- informations de débogage 121
- Initial State (options), menus 313
- inspecter
  - éléments de données 131
- Install Control Library (commande) 300
- instructions 80286
  - 16 bits 68
- instructions 80386
  - 16 bits 68
  - 32 bits 76
- instructions 80486
  - 16 bits 68
  - 32 bits 76
- instructions dans le volet Outline 310
- instructions Pentium
  - 32 bits 76
- Integral Height (option de style Boîte à options combinées) 299
- Integral Height (option de style List Box) 291
- Intel
  - définition des champs ACBP 179
- interface graphique GDI 395
- interface utilisateur, personnaliser 260
- interrompre un programme 126
- Invalid Breakpoints (boîte de dialogue) 135
- inverser les éléments de menu 330
- Invert Menu Item (option), éditeur Accelerator 330

## J

- Jg (options BCC.EXE)
  - options de génération de modèles 80
- jeu de caractères ANSI 294, 299
  - correspondance avec les polices 390
- jeux de caractères
  - correspondance avec les polices 390
  - définition 389
- jeux de commandes
  - utilisation de l'EDI 13
- jn (option BCC.EXE)
  - interruption sur n erreurs 87
- journaliser
  - Voir consigner

## K

- s-K (option BCC.EXE) 210
  - ,caractères non signés 63
  - cadre de pile standard 66
- KEEP (option MAKE) 211
- Keep Selection (option), boîte de dialogue Edit Text Style 294
- Kernighan et Ritchie
  - mots-clés 66
- Key (option), menus 314
- Key Type (option), menus 314
- Key Value (commande), éditeur Accelerator 330
- Key Value (mode), éditeur Accelerator 327

## L

- L (option BCC.EXE) 177
  - options d'éditeur de liens 91
  - code objet et répertoire de bibliothèque 60
- lancement d'autres programmes 23
- lancement de Borland C++ 8
- lancer les éditeurs ressource
  - Accelerator 325
  - Bitmap 345
  - Dialog 263
  - Menu 308
  - String 336
- langage assembleur
  - compilation 90
  - fichiers de sortie 92

- options
  - suppression 92
  - passage 92
- procédures en ligne 90
- répertoire 60
- langage C
  - # define 252
  - fichiers en-tête 252
  - séquences d'échappement 339
- LB\_ADDSTRING (message) 292
- LB\_GETTEXT (message) 292, 298
- LB\_INSERTSTRING (message) 292
- LB\_SETCOLUMNWIDTH (message) 291
- LB\_SETTABSTOPS (message) 291
- LB\_SETTEXT (message) 298
- LBS\_DISABLENOSCROLL (style) 292
- .LIB (fichiers)
  - BCC 48
- libellé, ajouter
  - à une boîte de dialogue 266
  - aux contrôles 279
- librairies
  - Voir* bibliothèques
- Ligne 351, 382
  - options de couleur 351
  - styles 364
- Ligne horizontale en creux 471
- Ligne verticale en creux 471
- ligne d'état
  - éditeur Bitmap 355
  - éditeur Dialog 264, 304
- lignes
  - droites 351
  - numération, dans les fichiers objets 67
  - numérotation dans TLINK 177
  - atteindre 9
- lignes en creux verticales et horizontales (BWCC) 475
- lignes en relief verticales et horizontales (BWCC) 475
- Line (options), boîte de dialogue Edit Text Style 293
- List Box Style (options de boîte de dialogue) 290
- ListClasses (fonction) 270
- liste d'arguments de variables 77

- liste de classes 143
- listes multiples
  - options de compilateur en ligne de commande, #define 62
  - options de compilateur en ligne de commande, définition de macros 62
  - options de compilateur en ligne de commande, include et 61
- Load on Call (option mémoire) 249
- Load Symbol Table (commande) 139
- LOADBWCC.EXE 476
- Local Edit (style), boîtes de dialogue 267
- Locate Function (commande de menu) 125

## M

- M (option BCC.EXE)
  - table de liaison 91
- macros
  - définition 62
- Main 353
- Maj-clic (technique de sélection) 273
- majuscules/minuscules 203
  - TLINK 176
  - TLINK32 176
  - TLINKfichiers de définition de modules et 176
- MAKE
  - aide sur la ligne de commande 208
  - annuler les définitions de macro 210
  - arrêter 208
  - auto-dépendance 223
  - build forcé 209
  - BUILTINS.MAK 208
    - description 208
    - ignorer les règles 209
  - changer le répertoire utilisé 210
  - cibles multiples 213
  - cibles symboliques 213
    - règles 213
  - commandes 215
  - compatibilité avec NMAKE 211
  - composants, :: (plusieurs règles explicites) 215
  - contrôles d'erreur 224
  - créer les cibles 213
  - créer toutes les cibles 210
  - débogage 219

