



Ressourcesinformatiques

WinDev

Les fondamentaux du développement d'applications

OUVRAGE AGRÉÉ

PCSOFT® 

David VANDEVELDE

Fichiers complémentaires
à télécharger



Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RIWIND** dans la zone de recherche
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Introduction

1. Présentation	17
2. Objectifs de cet ouvrage	19
3. Description de l'atelier	19

Chapitre 1 Installation

1. Prérequis matériels	21
2. Procédure d'installation	21
2.1 Première installation	21
2.2 Mise à jour automatique	33

Chapitre 2 L'environnement de développement

1. Introduction	35
2. L'interface utilisateur de WinDev	36
2.1 Principe général	36
2.2 Les outils d'ergonomie	39
2.2.1 Options générales de Windev	40
2.2.2 Options de l'environnement	41
2.2.3 Options d'édition	41
2.2.4 La barre standard	43
2.2.5 Le tableau de bord	43
2.2.6 Contrôle des volets	44
2.2.7 La barre contextuelle	45
2.2.8 La barre de composants	45
2.2.9 La barre d'alignement	45
2.2.10 La barre de cadre	45

2.2.11	La barre de documents ouverts	45
2.2.12	La barre du gestionnaire de sources (GDS)	46
2.2.13	Les volets WinDev	47
2.2.14	Le Navigateur	48
2.2.15	Le Modifieur	48
2.2.16	Les volets d'assistance.	50
2.2.17	La punaise	52
2.2.18	La Perso-Note	53
3.	Lexique des éditeurs.	54
3.1	L'éditeur de projet	54
3.2	La méthodologie de conception	54
4.	Projet.	57
4.1	Description	57
4.2	Perso-dossier	58
5.	Données	59
5.1	Modèle conceptuel de données.	60
5.1.1	Rappel sur MERISE.	60
5.1.2	Attribut	60
5.1.3	Entité	60
5.1.4	Relation	60
5.2	Éditeur d'analyses	62
5.2.1	Création de fichier	64
5.2.2	Création de rubrique.	65
5.2.3	Les métatypes	66
5.2.4	Import d'une description	67
5.2.5	Création de relation	68
5.2.6	Journalisation	69
5.2.7	Abonnement	70
5.2.8	Connexion à une base de données	70
5.2.9	Génération.	71
6.	Fenêtres.	72
6.1	Description de l'éditeur de fenêtres.	72
6.2	Positionnement automatique.	75
6.3	Vue en trois dimensions	75

7.	Requêtes	77
7.1	Création d'une requête	78
7.2	Génération du code SQL	79
7.3	Exécution d'une requête SQL	79
7.4	Outils annexes	79
7.4.1	Interrogateur SQL	79
7.4.2	Éditeur utilisateur	80
8.	États	80
9.	Code	83
9.1	Assistance et guide lors de la saisie du code	85
9.2	Enroulement et déroulement du code	86
9.3	Vérification de la syntaxe et compilation	86
9.4	Événements spécifiques complémentaires	87
9.5	Assistance directe	87
9.6	Traduction des messages	89
9.7	Aide au test (débogage)	89
9.8	Historique	90
10.	Atelier	91
10.1	Base GDS	91
10.2	Règles métier	92
10.3	Éditeur d'exigences	93
11.	Architecture	94
11.1	Éditeur UML	94
11.1.1	Rappel sur UML	94
11.1.2	Cas d'utilisation	94
11.1.3	Classe	95
11.1.4	Génération	98
11.1.5	Documentation	98
11.2	Éditeur de modélisation souple	99
11.3	Éditeur d'aide	100
11.4	Éditeur de documents	102
12.	Automatisation	102
12.1	Batches	102
12.2	Plan d'action	102

