

DÉVELOPPEZ 10 FOIS PLUS VITE

Auto-Formation

XIII
Express

PC SOFT

Version d'évaluation

Diffusion interdite sans l'accord écrit de PC SOFT

Visitez régulièrement le site www.pcsoft.fr, espace actualités pour vérifier si des mises à jour sont proposées.

Adresse e-mail Support Technique Gratuit : supportgratuit@pcsoft.fr

Conseil : Pour recevoir directement des mises à jour intermédiaires et des conseils d'utilisation, abonnez-vous à la **LST** (revue trimestrielle + CD), en français.

Cette documentation n'est pas contractuelle. PC SOFT se réserve le droit de modifier ou de supprimer tout sujet traité dans ce document.

Tous les noms de produits ou autres marques cités dans cet ouvrage sont des marques déposées par leurs propriétaires respectifs.
© PC SOFT 2008 : Aucune reproduction intégrale ou partielle du présent ouvrage sur quelque support que ce soit ne peut être effectuée sans l'autorisation expresse de PC SOFT.

SOMMAIRE

Licence d'utilisation

Licence d'utilisation du logiciel en version "Express"	19
--	----

Introduction

Préliminaires	23
Présentation du cours d'auto-formation	23
Comment accéder à l'aide en ligne ?	24
Le guide	24
Légende des symboles	24
Si vous connaissez WinDev 11	25
Que fait-on avec WinDev ?	25
Les nouveautés de WinDev 12	25
Environnement de travail	25
Programmation	26
WinDev 12 version "Express"	27

PARTIE 1 - A LA DÉCOUVERTE DE WINDEV

Leçon 1.1. Concepts

Présentation	32
Concepts de base	32
Terminologie	33
Dans l'analyse	33
Dans les fenêtres et les états	34

Leçon 1.2. Mes premières fenêtres

Présentation	36
Ma première fenêtre : un compteur	36
Présentation	36
Réalisation	36
Gestion du compteur	38
Améliorez le look de la fenêtre et créez un bouton pour sortir de la fenêtre	41
Gérer la saisie d'un texte	42
Gérer la saisie d'un numérique pour faire un calcul	44
Utiliser une procédure	47
Création d'une fenêtre avec un menu	47

Leçon 1.3. Bases de programmation

Introduction	52
Déclaration des différents types de variables	52
Les différents types de variables	52
La déclaration des variables et leur portée	52

Instructions de base du WLangage	53
Instructions conditionnelles.....	53
Instructions de boucle.....	54
Les commentaires	55
Un exemple pratique sur les différentes instructions de base	55
Procédures et fonctions	56
Définition	56
Procédure locale.....	56
Procédure globale et collection de procédures.....	56
Comment choisir si une procédure est globale ou locale ?.....	56
A propos du passage des paramètres	57
Appel d'une procédure	57
Création d'une procédure	57
Un exemple pratique sur les procédures et fonctions	58
Traitements des chaînes	59
Exemple pratique.....	59
Détails.....	59
Traitements des numériques	61
Exemple pratique.....	61
Détails.....	61
Traitement des monétaires	62
Exemple pratique.....	62
Détails.....	62
Traitements des dates et des heures	64
Présentation.....	64
Exemple pratique.....	64
Masque de saisie et valeur retournée	64
Les dates	65
Exemple pratique.....	65
Quel jour sommes-nous?	65
Quel jour sommes-nous, mais avec le jour et le mois en lettres ?	66
Quel est le nombre de jours entre 2 dates ?	66
Quel est le jour de la semaine d'une date donnée ?	67
Petit exercice pratique.....	67
Les heures	67
Exemple pratique.....	67
Quelle heure est-il ?.....	67
Combien de temps s'est écoulé entre deux heures ?.....	68
Variable de type Date, Heure, DateHeure, Durée	68
Calculs avec des dates et des heures.....	69

Leçon 1.4. Questions / Réponses

Questions / Réponses	71
Comment visualiser l'élément auquel appartient le traitement en cours ?.....	71
Comment imprimer le code source ?.....	71
Comment rechercher et/ou remplacer une variable dans le code ?.....	71
Que signifie les signes "+" ou "-" dans l'éditeur de code ?	71
Est-il possible d'identifier la personne qui a écrit un code ?.....	72
Existe-t-il un moyen d'obtenir facilement la syntaxe ou de l'aide sur une fonction ?.....	72

Comment utiliser un glossaire sous l'éditeur de code ?	72
Quels sont les raccourcis clavier à connaître sous l'éditeur de code ?	73

PARTIE 2 - APPLICATION AVEC DONNÉES

Leçon 2.1. Présentation

Présentation de l'application réalisée dans cette partie	78
--	----

Leçon 2.2. Projet et analyse

Présentation	80
Création du projet	80
Création de l'analyse	81
Création de la description des fichiers de données	82
Création d'un fichier de données et de ses rubriques sous l'éditeur	83
Importation d'un fichier XML	88
Importation de fichiers du dictionnaire.....	91
Importation directe de fichiers de données existant	91
Création des liaisons	92
Configuration de l'analyse pour le RAD	94
Configuration du RAD dans la description de fichiers	94
Configuration du RAD dans la description des rubriques.....	96
Génération de l'analyse	96

Leçon 2.3. Le RAD complet

Qu'est-ce que le RAD ?	99
Génération du RAD	99
Test de l'application	102
Modification rapide : verrouillage de l'application	102

Leçon 2.4. Fenêtre de saisie

Présentation	105
Création d'une fenêtre de saisie	105
Gestion de l'ajout d'un enregistrement	108
Création du bouton d'ajout	108
Ajout d'un masque de saisie	110
Effacement des données après l'ajout	111
Visualiser les enregistrements saisis.....	112
Gestion de la recherche puis de la modification	113
Recherche à l'identique	113
Recherche générique	114
Modification de la fiche affichée	115
Parcours des fiches	116