- définition 207
- définition de macros 210
- dépendance liée 208
- désactiver l'affichage 211
- description 207
- directives 223
  - !error 223
  - !include 223
  - liste 223
  - !message 223
  - !undef 224
  - .path.ext 224
  - .precious 224
  - règles conditionnelles 225
  - .suffixes 224
  - utilisation de macros 224
- erreurs 403
- état de sortie des programme 210
- fichier MAKEFILE 207
- fichiers MAKE
  - nom différent 210
- fichiers temporaires
  - conserver 210, 211
  - utilisés pour le débogage 219
- fichiers, affichage de la date et de l'heure 209
- ignorer l'état de sortie des programmes 210
- instructions 212
- KEEP (option) 211
- macros
  - \$d (tester) 229
  - définir 220
  - définition 220
  - ligne de commande ou fichier MAKE 221
  - liste de modificateurs 222
  - modifier 222
  - nom de fichier 221
  - par défaut (description) 222
  - par défaut (modifier) 209
  - substitution de chaînes 221
  - syntaxe 220
  - texte à utiliser 220
  - utiliser 221
  - vides 229
- mémoire cache et auto-dépendance 210
- modificateurs de commande 217
  - & (expansion macro) 217
  - (traitement codes d'erreur) 217
  - @ (inhiber la sortie) 217
- NMAKE, compatibilité 211
- NOKEEP (option) 212
- noms de macros, parenthèses 221
- opérateurs conditionnels 226
- opérateurs de commande, liste 218
- opérateurs en ligne de commande, && (créer fichier temporaire) 219
- options par défaut 210
- préfixes de commande 217
- règles explicites 214
  - plusieurs 215
  - sans commandes 215
  - syntaxe 214
- règles implicites 213
  - syntaxe 216
  - utilisées avec des règles explicites 216
- règles par défaut 207
- règles sur les commandes 217
  - format 213
  - ignorer 211
- sélectionner les options par défaut 210
- SHARE, incompatibilité 211
- sortir de la mémoire 211
- supprimer l'affichage 211
- syntaxe 208
- tâches par défaut 208
- texte à utiliser dans les macros 220
- TOUCH.EXE 209
- utiliser des fichiers MAKE 210
- variables d'environnement 210
- MAKEFILE
  - utilisation 212
- MAKEINTRESOURCE (macro) 251
- MakeIntResource (type) 251
- MAKER.EXE
  - définition 207
  - mémoire 211
- MAKESWAP.EXE 21
- Manual (mode), éditeur Accelerator 330
- Maximize Box (style), boîtes de dialogue 267
- mémoire
  - effet sur l'option Undo Levels 258

- et outils 32 bits en ligne de commande 21
- options 248
- ressources bitmap 387
- mémoire étendue
  - TLINK et 182
- mémoire insuffisante
  - et outils 32 bits 21
- mémoire paginée
  - TLINK et 182
- Memory Options (commande) 248
- Menu (éditeur) 307
  - commande View as Pop-up 310
  - composants de l'écran 309
  - définir l'affichage Test Menu 310
  - éditer les éléments de menu 312
  - lancer 308
  - options
    - Break Before 313
    - Checked 313
    - Disabled 313
    - Enabled 313
    - Grayed 313
    - Help Break 313
    - Item Help 313
    - Item ID 313
    - Item Text 313
    - Item Type 313
    - Key 314
    - Key Type 314
    - Menu Bar Break 313
    - Menu Break 313
    - Modifieurs 314
    - No Break 313
  - utiliser avec l'éditeur Accelerator 325
  - volet Attribute 309
  - volet Outline 310
  - volet Test Menu 309
  - volets 309
- Menu Bar Break (option), menus 313
- Menu Break (option), menus 313
- menu d'exemple 319
- menu Widgets
  - créer 319
  - raccourcis, créer 331
- Menu | Track test (menu) 317

- menus 307
  - activer des commandes 313
  - afficher
    - en menu flottant 310
    - les valeurs d'identificateurs 317
  - ajouter
    - à une boîte de dialogue 268
    - commandes 312
    - commandes surgissantes 312
    - instructions 311
    - raccourcis 315
    - séparateurs 312
  - astuces et réserves 478
  - clignoter 330
  - commandes 307
    - lier aux raccourcis 329
    - tabulations 315
  - copier des instructions 315
  - créer 308
  - déboguer 317
  - déplacer des instructions 315
  - déroulants 308
  - désactiver des commandes 313
  - éditer 311
    - éléments de menu 312
      - script ressource 318
    - Voir aussi* éléments de menu
  - en cascade 307
  - exemple 319
  - flottants 308
    - créer 316
    - tester 310
  - griser des commandes 313
  - Help Break dans barre menu 313
  - identificateurs 315
  - instructions 310
    - supprimer 316
  - inversion vidéo 330
  - options mémoire 318
  - rechercher les identificateurs d'élément en double 317
  - script ressource 318
  - séparateurs 308
  - souligner des caractères 314
  - supprimer des commandes surgissantes 316

- surgissants 308
- tester 309, 317
  - affichage surgissant 310
- touches de raccourci 323
- menus flottants
  - créer 316
- menus locaux
  - ouverture 9
- Message (fenêtre)
  - afficher les erreurs et les avertissements 119
- Messages | General 90
- Messages | Inefficient C++ Coding 89
- Messages | Inefficient Coding 89
- Messages | Obsolete C++ 88
- Messages | Potential C++ Errors 88 - 3-89
- messages d'erreur 403
  - définition 404
  - erreur fatale 403
  - dans les tables de chaînes 335
- messages d'invite, dans les tables de chaînes 335
- messages de mise en garde 403
  - définition 404
- métafichiers 395
- Minimize Box (style), boîtes de dialogue 267
- mise en couleur du texte 14
- mise en surbrillance de la syntaxe
  - ajout de fichiers 14
  - désactivation 14
  - modification 14
  - utilisation 14
- mise en surbrillance du texte 14
- mises en garde
  - Voir aussi* avertissements
  - générales 90
- Modal Frame (style), boîtes de dialogue 267
- mode
  - hard 139
  - soft 139
- mode de sélection 273
- mode pas à pas
  - définition 122
- modèle Tiny
  - fichiers .COM et 180
- modèles
  - génération 80