Chapitre 3**Les éléments d'une application WinDev**

1. Introduction	103
2. Description d'un objet dans WinDev	103
2.1 Description générale	103
2.2 Onglets communs	105
2.2.1 Onglet Note	105
2.2.2 Onglet Aide	106
2.2.3 Onglet Style	107
3. Le projet (application)	109
3.1 Description d'un projet	110
3.2 Code du projet	113
4. Les gabarits et les styles	114
5. Les fichiers, tables et rubriques	115
5.1 Les fichiers ou tables	116
5.2 Les rubriques	119
5.2.1 Les onglets généraux	120
5.2.2 Les onglets complémentaires	121
5.2.3 Les métatypes	121
5.2.4 L'index full-text	121
5.3 Les liaisons entre fichiers	122
6. Les fenêtres	123
6.1 Mode d'interface des fenêtres : MDI, modal ou non modal	124
6.2 Description	124
6.3 Fenêtres générées par les assistants	128
6.3.1 Les fenêtres Fiche	130
6.3.2 Les fenêtres Table	134
6.3.3 Les fenêtres Zone répétée	138
6.4 Modèles utilisés par le RAD	138
6.5 Plans, onglets et groupes de champs	140
6.5.1 Les plans	140
6.5.2 Les onglets	141
6.5.3 Les groupes de champs	141
6.6 Fenêtre interne et champ interne	142
6.7 Macro-code utilisateur (MCU)	143

6.8	Effets automatiques	144
7.	Modèle de fenêtres	145
8.	Les requêtes	146
9.	Les champs	149
9.1	Description générale	151
9.2	Menu contextuel	153
9.3	Les champs classiques	153
9.3.1	Champ de saisie	153
9.3.2	Libellé	154
9.3.3	Bouton	154
9.3.4	Image	155
9.3.5	Sélecteur	156
9.3.6	Interrupteur	156
9.3.7	Liste	157
9.3.8	Liste Image	157
9.3.9	Combo	158
9.3.10	Combo avec table	158
9.3.11	Table	159
9.3.12	Table Hiérarchique	163
9.3.13	Zone répétée	163
9.3.14	Zone Multiligne	164
9.3.15	Arbre	164
9.3.16	Séparateur	166
9.3.17	Onglet	166
9.3.18	Onglet dynamique	167
9.4	Les champs spécialisés	167
9.4.1	Carrousel	167
9.4.2	Champ Cube	167
9.4.3	Note repositionnable	168
9.4.4	Champ Fil	169
9.4.5	Tableau croisé dynamique	170
9.4.6	Jauge	172
9.4.7	Jauge infinie	172
9.4.8	Champ Notation	173
9.4.9	Ascenseur	173
9.4.10	Range Slider	174

9.4.11	Potentiomètre	174
9.4.12	Potentiomètre rotatif	174
9.4.13	Spin	175
9.4.14	Champ Tiroir	175
9.4.15	Boîte à outils	175
9.4.16	Graphe	176
9.4.17	Champ Code-barres	177
9.4.18	Champ ActiveX	177
9.4.19	Champ OLE	178
9.4.20	Champ Html	178
9.4.21	Champ Caméra	178
9.4.22	Champ Multimédia	179
9.4.23	Champ Conférence	179
9.4.24	Champ .NET 2.0 (WinForm)	179
9.4.25	Champ .NET 3.0, 4.0... (WPF)	179
9.4.26	Champ Calendrier	179
9.4.27	Champ Agenda	179
9.4.28	Champ Planning	180
9.4.29	Champ TimeLine	184
9.4.30	Champ Organigramme	184
9.4.31	Champ TreeMap	185
9.4.32	Champ Tableau de Bord	185
9.4.33	Champ Tableur	186
9.4.34	Diagramme de Gantt	187
9.4.35	Champ Forme	188
9.5	La création de champs	188
9.6	Le superchamp	189
9.7	Barre d'outils	190
10	Les états	190
10.1	Modèle d'état	191
10.2	Description	191
10.3	Champs d'un état	199
10.4	Aide à la mise en page	200