Leçon 2.5. Table sur une requête

Présentation	119
Création d'une requête paramétrée	119
Création d'une fenêtre avec une table basée sur la requête	122

Création d'un état automatique sur la table	123
---	-----

Leçon 2.6. Etat avec requête intégrée

Présentation	125
Création d'un état	125
Test d'un état	128
Modification d'un état	129

Leçon 2.7. Le groupware utilisateur

Présentation	131
Intégrer le groupware utilisateur	131
Configurer le groupware utilisateur	133

Leçon 2.8. Le tableau de bord

Présentation	135
Tests automatiques	135
Optimisation des requêtes	139

Leçon 2.9. Déploiement de l'application

Présentation	142
Création de l'exécutable	142
Création de l'installation	145
Conclusion	148

Leçon 2.10. Questions / Réponses

Comment créer le menu principal de mon application ?	150
Comment lier une fenêtre à une option de mon menu principal ?	150
Comment insérer automatiquement le menu "?" ?	150
Comment créer un menu contextuel ?	150
Comment fermer une fenêtre automatiquement après une durée prédéfinie ?	151
Comment dialoguer avec l'utilisateur ?	151
Comment récupérer des paramètres passés en ligne de commande à un exécutable ?	151
Comment désinstaller une application réalisée avec WinDev ?	152
Comment créer une installation par CD-ROM ?	152
Comment créer un exécutable ?	152
Comment installer une application ?	152
Comment associer un icône à mon exécutable ?	153
Comment associer une image de lancement (splash screen) à mon exécutable ?	153
Comment mettre l'icône de mon application en bas à droite dans la barre de tâches ?	153
Comment installer un raccourci d'une application sur le bureau ?	153
Comment partager le framework WinDev entre les différentes applications installées sur un même poste ?	154
Comment détecter les éléments non utilisés par mon application ?	154

PARTIE 3 - FENÊTRES ET CHAMPS

Leçon 3.1. Les éditeurs

L'environnement	158
Les volets	158
Le menu personnalisé	159

Leçon 3.2. Les fenêtres

Comment créer une fenêtre?	162
Description d'une fenêtre : 8 onglets à votre disposition	163
Exemple pratique pour nos manipulations.....	163
Onglet "Général"	163
Onglet "IHM"	164
Onglet "Détail"	164
Onglet "Image"	165
Onglet "Langue"	165
Onglet "Note"	165
Onglet "Aide"	165
Onglet "Style".....	165

Leçon 3.3. Les champs

Introduction	167
Exemple pratique.....	167
Les champs classiques	167
Type de champ : Libellé.....	168
Type de champ : Champ de saisie.....	169
Type de champ : Bouton	174
Type de champ : Image	178
Type de champ : Sélecteur.....	183
Type de champ : Interrupteur	184
Type de champ : Liste.....	185
Type de champ : Liste Image	189
Type de champ : Combo.....	192
Type de champ : Table	194
Type de champ : Zone répétée	198
Type de champ : Arbre.....	200
Type de champ : Table Hiérarchique.....	201
Type de champ : Jauge.....	201
Type de champ : Champ Graphe	202
Les champs spécialisés	203
Type de champ : Carrousel	203
Type de champ : Cube.....	203
Type de champ : Tiroir	204
Type de champ : Forme	204
Type de champ : Champ Code-Barres.....	204
Type de champ : Ascenseur	205
Type de champ : Potentiomètre.....	205
Type de champ : Spin	206

Type de champ : Web caméra	206
Type de champ : Conférence	207
Type de champ : HTML	207
Type de champ : OLE, ActiveX.....	208
Type de champ : Boîte à outils.....	209
Type de champ : Barre d'outils	209
Type de champ : Xaml	210

Leçon 3.4. ergonomie de l'application

Améliorer l'ergonomie des champs et des fenêtres	212
Ergonomie de l'application	221
Exemple de programme gérant le multi fenêtrage de fenêtres libres (non MDI)	221
Exemple de programme gérant des fenêtres MDI	224

Leçon 3.5. Réutilisabilité

Qu'est-ce que la réutilisabilité ?	229
Réutiliser un ensemble de champs	229
Le superchamp	229
Fenêtre interne	230
Modèle de champs	231

Leçon 3.6. Questions / Réponses

Comment dupliquer un champ dans une fenêtre par programmation ?	233
Comment supprimer un champ dans une fenêtre par programmation ?	233
Comment gérer les plans d'une fenêtre ?	233
Comment gérer le clic sur un volet d'un onglet ?	234
Comment afficher la progression d'un traitement ?	234
Comment afficher un site internet dans une fenêtre ?	234
Comment modifier la couleur d'un champ ?	235
Comment rendre un bouton invisible ?	235
Comment créer le menu principal de mon application ?	236
Comment créer un menu contextuel ?	236
Comment modifier le curseur de survol pour un champ ou une fenêtre ?	236
Comment passer des paramètres à une fenêtre ?	237
Comment regrouper des champs pour modifier leurs propriétés par programmation ?	237
Comment transformer un champ interrupteur en un champ sélecteur ?	238
Comment mettre des boutons à la même taille ?	238
Comment gérer la persistance des champs ?	238
Comment ajouter une image en fond dans une fenêtre ?	238
Comment harmoniser le look de mes fenêtres ?	238
Comment gérer la transparence d'une fenêtre ?	239

PARTIE 4 - BASES DE DONNÉES ET ANALYSES

Leçon 4.1. Introduction

Présentation	244
Les différents modes d'accès aux bases de données	244
Accès Natif	244
Accès ODBC direct.....	245
Accès OLE DB.....	245
Accès ODBC via OLE DB.....	245
Quel accès utiliser ?	245

Leçon 4.2. Parcours de fichiers et de requêtes

Parcours de type HLitPremier	247
Parcours de type HLitRecherche	247
Parcours de type POUR TOUT	248
Parcours de type POUR TOUT AVEC	249
Quel type de parcours choisir ?	249