- modèles de mémoire
  - options en ligne de commande 70
- modes (éditeur Dialog) 270
  - sélectionner 273
  - test 301
- modifier
  - point d'arrêt 136
  - propriétés d'un point d'arrêt 135
  - valeur des éléments de données 131
  - valeurs de variables 130
  - valeur d'un identificateur 251
- Modifiers (option), menus 314
- motifs
  - outil Aérographe 350
  - outil Pinceau 350
  - outils de cadre plein 353
  - sélectionner 363
- mots clés
  - Borland C++ 66
  - d'extension ANSI 66
  - Kernighan et Ritchie, utilisation 66
  - UNIX, utilisation 66
- Move resource (boîte de dialogue) 247
- Moveable (option mémoire) 249
- Multi Column (option de style List Box) 291
- Multi-Save (boîte de dialogue File Preferences) 258, 260
- Multiple Select (option de style List Box) 291
- mx (options)
  - modèles de mémoire 70

## N

- n (option BCC.EXE) 178, 210
  - logique de débordement de pile 67
  - répertoires .OBJ et .ASM 60
- New (commande), Resource (menu) 245
- New button (boîte de dialogueIdentifiers) 255
- New Edit Pop-up (commande), éditeur Menu 311
- New File Pop-up (commande), éditeur Menu 311
- New Help Pop-up (commande), éditeur Menu 311
- New Image (commande) 373
- New Menu Item (commande), éditeur Menu 311
- New Pop-up (commande), éditeur Menu 311
- New Project (boîte de dialogue) 239
- New Project (commande) 239

New Resource Type (boîte de dialogue) 396  
New Separator (commande), éditeur Menu 311  
NMAKE (Microsoft)  
remplacé par MAKE 210  
No Border (style de cadre), boîtes de dialogue 266  
No Break (option), menus 313  
No Idle Messages (style), boîtes de dialogue 267  
noeuds

ajout par "tirer et lâcher" 34  
dépendants 28  
police maigre (copie) 36  
.noIgnore (directive MAKE) 223  
NOKEEP (option MAKE) 212  
nombre de passages  
à un point d'arrêt 136  
NomDeMacro 227  
noms non significatifs  
DLL 200  
Normal (option d'affichage) 304  
.nosilent (directive MAKE) 223  
.noswap (directive MAKE) 223  
Not Owner Draw (option Owner Drawing) 290  
NOT WS\_VISIBLE (style) 279  
Notify (option de style List Box) 291  
numéros de ligne  
dans la barre d'état 13

## O

-o (option BCC.EXE) 178  
fichiers objet 91  
.OBJ (fichiers)  
BBC 48  
compilation 91  
numéros de ligne 67  
répertoires 60  
ObjectWindows  
applications AppExpert 95  
classes, visualisation 107  
création d'applications 96  
objets  
hiérarchie, visualisation 17  
Browser, dans l'EDI Windows 17  
voir les détails 18  
octal (format), spécificateurs 399  
OEM (jeu de caractères) 294, 299

ombre portée 376 - 20-377  
Open Project (boîte de dialogue) 240  
Open Project (commande) 240  
optimisations  
EDI 81  
en-têtes précompilés 482  
options de compilateur en ligne de commande 81  
registres, utilisation 64  
Option Browser reference information in OBJs 16  
Option Debug information in OBJs 16  
Option Defpushbutton (BWCC) 473  
option owner-draw (BWCC) 472  
options  
BCC.EXE 47  
compatibilité ascendante 93  
compilateur, table 50 - 3-60  
éditeur de l'EDI 13  
génération de modèles C++, option en ligne de  
commande 80  
insertion dans les fichiers 49  
priorité 49  
projet, modification 30  
TLINK, tableau 173  
Options | Save 12  
options de compilateur  
règles 49  
options de couleur  
Aérographe 350  
Boîte de peinture 350  
Gomme 349  
Ligne 351  
Pinceau 350  
Stylo 349  
Texte 352  
cadre vide 352  
cadre plein 353  
options en ligne de commande  
insertion dans les fichiers 49  
options MAKE  
aide 210  
liste 210  
-N (compatibilité NMAKE) 210  
utilisation 210  
Options Settings (boîte de dialogue) 10  
options TLINK



- tableau 173
- ordre d'évaluation
  - options de compilateur en ligne 49
- outil texte
  - options de couleur 352
- Outil Texte static Borland 471
- outils
  - éditeur Bitmap 348
  - éditeur Bitmap, sélectionner 346
  - éditeur Dialog 270
- Outline (volet), éditeur Accelerator 326
- Outline (volet), éditeur Menu 310
- ouvrir un projet 240
- Overlapped
  - style de fenêtre 266
- Owner Draw (contrôle bouton 288
- Owner Drawing (options)
  - boîtes à options combinées 298
  - boîtes liste 290 - 15-291