Chapitre 4
Assistance au développement

- 1. Introduction 201
- 2. Aide à l'écriture 203
 - 2.1 Documentation imprimée 203
 - 2.2 Aide au développement (AAD) 206
 - 2.3 Fonctionnalités automatiques de l'application (FAA) 207
 - 2.4 Actions prédéfinies 212
 - 2.5 Catalogue d'exemples et de composants 214
 - 2.6 Les assistants 214
 - 2.7 Utilisation de la soucoupe 214
 - 2.8 Saisie de code assistée 217
- 3. Partage de connaissances et documentation du projet 218
 - 3.1 Dictionnaire 218
 - 3.2 Documentation du projet 220
- 4. Aide à la mise au point 223
 - 4.1 Compilation 223
 - 4.2 Débogueur 223
 - 4.3 Détection de code mort 226
 - 4.4 Éléments orphelins 227
 - 4.5 Audit statique 227
 - 4.6 Audit dynamique 229
- 5. Optimisation des performances 230
- 6. Technologie JITC (Just In Time Compilation) 234
- 7. Optimiseur de requêtes 234
- 8. Amélioration de l'IHM 235
- 9. Charte de programmation 236
- 10. Briques de code 237
- 11. Règles de gestion 238
- 12. Rétro-modélisation 240
- 13. Analyse d'impact 241
- 14. Références croisées 242
- 15. Gestion de configuration d'exécutable 243

16. Génération de la version client	246
16.1 Génération de l'exécutable	247
16.2 Création de la procédure d'installation	247

Chapitre 5

Le WLanguage et la programmation

1. Introduction	249
2. Variable, portée ou domaine de visibilité	249
3. Les constantes	250
4. Les types de variable	252
4.1 Booléen	253
4.2 Entier	254
4.3 Monétaire	254
4.4 Réel	254
4.5 Chaîne	255
4.6 Buffer	255
4.7 Date, Heure, DateHeure, Durée	255
4.8 Numérique	256
4.9 Variant	256
4.10 Constante Null	257
4.11 Types de variable avancés	257
4.11.1 Tableaux	257
4.11.2 Objets dynamiques	258
4.11.3 Variables composées et structures	259
4.11.4 Objets automation	260
4.11.5 Source de données	260
4.11.6 Description de fichier, de relation, de rubrique	261
4.11.7 Police	261
5. Les opérateurs	262
5.1 Opérateurs logiques	262
5.2 Opérateurs arithmétiques	263
5.3 Opérateurs binaires	263
5.4 Opérateurs de comparaison	264
5.5 Opérateurs sur chaînes de caractères	265
5.6 Opérateurs sur adresse	265

5.7	Opérateurs d'indirection	266
5.8	Opérateurs d'échange	266
5.9	Opérateurs divers	267
6.	Les instructions du WLangage	267
6.1	Instruction SELON	268
6.2	Instruction SI	269
6.3	Instructions simples : SORTIR, RETOUR, RENVOYER, CONTINUER	269
6.4	Instruction BOUCLE	270
6.5	Instruction GOTO	270
6.6	Instruction POUR	270
6.7	Instruction POUR TOUT	271
6.8	Instruction TANTQUE	271
7.	Les mots réservés	271
7.1	Allouer	272
7.2	Externe	272
7.3	Libérer	272
7.4	MaFenetre, MaSource, MonEtat, MoiMeme, MonParent, MonChampPopup	273
7.5	Modulo	273
7.6	Null	273
7.7	STOP	274
8.	Procédures et fonctions	274
8.1	Procédures globales	274
8.2	Procédures locales	274
8.3	Déclaration des procédures et appel	275
8.4	Passage de paramètre par valeur ou par adresse (référence)	276
8.5	Documentation automatique et aide à la saisie	276
8.6	Collection de procédures	276
9.	Gestion des exceptions	277
10.	Accès FTP	279
11.	Gestion des e-mails	280
12.	Sockets	281
12.1	Les sockets côté serveur	281
12.2	Les sockets côté client	283