Leçon 4.3. Gestion des erreurs

Présentation	251
Gestion des erreurs en mode automatique	251
Gestion des erreurs en mode programmé manuellement	253
Gestion des erreurs en mode programmé avancé	255

Leçon 4.4. La Journalisation

Qu'est-ce que la journalisation ?	259
La journalisation pour quoi faire ?	259
Comment mettre en place la journalisation	259
Physiquement, comment se matérialise la journalisation ?	260
Comment réussir la mise en place de la journalisation ?	261

Leçon 4.5. Les transactions

Qu'est-ce qu'une transaction ?	263
Un exemple simple de la nécessité des transactions pour certains types de traitements	263
Les transactions sur Hyper File	264
Utilisation des transactions	264
Mettre en place la gestion des transactions.....	264
Manipuler les enregistrements lors d'une transaction : les règles à suivre	265
Que faire si une transaction a été interrompue ?	265

Leçon 4.6. Cryptage de données

A quoi sert le cryptage des données ?	267
Comment crypter les fichiers ?	268
Mettre en place le cryptage des fichiers.....	268
Gestion du cryptage en WLangage.....	270

Leçon 4.7. La réplication de données

A quoi sert la réplication de données ?	272
Mise en œuvre de la réplication de données	272
Mise en place	273
1. Activation	273
2. Déclaration de la base maître	273
3. Déclaration des bases abonnées.....	273
4. Réplication	273

Leçon 4.8. Les triggers

A quoi sert un trigger ?	275
Mise en place	275
Etape 1 : décrire le trigger (ici dans le code d'initialisation du projet)	275
Etape 2 : créer une procédure globale liée au trigger	276

Leçon 4.9. Bases externes

Présentation	278
Connexion à une base de données dans l'éditeur d'analyse	278
Connexion à une base de données par programmation	280
Création de la connexion : HOuvreConnexion	280
Association de la connexion aux fichiers de données : HChangeConnexion	281

PARTIE 5 - ADMINISTRER, METTRE EN PLACE UNE BASE HYPER FILE CLIENT / SERVEUR

Leçon 5.1. Introduction

Présentation	286
Pourquoi passer une application en mode Hyper File Client / Serveur ?	287

Leçon 5.2. Mise en place d'une application Client / Serveur

Présentation	289
Installation d'un serveur Hyper File local	289
Créer une application en mode Hyper File Client / Serveur	289
Migrer une application Hyper File Classic en mode Client / Serveur	290
Présentation.....	290
Migration de l'exemple.....	290
Fonctionnalités disponibles en mode Hyper File Client / Serveur	292

Leçon 5.3. Administrer

Présentation	294
Configuration des postes	294
Le centre de contrôle Hyper File	294
Créer un compte utilisateur dans le Centre de Contrôle Hyper File	296
Sauvegarde de la base	298
Conclusion	299

Leçon 5.4. Installation en clientèle

Présentation	301
Lancement de l'assistant de création d'installation	301

PARTIE 6 - LES REQUÊTES**Leçon 6.1. Créez une requête de sélection**

Introduction	306
Votre première requête	307

Leçon 6.2. Requêtes avec comptage

Réalisation d'une requête avec comptage	314
---	-----

Leçon 6.3. Requêtes Somme

Réalisation d'une requête "Somme"	317
---	-----

Leçon 6.4. Le langage SQL

Code source d'une requête	321
Les ordres SQL	322
L'instruction SELECT	322
L'instruction FROM	322
La clause WHERE	322
La clause GROUP BY	323
La clause ORDER BY	323

Leçon 6.5. Utilisation des Requêtes

Utilisation d'une requête depuis un état	325
Utilisation d'une requête depuis un champ	325
Utilisation d'une requête depuis le WLangage	325

PARTIE 7 - LES ÉTATS**Leçon 7.1. Votre premier état**

Vocabulaire de base	330
Présentation	332
Imprimer avec l'éditeur d'états ou imprimer en WLangage	332
Les états à créer	333
Création de votre premier état	333
Création de l'état : Liste des clients triée par code postal	333
Qu'est-ce qu'une rupture?	337

Leçon 7.2. Etats statistiques

Présentation	343
Etape par étape	343

Leçon 7.3. Etats Tableau Croisé

Présentation	354
Etape par étape	354

Leçon 7.4. Champs spécifiques

Présentation	360
Manipulations pour créer un champ Lien	360
Manipulations pour créer un champ clicable	361

Leçon 7.5. Exécution des etats

Exécuter les états en WLangage	364
Fonction iAperçu	364
Fonction ilprimeEtat	365

Leçon 7.6. Diffusez " Etats et Requêtes " avec vos applications

Lancement du logiciel Etats et Requêtes	367
Paramétrage du projet	368
Paramétrage de l'analyse	368
Paramétrage des états	369
Paramétrage des requêtes	369
Création de l'exécutable et diffusion de l'application.....	369
Installation de l'application.....	370
Test de l'application	372

PARTIE 8 - PROGRAMMATION AVANCÉE

Leçon 8.1. Import / Export

Présentation	378
Manipulation des fichiers texte, CSV,	378
A propos des caractères accentués	378
Exemple	379
Manipulation des fichiers XML	380
Exemple de lecture d'un fichier XML.....	381
Récupérer des valeurs	383
Effectuer une recherche.....	384
Manipulation des fichiers XLS	384

Leçon 8.2. Compilation dynamique

Présentation	386
Exemple	386

Leçon 8.3. Timer

Présentation	389
Créer un timer automatique	389
Créer un timer programmé	391

Leçon 8.4. Événement Windows

Introduction	393
Exemple : Détecter le clic sur une liste	394

Leçon 8.5. Gestion automatique des erreurs

Présentation	397
Fonctionnement.....	397
Mise en place.....	397
Types d'erreurs concernées.....	397
Utilisation de la gestion automatique des erreurs	398