## P

- p (option BCC.EXE) 179, 211
  - compilations C++ et C 92
  - conventions d'appel Pascal 69, 77
- p- (option BCC.EXE)
  - conventions `__stdcall` 69
- pages, alignement 176
- palette Colors 356
  - afficher 359
  - cache 359
  - index 360
  - montrer 359
- palettes
  - Colors, index 360
  - couleurs 356
  - Tools (éditeur Bitmap) 346
  - Tools (éditeur Dialog) 270
- palettes couleur, personnalisées 367
- palettes couleur, personnaliser 360
- paramètre d'alignement de mot 68, 76
- paramètres
  - alignement de mot 68, 76
- pas à pas
  - définition 122
- `__pascal` 69, 77

- option en ligne de commande 69, 77
- Pass Keyboard Input (option de style List Box) 291
- passages
  - nombre 136
- Password (option), boîte de dialogue Edit Text Style 294
- Paste (commande) 247
- `.path.ext` (directive MAKE) 224
  - description 227
- Pattern (commande) 364
- Pause Program (commande) 126
- pc (option BCC.EXE)
  - conventions C 69, 77
- Pen Style (boîte de dialogue) 364
- Pen Style (commande) 364
- pile
  - cadrere standard, génération 66
  - débordement 67
- pilotes vidéo, couleur 356
- Pinceau 350
  - forme de pinceau, choisir 363
  - motifs 363
  - options de couleur 350
- pixels 346
  - unité de mesure dans l'éditeur Dialog 304
- po (option BCC.EXE)
  - convention d'appel `fastcall` 63
- point actif
  - définir 382
- point d'exécution 122
  - personnaliser l'affichage 137
  - saut au 125
- pointeur d'instruction 125
- pointeurs
  - conversion suspecte 87
  - fast huge 72
  - huge 72
  - table virtuelle, 32 bits 71
    - compatibilité 79
- pointeurs membres
  - contrôle 77
- points d'arrêt 133
  - activer 134
  - conditionnels 136
  - Voir aussi* débogage, expressions de suivi

- définition 133
- désactiver 134
- effacer 133
- incorrects 135
- modifier 136
- modifier les propriétés 135
- personnaliser l'affichage 137
- poser 133
  - après le lancement du programme 136
  - dans un fichier non ouvert 136
- supprimer 134
- polices 237, 385
  - Attributes (options) 393
  - attributs, définir 393
  - bitmap, définir la largeur 391
  - caractères
    - définir la largeur 391
    - ajouter 390
  - contenu de l'en-tête 392
  - correspondance avec les jeux de caractères 390
  - créer 387
  - définition 234, 385
  - description 392
  - éditer 389
  - fichier .FNT autonome 388
  - format binaire 387
  - format source 387
  - graphiques par points, multiples, stocker 386
  - image, définir la largeur 391
  - informations de copyright 392
  - jeux de caractères, définir 389
  - largeur fixe 389
  - largeur variable 389
    - créer 391
  - noeuds et 36
  - nombre de caractères, définir 390
  - options
    - Average Width 390
    - Break 391
    - Copyright 393
    - Default 391
    - Device 393
    - Face Name 393
    - Height 390
    - Last 391
    - Stretch Current Chars 390
    - Character 391
    - spécifier pour une boîte de dialogue 268
    - taille, définir 389
    - tramées 386
    - utilisation dans les applications 394
    - vectérielles 386
- POPUP (instruction)
  - supprimer 316
- Popup, style de fenêtre 266
- portage
  - options 66
- portée des éléments de données affichés 131
- poser un point d'arrêt 133
  - après le début de l'exécution 136
- pr (option BCC.EXE)
  - convention d'appel `_ _fastcall` 69, 77
- #pragma (options)
  - warn 87
- #pragma hdrfile 480 - C-481
- #pragma hdrstop 480, 483
- pragma intrinsic 84
- .precious (directive MAKE)
  - description 224, 227
- préférences
  - définition 12
  - menu File 258
    - Multi-Save 260
- Preferences (boîte de dialogue, éditeur Dialog) 304
- Preferences (commande), éditeur Dialog 304
- Preferences (commande), menu File) 258
- Preferences (éditeur Dialog)
  - Draw Custom Controls as Frames 305
  - Drawing Type 304 - 15-305
  - Generate CONTROL Statements Only 305
  - options Selection 305
  - Selection Border 304
  - unités de la ligne d'état 304
- Preload (option mémoire) 249
- premier plan (couleurs) 357
- presse-papiers
  - copier une ressource 247
- priorité
  - options de compilateur en ligne 49
- problèmes de remplissage, images bitmaps 351

- processeurs 80x86
  - instructions 68
  - instructions 32 bits 76
- programmes 16 bits DPMI
  - incidents 21
- projets 239
  - ajouter une ressource 244
  - copier une ressource entre 247
  - créer 239
  - enregistrer 302
  - mise en surbrillance de la syntaxe 14
  - options, modification 30
  - ouvrir 240
  - polices et noeuds 36
  - ressources définies par l'utilisateur, ajouter 397
  - ressources intégrées, ajouter 245
  - ressources liées, ajouter 245
  - types de fichier, choisir 239
- projets d'exemple
  - icône 375
  - raccourcis 331
  - tables de chaînes 341