13. Threads	284
14. Programmation orientée objet (POO) et classe	285
14.1 Éléments de la POO	285
14.1.1 Classe, objet, membre, méthode	285
14.1.2 Membre global	285
14.2 Principes de la POO	286
14.2.1 L'héritage	286
14.2.2 L'encapsulation	286
14.2.3 Le polymorphisme	287
14.2.4 La redéfinition de méthode	287
14.3 Syntaxe de déclaration	287
14.4 Création de classes avec WinDev	289
14.5 Objet dynamique	290
14.6 Appel des méthodes	290
15. Propriétés	290
16. Exemples de traitements opérationnels en WLangage	291
16.1 Premier exemple	291
16.2 Deuxième exemple	294
16.3 Troisième exemple	301
16.4 Quatrième exemple	311
17. Langages tiers	311
17.1 Introduction	311
17.2 Étapes de réalisation	312

Chapitre 6

Accès aux données

1. Introduction	315
2. Les fichiers textes	316
2.1 Principales fonctions	316
2.1.1 Manipulation du contenu des fichiers	316
2.1.2 Manipulation de fichiers présents	318
2.1.3 Manipulation des répertoires sur les disques	319
2.2 Exemples	320

- 3. Accès aux données via les ordres WLangage 324
 - 3.1 Accès aux données par le WLangage 325
 - 3.2 HFSQL Classic et HFSQL Client/Serveur 325
 - 3.3 Accès natif 326
 - 3.4 Accès OLE DB 326
 - 3.5 Exemple de code WLangage 327
- 4. Accès aux données via les ordres SQL 328
 - 4.1 HFSQL Classic et HFSQL Client/Serveur 328
 - 4.2 Accès natif 328
 - 4.3 Accès OLE DB 329
 - 4.4 Exemple de code SQL 329
- 5. Accès aux données via les ordres SQL ODBC 331
 - 5.1 HFSQL Classic et HFSQL Client/Serveur 331
 - 5.2 Accès natif 332
 - 5.3 Accès ODBC 332
 - 5.4 Exemple de code SQL ODBC 332
- 6. Databinding 334

Chapitre 7
Les outils et fonctionnalités

- 1. Introduction 337
- 2. Centre de maintenance : WDOutil 338
 - 2.1 Utilitaires de développement 339
 - 2.1.1 WDMap : éditeur de données HFSQL 339
 - 2.1.2 WDZip : gestionnaire d'archive 341
 - 2.1.3 RequêteAuST : requête au support technique 342
 - 2.1.4 WDInst : éditeur d'installation 342
 - 2.1.5 WDAPI : assistant à l'appel des API Windows 345
 - 2.1.6 WDXView : navigateur d'ActiveX 345
 - 2.1.7 WDDiagnosticConnexion 348
 - 2.1.8 WDAalytics : télémétrie 348
 - 2.1.9 WDHFDiff : comparateur HFSQL 349
 - 2.2 Utilitaires distribuables 350
 - 2.2.1 WDOptimiseur : optimiseur HFSQL 350
 - 2.2.2 WDModFic : modification automatique HFSQL 351