Leçon 8.6. Indirection

Présentation	400
Exemple	400

Leçon 8.7. Les threads

Définition	402
Exemple	402

Leçon 8.8. Les sockets

Présentation	404
Application Serveur (simplifié)	404
Création de la socket.....	404
Echange de données.....	405
Fermeture de la socket	405
Application Client	405
Connexion au serveur.....	405
Echange de données.....	405
Fin de communication.....	406

Leçon 8.9. Le FTP

Présentation	408
Connexion à un serveur FTP	408
Envoyer un fichier	408
Récupérer un fichier	409
Déconnexion	409

Leçon 8.10. Les emails

Présentation	411
Les protocoles POP3 / SMTP	411
Ouvrir une session de messagerie	412
Envoyer un email	412
Lire un email	414
Déconnexion	415
Autres possibilités	415

Leçon 8.11. Le grapheur

Présentation	417
Le champ Graphe	417
Les fonctions graphes	420

Leçon 8.12. La POO

Concepts	422
Les classes	422
Les objets	422
Les membres	422
Les méthodes	422
Notion d'héritage	422
Constructeur et Destructeur	423
Encapsulation de données	423
Créer un programme orienté objet.....	423
Déclarer une classe.....	423
Décrire les méthodes	424
Déclarer et manipuler les objets	426

PARTIE 9 - GESTION DE PROJET AVANCÉE**Leçon 9.1. Tableau de bord**

Présentation	430
Les différents éléments du tableau de bord	430
Les listes.....	431
Les voyants	432
Les compteurs en relation avec les centres de contrôle.....	434
Configuration du tableau de bord	435

Leçon 9.2. GDS

Introduction	437
GDS	437
Principe du GDS.....	437
Création de la base de sources	438
Intégration d'un projet dans le GDS.....	439
Ouverture du projet depuis le GDS.....	442
Extraction d'un élément	442
Modification de l'élément extrait.....	443
Réintégration de l'élément extrait.....	444
Synchronisation du projet	444
Mode déconnecté (ou nomade).....	445
Maître et invité	445
Administrateur du GDS.....	446
Groupware développeur	446

Leçon 9.3. Centres de contrôles

Présentation	448
Le Centre de Suivi de projets	448
Gérer son temps	449
Suivre les retours client avec centres de contrôle	451
Les autres centres de contrôle	451

Leçon 9.4. Web Services

Introduction	453
Aspect théorique	453
Le standard SOAP	453
Etape par étape, l'importation d'un service Web	454
Etape 1 : Création du projet.....	454
Etape 2 : Importation du service Web.....	454
Etape 3 : utilisation de la collection de procédures.....	457
Créer des services Web XML	461

Leçon 9.5. Les composants

Présentation	463
Travail en équipe	463
Les projets volumineux	463
Les bases de données accédées par plusieurs projets.....	464
Les traitements utilisés dans plusieurs projets	464
La possibilité de diffuser une fonctionnalité ou un ensemble de fonctionnalités.....	464
Composant multiproduits.....	464
Etape par étape	465
Etape 1 : Création d'un composant.....	465
Etape 2 : Utilisation du composant	470
Distribuer un composant	472
Distribution simple	472
Distribution professionnelle.....	472

Leçon 9.6. Multi-configurations

Présentation	474
Création d'une configuration de projet	474
Gestion des configurations de projet	475
Génération multiple	476

Leçon 9.7. Le débogueur

Présentation	478
Déboguer une fenêtre	478
Lancer le débogueur.....	478
Visualiser les informations sous le débogueur	479
Fenêtre de trace du débogueur	480
Déboguer un projet	481
Options avancées du débogueur	481

Leçon 9.8. L'analyseur de performances

Présentation	483
Utiliser l'analyseur de performances	483
Optimiser un traitement avec l'analyseur de performances	484
Lire le résultat de l'analyseur de performances.....	484
Choisir un traitement à optimiser.....	484
Optimiser un traitement	484

Leçon 9.9. Import / Export

Importer des éléments	487
Exporter des éléments	489
Importations spécifiques	490
Importation d'un projet WebDev	490
Importation d'une fenêtre	490
Importation Access	491
Importation Visual Basic	491

Leçon 9.10. Installation d'une application

Présentation	493
Update réseau / update Web	493
Installation avec mise à jour réseau	493
Installation avec mise à jour Internet.....	495
Editeur d'installation	497
Questions / Réponses	498
Comment désinstaller une application réalisée avec WinDev ?	498
Comment créer une installation par CD-ROM ?	498
Comment installer une application ?	498

Leçon 9.11. UML et 3-tiers

Qu'est-ce que UML ?	500
Les diagrammes gérés par WinDev	500
Diagramme de classes.....	500
Diagramme de cas d'utilisation	500
Diagramme d'objets	500
Diagramme de composants.....	500
Diagramme d'activité	500
Diagramme de séquence.....	500
Diagramme de collaboration	501
Diagramme d'états-transitions	501
Diagramme de déploiement	501
3-Tiers	501
Le 3 tiers, c'est quoi ?	501
Comment mettre en oeuvre le 3 tiers ?	502
Manipulation des différentes couches.....	502

Leçon 9.12. Modélisation souple

Principe	504
Manipulations	504

Leçon 9.13. Multilingue

Qu'est-ce qu'une application multilingue ?	508
Choix des langues du projet	508
Internationalisation de l'analyse	510
Internationalisation des éléments du projet	511
Caractéristiques à traduire	511
Comment traduire toutes ces informations ?.....	514
Saisie directe des traductions	514
Cas particuliers	515
Traduction avec WDMSG et WDTRAD	517
Programmation du changement de langue	518
Utilisation d'un fichier de paramètres.....	518
Utilisation des options de menu.....	519
Gestion des alphabets spécifiques	519

Conclusion



LICENCE D'UTILISATION

Licence d'utilisation du logiciel en version "Express"

Attention: En installant et en utilisant un logiciel WINDEV version Express vous avez accepté les termes de la licence suivante :

Le LOGICIEL désigne le droit d'utilisation du logiciel WinDev, dans cette version de démonstration (appelée également Version Express, Version d'évaluation ou Version limitée).