## Q

- q (option MAKE) 211

## R

- r (option BCC) 211
  - addition d'informations browser 67
  - et BUILTINS.MAK 208
  - variables de registre 64
  - et DLL 139
- raccourcis 234, 323
  - ajouter 315
  - commandes 314
  - créer 325
  - déboguer 330
  - définition 323
  - éditer 328
  - exemple 331
  - fonction Flash 330
  - identificateurs 326
  - rechercher les identificateurs en double 330
  - sélectionner 328

- tabulations 315
- touches ASCII 324
- touches virtuelles 324
- Windows prédéfinis 329
- raccourcis clavier
  - configuration 13
- RC 191
- RC\_INVOKED 193
- RCDATA (type de ressource) 401
- rcinclude (référence) 302
- s-rd (option de compilateur) 64
- Read Only (option), boîte de dialogue Edit Text Style 294
- recherche
  - automatique d'erreur 15
  - de bibliothèques 61
  - de fichiers include 61
  - d'une fonction 125
- recréation de bibliothèques 65
- Rectangle
  - cadre plein 353
    - couleurs 353
    - motifs 353, 363
  - cadre vide 352
    - couleurs 352
    - styles de ligne 364
- rectangle de sélection 273
  - outil 348
- rectangle noir (contrôle boîte de dialogue) 273
- redimensionner des zones sélectionnées (éditeur Bitmap) 365
- redimensionner une boîte de dialogue 265
- Redo Levels (boîte de dialogue File Preferences) 258
- Register
  - convention d'appel 77
  - conventions d'appel 63
- Register Keyword
  - option de compilateur 64
- Registers (fenêtre) 132
- registres 132
  - variables
    - bascule 64
    - explicités 64
    - mot clé 64

- suppression 64
- Relative (option de grille), boîtes de dialogue) 277
- relief
  - boîtes de dialogue avec Resource Workshop 306
- remplacer (action TLIB) 206
- Rename (commande) 248
- renommer
  - ressources 248
- Repeat Count (option) 128
- répertoires
  - bibliothèques 61
    - option en ligne de commande 60
  - ASM et .OBJ, options en ligne de commande 60
  - fichiers include 60
- Resource Workshop
  - à partir de l'EDI 114
  - AppExpert et 114
  - ClassExpert et 115
  - compilateur de ressource Microsoft, incompatibilité 279
  - configurer 258
  - modifications avec AppExpert et 115
  - préférences 258
- Resource | Move 247
- ressource RC
  - définition 235
- ressource utilisateur
  - définition 235
- ressource VERSIONINFO
  - définition 235
- ressources 192, 239
  - ajouter au projet 244
  - charger 243
  - compiler 192, 236
  - copier entre projets 247
  - créer, nouveaux types 396
  - décompiler 236
  - définies par l'utilisateur 395
    - ajouter au projet 397
    - éditer 397
    - métafichiers 395
  - définition 233
  - déplacer 247
  - éditer dans les fichiers exécutables et DLL 260
  - éditeurs 235
  - édition de liens 193
  - enregistrer, boîte de dialogue File Preferences 260
  - identificateurs 250
    - intégrées 241
    - intégrées au projet 245
    - liées 241
    - lier 236
    - lier au projet 245
    - menus 307
    - noms par défaut 250
    - options d'affichage 242
    - options mémoire 248
    - polices et graphiques par points 386
    - raccourcis 323
    - RCDATA 401
    - renommer 248
    - sélectionner 243
    - types 233
    - types de fichiers 235
  - ressources bitmap
    - mémoire 387
    - polices et graphiques par points 386
    - types 345
  - ressources définies par l'utilisateur
    - chaînes, terminées par un zéro 400
    - données, ajouter 398
    - formats de données 399
    - identificateurs 396
    - valeurs hexadécimales 400
  - rétablir 258, 287
  - RGB (valeurs) 359
    - définir 360
    - ligne d'état 355
  - RLINK 193
    - apel 193
  - RLINK.DLL
    - défini 167
  - RLINK.EXE
    - défini 167
  - .RTF (fichiers)
    - création 96
  - RTM (variable d'environnement) 22
  - Run Arguments (option) 121
  - Run to Cursor
    - commande de menu 125

- run-time, gestionnaire 22
  - contrôle de la mémoire utilisée 22
- rupture de ligne (texte statique) 295
- RWS\_Icon (section), WORKSHOP.INI 351
- RWS\_OwnFloodfill (WORKSHOP.INI) 351