2.2.3	WDSQL : interrogateur SQL	356
2.2.4	WDServeurRPC : serveur d'accès distant	359
2.2.5	WDConver : import de fichiers de données	359
2.2.6	WDJournal : sauvegardes et restaurations	362
2.2.7	Cycle de vie : sauvegarde interactive	363
2.2.8	Cycle de vie : examen des journaux	365
2.2.9	Cycle de vie : restauration	365
2.2.10	WDTrans : gestionnaire de transaction	365
2.2.11	Mise en place d'une transaction	366
2.2.12	WDReplc : gestionnaire de réplication	367
2.2.13	WDTelDiagno : diagnostic de la configuration téléphonique	373
3.	Centre de Contrôle HFSQL	373
4.	Centre de Suivi de Projets	375
5.	Centre de Contrôle Utilisateur	376
6.	Éditeur de tests de non-régression	377
7.	WDFAdmin : administrateur de la fabrique logicielle	381
8.	RobotMonitor : moniteur du robot de surveillance	384
9.	Le SILO	384
10.	Gestionnaire de sources	385
10.1	Déclaration d'un projet dans le GDS	386
10.2	Comparaison de versions	394
10.3	Propriétés d'un élément	395
10.4	Modification d'un élément	397
10.5	Gestion des accès concurrentiels par le GDS	399
10.6	Travail nomade	400
11.	Composants	401
11.1	Présentation des composants	401
11.2	Création d'un composant externe	402
11.3	Utilisation d'un composant externe	412
12.	Groupware utilisateur	415
13.	Logiciel Etats et Requêtes	422

Chapitre 8
Le RAD

- 1. Introduction 423
- 2. Création d'un projet 424
- 3. Modèle conceptuel des données (MCD) 425
 - 3.1 Création du MCD 426
 - 3.2 Création d'une entité 427
 - 3.3 Création des propriétés des entités 427
 - 3.4 Création d'une association 430
 - 3.5 Création des relations entre entités et associations 431
 - 3.6 Harmonisation de la présentation
du modèle conceptuel de données 432
- 4. Génération de l'analyse 433
 - 4.1 Conversion du MCD en MLD 433
 - 4.2 Ajout de fichier dans l'analyse 438
- 5. Construction automatique de l'application (RAD) 445
- 6. Mise en place de traitements particuliers 449
 - 6.1 Adaptation de la fenêtre FEN_Table_COURRIER 450
 - 6.1.1 Création d'un champ de saisie avec aide à la recherche
pour la filière et le modèle 450
 - 6.1.2 Création du bouton de déclenchement de la sélection 452
 - 6.1.3 Création du bouton pour fusion par Word 453
 - 6.2 Mise en place d'un bouton d'impression et de fusion avec un état . . 458
 - 6.2.1 Création d'un état 458
 - 6.2.2 Création d'un bouton d'appel de l'état 465
- 7. Finalisation du développement 466
- 8. RAD MVP et base de données existante 466
 - 8.1 Création d'un projet avec une base de données existante 467
 - 8.2 Construction RAD MVP 474
 - 8.3 Analyse du code généré 483

Chapitre 9

Étude de cas

1. Introduction	487
1.1 Objectif de l'étude de cas	487
1.2 Présentation de l'étude de cas.	488
2. Le projet WinDev.	488
2.1 La description du projet	488
2.2 Le gestionnaire de sources	493
2.3 Les chartes	494
2.4 L'internationalisation	496
2.5 « Le saviez-vous ? »	499
2.6 La base de données	502
3. Les fichiers de données.	506
3.1 La création de fichiers	507
3.2 La création de liaisons.	521
3.3 La synchronisation de l'analyse avec le projet.	524
3.4 L'utilisation d'un fichier prédéfini	528
3.5 La génération d'un script SQL	534
4. Les fenêtres et les champs	538
4.1 La création de fenêtres	538
4.2 La création de champs	544
4.3 La création d'une fenêtre principale et d'un menu	572
5. La programmation et les fonctions du WLangage	578
5.1 Programmation des fenêtres et des champs	578
5.2 Les variables.	583
5.3 Les requêtes via l'éditeur de requêtes (requête de sélection)	586
5.4 La programmation d'un champ table.	595
5.5 Les procédures locales	598
5.6 Passage de valeurs par procédure	602
5.7 Les requêtes via l'éditeur de requêtes (requête de modification)	603
5.8 Les requêtes via l'éditeur de requêtes (requête d'insertion)	609
5.9 Les requêtes via l'éditeur de requêtes (requête de suppression)	611
5.10 Les requêtes en mode SQL (sans passer par l'éditeur de requêtes)	613