L'EVALUATEUR représente la personne (physique ou morale) installant et / ou utilisant le logiciel.

1. Version Express

Ce LOGICIEL est proposé ici dans une version de "démonstration".

Cette version de démonstration est exclusivement destinée au test de ses fonctionnalités, et/ou à une utilisation par des étudiants d'un établissement reconnu par l'état pendant la durée de leurs études, en dehors de toute utilisation commerciale, partielle ou totale.

De nombreuses fonctionnalités sont bridées, limitées ou différentes de la version "commerciale".

Pour créer des applications destinées à être utilisées, il est nécessaire d'acquérir et d'utiliser une version "commerciale" et non cette version de démonstration et de test.

Il est strictement interdit de créer en totalité ou en partie une ou des applications dans un autre but que le TEST du LOGICIEL, ou la réalisation de Travaux Pratiques dans le cadre d'études.

L'utilisation des applications créées dans un but autre que le test de courte durée est interdite.

Toute utilisation autre que l'essai du logiciel, et/ou une utilisation pour une durée supérieure à celle autorisée, constitue une utilisation illégale qui donnera lieu à des poursuites.

Il est interdit de dupliquer et diffuser ce logiciel, même dans cette version de démonstration, sans l'accord écrit de PC SOFT.

2. Licence

Ce LOGICIEL doit être utilisé sur une et une seule machine par une et une seule personne à un instant donné. Le LOGICIEL est déclaré utilisé dès qu'il se trouve en "mémoire centrale" (également appelée RAM) d'un ordinateur.

Ce logiciel ne peut être utilisé que pour la durée autorisée.

3. Propriété du LOGICIEL

Ce logiciel est la propriété de PC SOFT. L'EVALUATEUR a simplement le droit d'utiliser ce LOGICIEL selon les conditions définies.

4. Documentation

La duplication de la documentation, en totalité ou en partie, est strictement interdite.

5. Utilisation

L'EVALUATEUR doit utiliser le LOGICIEL pour l'usage pour lequel il est diffusé. Il est interdit de modifier ou tenter de modifier ce LOGICIEL, de désassembler ou tenter de désassembler ce LOGICIEL. Il est interdit d'enlever ou de tenter d'enlever les mentions de copyright pouvant apparaître et/ou étant contenues dans le LOGICIEL.

La location ou le prêt de ce LOGICIEL est interdit.

Il est interdit d'utiliser WinDev "Express" pour créer et diffuser tout ou partie d'outil de développement, de LOGICIEL de création d'applications, de générateur de programmes, ou de tout LOGICIEL de même nature que WinDev lui-même. Interrogez-nous en cas de doute.

Les fonctions de génération de code JAVA ne peuvent être utilisées que dans le cadre de la manipulation habituelle du LOGICIEL, dans le but de créer une APPLICATION autonome.

Le code généré par WinDev "Express" ne doit pas être réutilisé dans d'autres logiciels ou dispositifs de création de logiciel. Le code généré avec cette fonctionnalité ne peut pas être inclus dans un logiciel, payant ou gratuit, qui ferait concurrence à WINDEV, WINDEV MOBILE ou WEBDEV, en totalité ou en partie.

6. Absence de responsabilités

Le LOGICIEL et la documentation qui l'accompagne sont fournis en l'état, SANS AUCUNE GARANTIE D'AUCUNE SORTE. PC SOFT ne saurait être tenu pour responsable pour tout dommage de quelque nature que ce soit, et en particulier en cas de perte ou détérioration des données, en cas de perte financière, en cas de perte d'exploitation, en cas de divergence des informations contenues dans la documentation avec le comportement du LOGICIEL, en cas de comportement du LOGICIEL différent de celui attendu. L'EVALUATEUR est seul et unique responsable du fonctionnement et du support des programmes qu'il aura réalisés avec le LOGICIEL.

7. Cas particuliers

7.1 Compilateur d'aide

Si le LOGICIEL est accompagné du compilateur d'aide "MS Help Compiler", il est précisé que la société Microsoft en détient les droits. PC SOFT ne détient que le droit de permettre à l'EVALUATEUR de créer et diffuser des aides réalisées à l'aide de ce compilateur d'aide.

7.2 Impressions sous Linux

Si le LOGICIEL permet l'impression à travers des serveurs LINUX, la licence de l'organisme ImageMagick Studio LLC s'applique.

Cette licence est présente sur le support et doit être lue et acceptée.

Acceptation de la licence :

Le fait de conserver et d'utiliser ce logiciel indique la compréhension et l'acceptation des termes de cette licence.

Ce contrat ne peut pas être modifié, sauf par un courrier original paraphé de la direction générale de PC SOFT.

Ce contrat de licence est régi par le droit français; tout litige qui pourrait en résulter sera de la compétence exclusive des tribunaux du siège social de PC SOFT.

PC SOFT,
3, rue de Puech Villa
BP 44 408
34197 Montpellier Cedex 5
France



INTRODUCTION

Préliminaires

Attention : Ce manuel est un cours d'auto-formation. Il est conseillé de consulter l'aide en ligne lorsque vous utilisez WinDev.

Le manuel du cours d'auto-formation a pour objectif de vous faire découvrir WinDev, de vous familiariser avec les éditeurs et de vous apprendre les concepts de WinDev. Ce manuel n'est PAS exhaustif des possibilités de WinDev.

Consacrez au moins quelques heures pour suivre ce cours et pour apprendre WinDev : vous les rentabiliserez vite !