## S

- S (option BCC.EXE) 180, 211
  - produit un fichier .ASM mais n'effectue pas l'assemblage 92
- Save with Default Device Colors (option) 361, 367
- scripts ressource
  - commentaires 244
  - contrôles dialogue 305
  - fichiers .RC 192
  - menus 318
  - ressources définies par l'utilisateur, créer 399
  - spécifications de format 339
  - table de chaînes 340
- Scroll Bar (option Style) 278
- Scroll Bar Always (option de style List Box) 292
- Scroll Bar Style (boîte de dialogue) 289
- segment de table virtuelle far
  - nom et changement de nom 73
- segments
  - contrôle 72
  - non initialisés 177
- segments (tables de chaînes) 338
- segments de code
  - enregistrement de tables virtuelles 71
  - groupe 74
  - nom et changement de nom 72, 74
  - réduction 179
  - stockage 179
- segments de données
  - groupe 73
  - nom et changement de nom 72 - 3-73
  - suppression de tables virtuelles 71
- segments de fin WORD
  - non initialisés 177
- segments de pile
  - segment de données et (bouton Advanced de TargetExpert) 28
- segments, mappe de
  - champ ACBP et 179

- TLINK et 179
- Select All (commande)
  - éditeur Dialog 274
- Select All (commande), éditeur Bitmap 348
- Select Font (boîte de dialogue) 268
- sélecteur (outil) 273
- Selection (options) 305
- sélectionner
  - des contrôles 273
  - des éléments de menu 311
  - toute l'image 348
  - une boîte de dialogue 265
- sémantique
  - erreurs (définition) 120
- séparateurs (menus) 308
  - ajouter 312
- séquences d'échappement, type C (éditeur String) 339
- Set Groups (commande) 281
- Set Groups (outil) 281
- Set Hot Spot (commande) 383
- Set Order (commande) 282
- Set Order (outil) 281
- Set Tabs (commande) 280
- SHARE (DOS)
  - et MAKE 211
- Show Identifiers (commande) 242
- Show Items (commande) 242
- Show Palette (commande) 359
- Show Resources (commande) 242
- Show Unused Types (commande) 243
- ShowWindow (fonction) 279
- .silent (directive MAKE) 224
- Simple Text (type de contrôle statique) 296
- Size (commande), éditeur Dialog 265, 269, 276, 285
- Size and Attributes (commande) 356
  - éditeur Icon 374
- Size Controls (boîte de dialogue) 285
  - option Enter Values 287
- soft (mode de débogage) 139
- Sort (option de style List Box) 291
- sorties
  - vérification pour un programme 122
- souligner les caractères

- dans les menus 314
- souris
  - bouton droit (éditeur Bitmap) 346
  - d'expression 129
- spécificateurs de format
  - hexadécimal 339, 399
  - octal 399
- Split Horizontal (commande) 354
- Split Vertical (command) 354
- Standard stack frame
  - commande 66
- Static Style (boîte de dialogue) 295
- statique en forme d'icône (contrôle boîte de dialogue) 272
- Step Over (commande de menu) 122
- structures
  - indéfinies 87
  - longueur de zéro 87
  - violations ANSI 87
- style (boîtes de dialogue), contrôles 277
- styles de boîte de dialogue 266
- styles de boîte de dialogue (boîte de dialogue Window Style) 266
- styles de cadre (boîte de dialogue Window Style) 266
- styles de ligne (éditeur Bitmap) 364
- Stylo
  - styles de ligne 364
  - options de couleur 349
- suivi d'expressions
  - ajouter 127
- sujets
  - choix dans les boîtes de dialogue 10
- supprimer
  - action TLIB 206
  - expressions de suivi 129
  - identificateurs 256
  - point d'arrêt 133 - 6-134
- .swap (directive MAKE) 224
- symboles
  - affichage des déclarations 18
  - Visualisation dans le code source 20
- syntaxe
  - erreurs (définition) 119
  - TLINK 167

- System Menu (style), boîtes de dialogue 266
- System Modal (style), boîtes de dialogue 267

## T

- T- (option BCC.EXE) 180 - 9-181
  - suppression des options assembleur 92
- Tab Set (outil) 280
- Tab Stop (attribut de contrôle) 279
- Tab Stops (option de style List Box) 291
- table de liaison intégrale 91
- table des symboles 121
  - passer de l'une à l'autre 139
- tables
  - options de compilateur 50 - 3-60
- tables de chaînes 335
  - astuces et réserves 478
  - définition 234
  - déplacement 340
  - écran de l'éditeur String 337
  - éditer 340
  - identificateurs 338
  - modifier 340
  - nommer 343
  - projet exemple 341
  - script ressource 340
  - segments 338
  - utilisation de la mémoire 338
  - valeurs d'identificateur 337
  - valeurs hexadécimales 339
- tables de raccourcis
  - éditer 328
- tables virtuelles
  - contrôle 79
  - enregistrement dans le segment de code 71
  - externes
    - option en ligne de commande 79
  - locales
    - option en ligne de commande 79
  - option en ligne de commande 79
  - pointeurs 32 bits 71
  - smart
    - option en ligne de commande 79
- tabulation
  - dans les menus 315
- taille de seuil