- 6. La compilation et le débogage. 615
 - 6.1 Utilisation des points d'arrêt 615
 - 6.2 Erreurs de compilation 616
 - 6.3 Débogage et suivi de code 616
 - 6.4 Utilisation de l'aide 622
- 7. Les états 624
 - 7.1 Création d'un état simple 624
 - 7.2 Création d'un état sur table 635
- 8. Les procédures et fonctions. 646
 - 8.1 Les collections de procédures et procédures globales 646
 - 8.2 Les fonctions 650
- 9. Les classes. 652
 - 9.1 Création d'une classe 653
 - 9.2 Création de méthodes. 655
 - 9.3 Implémentation de la classe. 659
- 10. Utilisation des fonctions e-mails 660
- 11. Utilisation des fonctions FTP 664
- 12. Tests de l'application. 675
 - 12.1 Création et modification de données 675
 - 12.2 Tests automatiques 688
- 13. Contrôle des données 699
 - 13.1 Utilisation de l'outil WDSql 699
 - 13.2 Utilisation de l'outil WDMMap 705
- 14. Intégration du RAD. 706
- 15. Génération de l'exécutable et de la procédure d'installation 715
 - 15.1 Création d'un exécutable autonome 715
 - 15.2 Création de la procédure d'installation 729
- 16. Le dossier du projet 740
- 17. Projet VICTEAL. 747
 - 17.1 TEST_MENU 747
- Index 749

Chapitre 5

Le WLangage et la programmation

1. Introduction

WinDev propose un langage qui comporte classiquement des variables de plusieurs types, des opérateurs et des instructions qui offrent la possibilité d'écrire des procédures et des fonctions. Il dispose de plusieurs familles de fonctions standards pour faciliter la tâche du développeur. Parmi elles, les fonctions de dessin, les fonctions HTTP, les fonctions de gestion des plannings, des fenêtres, LDAP, SAP, Wi-Fi, etc.

Les instructions et les fonctions peuvent être écrites en français, en anglais ou même en chinois. Chaque nom de fonction a son équivalent.

Nous allons décrire synthétiquement ces éléments de manière à être capable de les utiliser pour les besoins courants.

Rappelons auparavant quelques principes sur l'utilisation de variables.

2. Variable, portée ou domaine de visibilité

Pour avoir un code le plus fiable possible, il convient de déclarer les variables qui vont être utilisées dans un traitement. Cela permet au compilateur de signaler l'inexistence de déclaration ou de vérifier que certaines opérations d'affectation ou d'emploi dans des fonctions, sont possibles.

Une variable a donc un type. Il est de bonne pratique d'initialiser la variable à sa déclaration. WinDev nous facilite la tâche en initialisant une variable à une valeur naturelle selon le type : zéro pour les numériques, une chaîne vide pour le type caractère ou chaîne de caractères.

Une fois déclarée, une variable a "un espace de vie" qui est plus ou moins étendu selon qu'en plus du type nous précisons que la variable est locale ou globale.

Une variable est par défaut locale ; c'est-à-dire qu'elle n'existe que dans le traitement dans lequel elle apparaît : section de traitement, procédure, fonction. Ainsi, si le traitement appelle une fonction, la variable ne sera pas connue du code de la fonction appelée. Le seul et le meilleur moyen est de passer cette variable en paramètre de la fonction.

Si la variable est déclarée comme globale, elle est accessible par les traitements déclenchés depuis le traitement où elle a été déclarée.

Pour déclarer des variables comme étant globales, le mot-clé `Global` doit être indiqué avant la liste des déclarations des variables concernées. De même, pour les déclarer locales, le mot-clé `Local` est à placer avant la liste correspondante.

```
Global
CodeClient, NumFacture sont des entiers
Local
cNom, cPrenom sont des chaînes
```

Si une variable est déclarée globale dans le code d'initialisation d'un projet, elle est connue de tous les traitements du projet.