Si vous essayez de démarrer le développement d'une application sans avoir suivi ce cours, vous perdrez du temps, beaucoup plus que deux jours.

Le cours a été conçu pour être suivi de deux façons :

- soit vous suivez tous les exercices détaillés dans les leçons du cours (méthode conseillée).
- soit, si vous êtes pressé et disposez d'une expérience significative, vous pouvez uniquement le lire sans exécuter les exercices, tous les exercices sont illustrés. Toutefois, pour que les concepts soient plus rapidement assimilés, il est préférable de suivre les manipulations.

WinDev évoluant en permanence, les copies des fenêtres illustrant le cours peuvent être différentes de celles du produit que vous possédez.

L'aspect langage n'est qu'un des nombreux aspects de développement. En prenant en compte tous les aspects du développement, la programmation devient beaucoup plus simple.

Présentation du cours d'auto-formation

Le cours d'auto-formation a été conçu pour vous permettre d'apprendre progressivement à utiliser WinDev. En suivant ce cours :


- d'une part, vous découvrirez les concepts importants expliqués d'une façon informelle; en effet dans ce cas, il s'agit de concepts à acquérir et à comprendre.
- d'autre part, vous serez amené à effectuer des manipulations pour illustrer les concepts qui viennent d'être expliqués.

Au fil de la lecture du cours, si vous voulez approfondir un concept ou si vous voulez avoir plus de détails sur une fonction de programmation, consultez l'aide en ligne (accessible directement depuis les éditeurs ou dans le guide).


La taille d'une leçon n'est pas obligatoirement proportionnelle à son intérêt ...

N'oubliez pas d'étudier ensuite les exemples livrés avec WinDev : ils sont très didactiques !

Comment accéder à l'aide en ligne ?

1. Sous l'éditeur de code, une aide spécifique est directement accessible pour chaque nom de fonction sélectionné, grâce à la touche [F1].
2. Le bouton  accessible depuis chaque fenêtre
3. Sous les éditeurs, directement par la touche [F1].
4. Sous les éditeurs, le menu d'aide (symbolisé par "?") vous permet d'obtenir le sommaire de l'aide ou de rechercher une information précise.

Le guide

Le guide regroupe plus de 150 assistants et de nombreux exemples. Accessible grâce à la "soutourne" (), il vous permettra de vous familiariser avec les fonctionnalités de WinDev et même de découvrir des fonctionnalités que vous ne connaissiez pas ou peu !

Légende des symboles



Ce symbole indique la durée de la leçon et de ses manipulations. Attention, le temps réel peut varier selon votre expérience



Un exemple est disponible pour compléter le cours. Les exemples sont disponibles dans le volet "Assistants, Exemples et Composants" de WinDev Mobile.



Ce symbole présente une "Astuce", la lecture du texte associé est vivement conseillée.



Ce symbole présente un "Avertissement", la lecture du texte associé est primordiale.



Ce symbole présente une "Note", la lecture du texte associé est conseillée.



Ce symbole présente le résultat d'un "Test", la lecture du texte associé est conseillée.

Si vous connaissez WinDev 11 ...

Si vous connaissez déjà WinDev 11, le suivi de ce nouveau cours ne pourra être que bénéfique : ce sera une bonne occasion de "réviser" les possibilités de WinDev !

Que fait-on avec WinDev ?

WinDev est un AGL (Atelier de Génie Logiciel). Il vous permet de développer des applications dans tous les domaines :

- Gestion des stocks
- Inventaire, traçabilité des marchandises
- Réglage et suivi de machines sur chaîne de production
- Prise de commandes pour traitement rapide sur un lieu de vente de passage (foire, école, stand, ...)
- Fiches clients
- Outil d'aide à la prise de décision d'urgence sur téléphone portable
- Vérification d'identité des visiteurs d'une manifestation : salon, présentation de produits, ...
- Médecin ou vétérinaire en déplacement
- Prises d'informations sur un lieu de passage : salon professionnel, rue pour un sondage, stade, ...
- Retour de matériel de location encombrant (outils, véhicules, ...) directement sur le parking
- ...

WinDev est un outil de développement complet qui intègre tous les outils nécessaires au cycle de réalisation d'une application.

Contrairement à d'autres langages de développement traditionnels, il n'est pas nécessaire de chercher et de rajouter des modules pour pouvoir concevoir, tester et installer une application.

Le L5G (Langage de 5ème Génération) de WinDev, le WLangage, vous étonnera par sa simplicité : quelques heures suffisent pour appréhender le langage, une semaine suffit en général pour maîtriser toute sa puissance !

Comme il est en français, le WLangage (disponible également en anglais) vous fera gagner du temps !

Les nouveautés de WinDev 12

Voici la liste, non exhaustive, des principales nouveautés de WinDev 12 par rapport à WinDev 11 :

Environnement de travail

Les principales nouveautés de l'environnement de travail :

- **Editeur de projet :**
 - Nouveau RAD (Rapid Application Development)
 - Composant interne
- **Editeur d'analyse :**

- Nom des rubriques et des fichiers multilangue
- Groupes de l'analyse manipulables par programmation
- **Editeur de fenêtres :**
 - Nouveau catalogue d'images.
 - Refactoring d'une sélection de champs en modèle de champs.
 - Modèles de modèles de champs.
 - Modèles de modèles de fenêtres.
 - Héritage de styles.
 - Nouvelles options de tri dans les listes et les tables.
 - Colonne mémoire et colonne interrupteur dans les tables fichier.
 - Table : colonne de type jauge, fusion de cellule.
 - Table Fichier chargée en mémoire.
- **Editeur de code :**
 - Go de procédure globale.
 - Volet Trace du débogueur pour afficher le résultat de la trace lors du dernier test.
 - Test automatique de fenêtre.
 - Itérations dans les tests automatiques.
 - Automate de tests pour lancer des tests sur le Pocket PC.
- **Hyper File Client / Serveur :**
 - Sauvegarde à chaud.
 - Tâche programmée sur le serveur
 - Triggers serveur.
- **Gestionnaire de Sources :**
 - Extraction automatique de la fenêtre dès sa modification.
 - Gestion des droits.
 - Gestion du changement physique de machine.
 - Masquage des fichiers de l'analyse.
 - Possibilité de traiter les tâches et les bugs lors de la ré-intégration.