- variables globales far, définition] 71
- taquets de tabulation, contrôles de boîte de dialogue 280
- Target Windows Version (boîte de dialogue File Preferences) 259
- TargetExpert
  - utilisation 33
- Tchaîne (option BCC.EXE)
  - passage de chaîne à l'assembleur 92
- Test Dialog (commande) 301
- Test Menu (fenêtre), changer l'affichage 310
- Test Menu (fenêtre), menu surgissant, afficher 310
- Test Menu (volet), éditeur Menu 309
- tester
  - boîtes de dialogue 301
  - menus 317
- Text Editor (boîte de dialogue FilePreferences) 258
- texte
  - chaînes, terminées par un zéro 400
  - couleur 14
  - éditeur Bitmap 352, 361
  - images bitmap 361, 352
  - mise en surbrillance de la syntaxe (couleurs) 14
- texte de message
  - éditer 335
  - traduire 335
- texte statique 272
- texte statique (contrôle boîte de dialogue) 272
- Thick Frame (style), boîtes de dialogue 267
- 'this' (pointeur) des fonctions membres 'pascal' 78
- tilde (~), symbole de fichier de sauvegarde 259
- TLIB
  - erreurs 403
- TLINK (éditeur de liens)
  - bibliothèques 171
    - mathématiques et 172
  - champ ACBP et 179
  - compilateur de lignes de commande et 171
  - contrôle de mise en garde 181
  - démarrage 167
  - écrasement 178
  - erreurs 403
    - de startupfile 168
  - fichier de configuration 169
  - exemple 169
- fichiers de recherche 170
- informations de débogage 181
- informations de débogage dans les fichiers .OBJ et 177
- mappe de fichier exécutable généré par 178
- modules non-C et 178
- options 173
  - alignement des pages (/A) 176
  - application .COM (/t) 180
  - bibliothèques, ignorer (/n) 178
  - contrôle de mise en garde (/w) 181
  - dictionnaires étendus (/E) 177
  - dictionnaires étendus (/f) 177
  - différence majuscules/minuscules (/C) 176
  - DLL (/Twe) 181
  - exécutable Windows (/Tw) 181
  - extensions de fichiers 169
  - fichiers application (/Tw) 181
  - fichiers de mappe (/m) 178
  - fichiers exécutables (/Tw) 181
  - fichiers mappe (/m), segments des 179
  - /Gx ("Goodies") 177
  - /i (segments de fin non initialisés) 177
  - ignorer les dictionnaires étendus (/e) 177
  - informations de débogage (/v) 181
  - /l (numéros de lignes du code source) 177
  - mémoire étendue (/yx) 182
  - mémoire paginée(/ye) 182
  - /n (ignorer les bibliothèques par défaut) 178
  - /o (écrasement) 178
  - option application (/Td) 180
  - segments de fin non initialisés (/i) 177
  - stockage de segments de code (/P) 179
  - taille de la pile (/S) 180
  - /Tw (fichiers application) 181
  - type d'application (/a) 175
  - /v (informations de débogage) 181
  - /w (contrôles de mise en garde) 181
  - /x 182
  - /ye (mémoire paginée) 182
  - /yx (mémoire étendue) 182
- syntaxe 167
- TLINK.CFG 169
  - exemple 169
- TLINK32

- TLINK32 (éditeur de liens)
  - bibliothèques 172
  - fichier de configuration 169
  - options
    - adresse de base (/B) 176
    - majuscules/minuscules (/c) 176
    - nombre maximum d'erreurs (/Enn) 177
    - options application (/Tp) 181
  - RLINK et 167
- TLINK32.CFG 169
- Tools (palette)
  - éditeur Dialog 270
  - éditeur Bitmap 346
- TOUCH.EXE
  - description 209
- touches
  - définir comme raccourcis 323
  - éditeur de texte 399
- Tracé des messages 143
- Trace Into (commande de menu) 124
- TrackPopupMenu (fonction) 308
- traducteurs
  - dérogation aux valeurs par défaut 33
  - installation 42
- traduire un texte de message 335
- tramées (polices) 386
- transtyper les identificateurs de ressource 251
- Turbo Assembler
  - défaut 91
- TURBOC.CFG
  - description 48
- TurboMenus
  - utilisation 9
- tW (option BCC.EXE)
  - compilation avec toutes les fonctions far exportables 74
- tWD (option BCC.EXE)
  - compilation sous forme de DLL, toutes fonctions exportables 75
- tWDE (option BCC.EXE)
  - compilation sous forme de DLL, avec fonctions explicit \_export exportables 76
- tWE (option BCC.EXE)
  - compilation avec fonctions explicit \_export exportables 74

- tWS (option BCC.EXE)
  - compilation avec rappels smart 75
- tWSE (option BCC.EXE)
  - compilation avec rappels smart 75
- type d'élément (menus) 313
- types de données
  - char 63
  - hexstring 399
- types de fichier
  - .CUR 381
  - .DLG 302
  - .FNT 388
  - .ICO 371
  - choisir 239
  - ressource
    - choisir 239

## U

- u (option BCC.EXE) 210
  - annulation 93
  - caractères de soulignement 65
- !undef (directive MAKE) 224
  - description 228
- Undo (outil), éditeur Dialog 287
- Undo Levels (boîte de dialogue FilePreferences) 258
- unités dialogue 304
  - définition 265
- UNIX
  - conformité de langage 66
  - portage de fichiers 66
- User Button (contrôle) 288

## V

- v (option BCC.EXE) 79, 181
  - compatibilité d'arguments de classes 78
  - compatibilité de pointeur 'this' dans les fonctions membres 'pascal' 78
  - compatibilité de pointeur de classe de base virtuelle 78
  - informations de débogage 67
  - pointeurs de table virtuelle 79
  - tables virtuelles C++ 79
- s-Vc (option BCC) 78