Si une variable est déclarée globale dans une fenêtre, elle est connue de tous les traitements découlant de cette fenêtre.

■ Remarque

Les champs d'une fenêtre réagissent comme des variables globales pour la fenêtre.

Si une variable est appelée du même nom qu'une variable définie dans un traitement de plus haut niveau, la variable du traitement courant "masque" la variable de plus haut niveau. C'est le contenu de la variable du traitement en cours qui est utilisé.

3. Les constantes

WinDev fournit différents fichiers qui comportent des listes de constantes. Les constantes Windows sont déclarées dans le fichier `WINCONST.wl`. Les limites des différents types de variables de WinDev sont dans le fichier `LIMITES.wl`.

Voici un extrait d'un fichier `WINCONST.wl` :

```
// définition des constantes système windows pour SendMessage à
utiliser en WLangage
// utilisation : EXTERNE "WINCONST.wl"

CONSTANT
```

```
WM_NULL = 0x0000
WM_CREATE = 0x0001
WM_DESTROY = 0x0002
WM_MOVE = 0x0003
WM_SIZE = 0x0005
WM_ACTIVATE = 0x0006
WA_INACTIVE = 0
WA_ACTIVE = 1
WA_CLICKACTIVE = 2
WM_SETFOCUS = 0x0007
WM_KILLFOCUS = 0x0008
WM_ENABLE = 0x000A
WM_SETREDRAW = 0x000B
WM_SETFONT = 0x0030
WM_GETFONT = 0x0031
WM_GETDLGCODE = 0x0087
WM_NCMOUSEMOVE = 0x00A0
WM_NCLBUTTONDOWN = 0x00A1
WM_NCLBUTTONUP = 0x00A2
WM_NCLBUTTONDBLCLK = 0x00A3
WM_NCRBUTTONDOWN = 0x00A4
WM_NCRBUTTONUP = 0x00A5
WM_NCRBUTTONDBLCLK = 0x00A6
WM_NCMBUTTONDOWN = 0x00A7
WM_NCMBUTTONUP = 0x00A8
WM_NCMBUTTONDBLCLK = 0x00A9
WM_KEYFIRST = 0x0100
WM_KEYDOWN = 0x0100
WM_KEYUP = 0x0101
WM_CHAR = 0x0102
WM_DEADCHAR = 0x0103
WM_SYSKEYDOWN = 0x0104
WM_SYSKEYUP = 0x0105
WM_SYSCHAR = 0x0106
WM_SYSDEADCHAR = 0x0107
WM_KEYLAST = 0x0108
WM_IME_STARTCOMPOSITION = 0x010D
WM_IME_ENDCOMPOSITION = 0x010E
WM_AFXLAST = 0x037F
WM_PENWINFIRST = 0x0380
WM_PENWINLAST = 0x038F
WM_USER = 0x0400
```

FIN

Nous pouvons créer nos propres constantes par une déclaration de la forme :

```
Constante
  CtEnCours=1
  CtResilie=2
Fin
```

Comme pour les variables, les constantes ne sont connues que des traitements déclenchés en dessous du traitement où elles ont été déclarées.

Si une variable est déclarée avec le même nom qu'une constante, c'est la variable qui est utilisée dans tout son domaine de visibilité. Elle masque la constante de même nom dans son espace.

L'intérêt d'utiliser une constante dans le code, au lieu d'une valeur, se manifeste s'il faut modifier la valeur initiale.

Avec l'emploi d'une constante il suffit de modifier la valeur dans l'instruction de déclaration, sinon cette valeur est à rectifier dans toutes les lignes de code où elle a été utilisée.

4. Les types de variable

WinDev propose des types simples de variables et des types avancés ou structurés.

Les types simples sont : booléen, entier, monétaire, réel, chaîne, buffer, date, heure, dateHeure, durée, variant, numérique.