Programmation

Les principales nouveautés de programmation :

- Hyper File : rubriques unicode.
- Pilotage des caméras et enregistrement vidéo
- Nouvelles fonctions :
 - Fonctions de manipulation de résolution vidéo
 - Fonctions TableauVersChaîne et ChaîneVersTableau
 - Fonctions SNMP
 - Fonctions de manipulation de services
 - Fonctions de gestion de la corbeille
 - ...

WinDev 12 version "Express"

Attention : Certaines fonctionnalités présentées ou citées dans ce cours d'auto-formation peuvent ne pas fonctionner dans cette version d'évaluation.



DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 1

A la découverte de WinDev

XIII ***Express***

LEÇON 1.1. CONCEPTS

Ce que vous allez apprendre dans cette leçon ...

- Les principaux concepts de WinDev
- La terminologie, le vocabulaire employé dans WinDev



Durée estimée : 10 mn

Présentation

WinDev est un AGL (Atelier de Génie Logiciel) complet permettant de développer des applications Windows dans de nombreux domaines : gestion, industrie, médical, ...

Les applications développées peuvent inclure l'accès à des informations stockées dans des bases de données.

En standard, WinDev inclut un puissant moteur de base de données Hyper File. Le moteur Hyper File est disponible en version réseau, et en version Client / Serveur.

De nombreuses autres bases de données peuvent être utilisées (comme nous le verrons dans la suite de ce cours) mais il est conseillé d'utiliser Hyper File pour vos applications afin d'optimiser vos traitements de fichiers de données.

Tout au long de ce cours d'auto-formation, vous allez apprendre à créer vos applications (avec ou sans base de données), et à les enrichir en utilisant les nombreuses fonctionnalités proposées par WinDev.

Mais pour bien débiter ce cours d'auto-formation, nous allons poser les bases : les concepts et le vocabulaire de base nécessaire pour aborder WinDev.

Concepts de base

WinDev permet de créer simplement une application. Mais que recouvre exactement ce terme **Application** ?

Une **application** est un outil qui permet de réaliser des tâches, des actions automatiquement. Une application est constituée d'un programme exécutable (ou d'un ensemble de programmes exécutables), de bibliothèques, de fichiers de données, ...

Un **programme exécutable** est un fichier composé d'éléments directement manipulables par l'utilisateur (fenêtres, états imprimés, ...). C'est ce que lance l'utilisateur final d'une application.

Pour créer un exécutable, WinDev propose de créer un **projet**. Un projet relie entre eux les différents éléments du programme, et les organise. C'est à partir du projet que pourra être créé le programme exécutable.

Si votre application manipule des données, WinDev permet de définir la structure de la base de données grâce à l'**analyse**. L'analyse WinDev contient la description des fichiers (appelés souvent "Tables" dans de nombreuses bases de données). Ces fichiers contiendront les données de l'application.

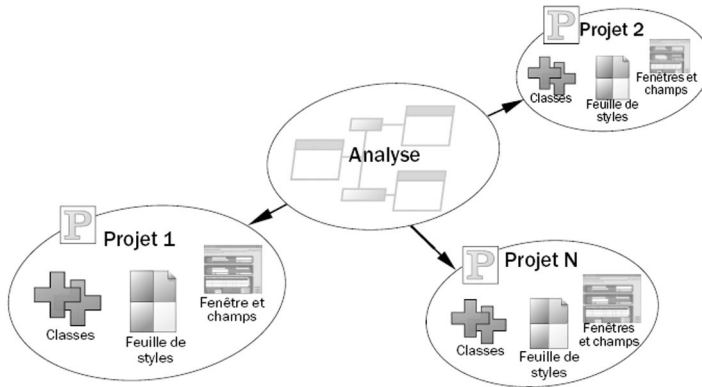


Notes

La description des fichiers de données dans l'analyse n'entraîne pas leur création. Les fichiers de données sont créés physiquement uniquement à l'exécution de l'application.

Un ou plusieurs projets WinDev peuvent être liés à la même analyse. Dans ce cas, on parle d'ana-

lyse partagée. Par exemple, une application de gestion commerciale peut être séparée en plusieurs modules exécutables. Chaque module utilise la même analyse (et en exécution, chaque exécutable peut également utiliser les mêmes fichiers de données).



Terminologie

Comme nous venons de le voir, un projet WinDev (lié si nécessaire à une analyse) permet de créer une application. Avant de commencer réellement à travailler avec WinDev, revenons sur les termes utilisés dans WinDev. En effet, si vous avez travaillé avec d'autres outils, de nombreux termes sont spécifiques à WinDev, et ne recouvrent pas toujours la même réalité que dans d'autres logiciels.

Dans l'analyse

Fichier : L'analyse permet de décrire la structure des fichiers de la base de données. Le terme "Fichier" correspond dans certaines bases de données à "une table".

Dans WinDev, le terme "**Table**" est réservé pour désigner un objet graphique permettant de visualiser le contenu d'un fichier sous forme de tableau, et / ou de saisir des lignes. Par exemple, une table peut permettre de saisir le détail d'une commande.

Enregistrement : Un enregistrement est quelquefois également appelé ligne. L'enregistrement d'un fichier correspond à l'ensemble des rubriques définies pour le fichier.

Rubrique : Dans l'analyse, le terme rubrique désigne une zone d'un fichier de données. L'ensemble des rubriques d'un fichier permet de définir la structure d'un enregistrement.