- valeurs
  - visualiser 126
- valeurs de registres 132
- valeurs hexadécimales
  - ressources définies par l'utilisateur 400
  - tables de chaînes 339
- valeurs numériques, notation, ressources définies par l'utilisateur 399
- Variable (option Owner Drawing) 291
- variable, largeur de police 389
- variables
  - alignement automatique de mot 68, 76
  - communes 65
  - globales, far 71
  - modifier la valeur 130
  - registres 64
  - suivi 126
- variables communes 65
- variables d'environnement
  - avec MAKE 210
  - et DPMI 21
- variables d'instance
  - ajout (ClassExpert) 113
  - suppression (ClassExpert) 113
- variables far 71
- variables globales
  - alignement de mot 68, 76
- vectérielles (polices) 386
- VERSIONINFO
  - définition 235
- Vertical Scroll (style), boîtes de dialogue 267
- vi (option BCC.EXE)
  - fontions en ligne C++ 67
- View (menu)
  - éditeur Menu 310
- View as Pop-up (commande), éditeur Menu 310, 316
- View Source (commande) 135
- virgule flottante
  - bibliothèque mathématique et 172
  - bibliothèques 64
  - option d'affichage 128
  - rapide 64
- virtuel (touches de raccourci) 324
- Visible (attribut de contrôle) 279
- Visible (style), boîtes de dialogue 268
- visualisation
  - dans l'EDI Windows, symboles dans le code 20
  - de la hiérarchie d'un objet 17
  - DE code source 135
- visuel
  - génération d'applications 95
- vitesse optimisation 81
- Vm (options BCC.EXE)
  - pointeurs membres C++ 77
- volet de fenêtre actif (éditeur Bitmap) 354
- volets de fenêtre
  - éditeur Bitmap 354

## W

- wxxx (options BCC.EXE) 181, 210
  - avertissements 87
  - console monothread NT 92
  - définir les options par défaut 211
- Want Return (option), boîte de dialogue Edit Text Style 294
- warn pragma 87
- Watch (commande) 127
- Watch (fenêtre) 127
- Watch Properties (boîte de dialogue) 127
- WCDE- (option BCC32.EXE)
  - DLL de console monothread NT 93
- WD (option BCC.EXE)
  - compilation sous forme de DLL, toutes fonctions exportables 75
- WCD- (option BCC32.EXE)
  - DLL de console monothread NT 93
- WDE (option BCC.EXE)
  - compilation sous forme de DLL, avec fonctions explicit \_export exportables 76
- WE (option BCC.EXE)
  - compilation avec fonctions explicit \_export exportables 74
- White Frame (type de contrôle statique) 296
- White Rectangle (type de contrôle statique) 296
- Width (option), polices 390
- WIN.INI
  - instruction "LOAD=" 477
  - modification avec BWCC 476
- Win32 261

Window Style (boîte de dialogue) 265  
   bouton Fonts 268  
   Caption (style de cadre) 266  
   styles de boîte de dialogue 266  
   styles de cadres 266  
 Windows  
   éditeur Notepad 244  
   raccourcis prédéfinis 329  
   valeurs d'identificateurs de table de chaînes 338  
   Presse-papiers  
     copier une ressource 247  
 Windows NT 261  
 WINDOWS.H 324  
 WinSight  
   aspect 141, 143  
   démarrage 141  
   présentation 141  
 WINSPECTR.BIN 162  
 WINSPECTR.LOG 162  
 WinSpector  
   Démarrage 153  
   DFA (traitement de données) 162  
   fichier .MAP 164  
   file d'attente des messages 159  
   iInformations système 161  
   Modules 160  
   registres 158  
   tâches 159  
   tas USER et GDI 160  
   WINSPECTR.LOG 154  
 -WM- (option BCC32.EXE)  
   DLL de console monothread NT 93  
 WM\_COMMAND (message) 323  
 WM\_COMPAREITEM (message) 290  
 WM\_CTLCOLOR (message) 296  
 WM\_DELETEITEM (message) 290  
 WM\_DRAWITEM 288, 290, 298  
 WM\_ENTERIDLE (message) 267  
 WM\_MEASUREITEM (message) 290, 298  
 WM\_SYSCOMMAND (message) 323  
 WORKSHOP.INI  
   fixer les problèmes de remplissage 351  
 -WS (option BCC.EXE)  
   compilation avec rappels smart 75  
 WS\_OVERLAPPED (style) 269  
 WS\_VISIBLE (style) 279  
 -WSE (option BCC.EXE)  
   compilation avec rappels smart 75  
 WYSIWYG (option d'affichage) 305

## X

-X (option BCC.EXE)  
   désactivation des informations  
   d'auto-dépendance 65  
   gestion d'exceptions 80

## Y

-y (option BCC.EXE)  
   numéros de ligne 67  
 /ye option TLINK (mémoire paginée) 182  
 /yx option TLINK (mémoire étendue) 182

## Z

zone client (boîtes de dialogue) 267  
 zoom (outil) 348  
 Zoom In (commande) 348  
 Zoom Out (commande) 348  
 -zV (options BCC.EXE)  
   segments de table virtuelle far 73