Les types avancés comprennent les tableaux, les tableaux dynamiques, les tableaux fixes, les variables composées, les buffers dimensionnés, les structures et les structures particulières qui servent à décrire les fichiers, les relations et les rubriques, de manière temporaire.

Pour déclarer une ou des variables d'un même type, la syntaxe est la suivante :

```
<Variable1>, <Variable 2> sont des <Type de variable>
```

On peut initialiser une variable lors de la déclaration, sous la forme :

```
■ <Variable1> est un(e) <Type de variable> =<Valeur>
```

On ne peut pas initialiser plusieurs variables dans la ligne de déclaration. Seule, la dernière variable citée est initialisée si nous mettons une valeur au bout d'une ligne de déclaration avec plusieurs variables.

Les types ont des valeurs maximales et minimales ! Si une variable dépasse ces limites, un message d'erreur est généré en mode test mais en exécution, la valeur est fautive et il n'y a pas de message.

Les valeurs limites sont récapitulées dans le fichier personnel\externe\limites.wl.

Exemple de déclaration de variables de types simples

```
fic est un entier
err est un booléen = Faux
tmp est une chaîne = "\\ServeurVIC\Echange\VP\ftpcentral.txt"
up est un booléen
res est une chaîne =""
```

Exemple de déclaration de variables de types avancés ou structurés

```
RequeteSelect est une Source de Données

TabFichiersXML2 est un tableau de chaînes

StructVirementPartielCartonReserve est une Structure
  CartNumero est une chaîne
  CartZone est un entier
  CartAllee est une chaîne
  CartVerticale est un entier
  CartPlateau est un entier
  CartQte est un entier
  CartBoit est une chaîne
FIN

GF_TB_ListeCartonReserve est un tableau dynamique

GF_TB_ListeCartonReserve = allouer un tableau de
0 StructVirementPartielCartonReserve

ConnexionSQL est une Connexion
SI HOuvreConnexion(ConnexionSQL, NomUser, Mdp, NomServeur,
NomBase, hOleDbSQLServer, hOLectureEcriture) = Faux ALORS
  FinProgramme("L'ouverture de la connexion a échoué.")
FIN
```

4.1 Booléen

Une variable de type booléen ne prend que deux valeurs : Vrai ou Faux. La valeur Faux vaut zéro. La valeur Vrai est différente de zéro.

Exemple

```
■ BValeur est un booléen
```

4.2 Entier

Les entiers de WinDev sont codés par défaut sur quatre octets, ce qui permet de manipuler des entiers assez grands avec le type de base.

Sinon, nous pouvons choisir les types suivants :

Entier	$-2 * 10^9$ à $2 * 10^9$
Entier sur 1 octet	-128 à 127 (incluses)
Entier sur 2 octets	-32 768 à 32 767 (incluses)
Entier sur 8 octets	$-9 * 10^{18}$ à $+9 * 10^{18}$
Entier non signé	0 à $4 * 10^9$
Entier sur 1 octet non signé	0 à 255
Entier sur 2 octets non signé	0 à 65 535
Entier sur 4 octets non signé	0 à $4 * 10^9$
Entier sur 8 octets non signé	0 à $18 * 10^{18}$

Le type **Entier Système** est disponible et s'adapte à la taille gérée par le système. Il est de 4 octets en 32 bits et de 8 octets en 64 bits.

4.3 Monétaire

Le type monétaire est un type important dans WinDev car il garantit la précision des calculs jusqu'à six décimales. Il est codé sur 10 octets et les valeurs extrêmes vont de + à - 600 mille milliards environ.

Exemple

■ **MMontant** est un monétaire

4.4 Réel

WinDev propose des réels sur 4 octets avec six chiffres significatifs ou 8 octets avec 15 chiffres significatifs. On pourra préférer utiliser ce type si la précision des décimales n'est pas importante et pour économiser un peu de place.