Clé/index : Avec WinDev et sa base de données Hyper File, la notion d'index est liée à la notion de clé. La notion de clé fait partie des caractéristiques d'une rubrique. Les clés permettent d'accélérer les accès aux données ou de faciliter les parcours de fichiers. En WinDev, si un fichier Hyper File a plusieurs rubriques clé, en exécution, un seul fichier d'index sera créé.

Dans les fenêtres et les états

Fenêtre : Les fenêtres permettent d'afficher ou de saisir à l'écran des informations. Les fenêtres sont également appelées "Écrans" ou "Boîtes de dialogue". L'utilisateur peut agir directement sur les fenêtres par l'intermédiaire de champs, de boutons, ...

Etat : Les états permettent d'obtenir une vue personnalisée d'informations. Ces informations peuvent provenir de la base de données, de fichiers texte, de champs présents dans les fenêtres, ... Les états peuvent être visualisés à l'écran, imprimés sur papier, générés en PDF ou en HTML, ...

Champ : Le mot "champ" est le terme utilisé pour désigner les différents objets graphiques affichés dans une fenêtre ou dans un état.

Gabarit : Le gabarit permet de définir le "look" de l'application : apparence visuelle des fenêtres, des boutons, des champs, ...

Style : Le style regroupe les caractéristiques graphiques d'un élément : image de fond, bordure, police, ... Les styles des différents éléments constituant l'interface d'une application WinDev sont regroupés dans une feuille de styles.



Important

Dans une application, l'entité "NomClient" peut correspondre :

- au nom d'un champ d'une fenêtre
- au nom d'un champ d'un état
- à la rubrique d'un fichier
- à une variable définie par le programmeur

Nous avons vu les termes de base à connaître pour réaliser une application WinDev. Nous allons commencer la programmation en concevant nos premières fenêtres.

LEÇON 1.2. MES PREMIÈRES FENÊTRES

Ce que vous allez apprendre dans cette leçon ...

- Comment créer une fenêtre compteur
- Comment saisir un texte
- Comment faire un calcul
- Créer une fenêtre avec menu



Durée estimée : 30 mn

Présentation

Pour commencer à utiliser WinDev, nous allons simplement créer quelques fenêtres. Ces quelques exemples vous permettront de connaître les bases de la programmation en WinDev et vous permettront ensuite d'aborder des sujets plus complets.

Les leçons de cette première partie vont vous permettre de :

- créer des fenêtres simples,
- manipuler des chaînes, des numériques, des monétaires,
- manipuler des dates et des heures.

Dans cette première leçon, nous allons créer les fenêtres suivantes :

- Une fenêtre compteur,
- Une fenêtre de saisie de texte,
- Une fenêtre de calcul,
- Un menu pour rassembler les fenêtres créées précédemment.

Ces fenêtres seront regroupées pour des besoins pratiques dans un projet.

Ma première fenêtre : un compteur

Présentation

Nous allons réaliser la fenêtre suivante :



Le compteur numérique sera incrémenté ou décrémenté avec les boutons fléchés.

Vous pouvez être tenté de dire que c'est trop simple, trop basique, ... mais nous vous conseillons de réaliser cette fenêtre. Vous risquez d'être surpris par la facilité et l'intuitivité de l'éditeur de WinDev. De plus, cette fenêtre vous permettra de découvrir des concepts fondamentaux pour la suite de ce cours.

Réalisation

- ▶ Lancez WinDev 12.
- ▶ Ouvrez le projet "Mes_Premières_Fenêtres". Pour cela, sélectionnez l'option de menu "Fichier .. Ouvrir un projet", et sélectionnez le projet "Mes_Premières_Fenêtres.WDP" situé dans le sous-répertoire "\\Autoformation\\Exercices\\Mes_Premières_Fenêtres" du répertoire d'installation de WinDev.

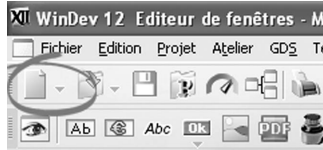


Important

Dans cette partie, nous nous concentrons sur la création de fenêtres simples. Le projet "Mes_Premières_Fenêtres" est un projet vide, déjà créé. La création de projet sera abordée dans une prochaine leçon.

► Pour créer la fenêtre :

1. Cliquez sur le bouton  dans la barre d'outils de WinDev :



2. Sélectionnez "Fenêtre". L'assistant de création de fenêtre s'affiche.

3. Sélectionnez "Vierge". Dans la liste des gabarits, sélectionnez par exemple le gabarit "Media Center". Vous pouvez choisir un autre gabarit proposé dans la liste.

4. Validez.

Nous allons renseigner les informations de la fenêtre (nom, titre, description).

► Faites un clic droit sur la fenêtre et sélectionnez l'option "Description". La fenêtre de description s'affiche. Cette fenêtre contient par défaut le nom de la fenêtre "FEN_Fenêtre1".




Note

Observez bien le nom de la fenêtre que WinDev propose par défaut : ce nom commence par les lettres "FEN_". Ce préfixe est automatiquement mis en place car le projet utilise une charte de programmation. La charte de programmation permet de définir un préfixe pour chaque type d'objet. Nous verrons cette charte de programmation en détail plus loin dans ce cours.

► Saisissez les informations suivantes :



1. Le nom "logique" de la fenêtre : remplacez "FEN_Fenêtre1" par "FEN_Compteur". Ce nom sera le nom sous lequel sera enregistrée la fenêtre sur le disque (avec l'extension "WDW") dans le répertoire des sources du projet. Ce nom sera également utilisé en programmation pour manipuler la fenêtre, nous y reviendrons.
2. La description de la fenêtre : "Exemple d'un compteur". Cette description n'est pas destinée à l'utilisateur final mais au développeur. Cette description est affichée par exemple dans le dossier, lors de la visualisation des éléments du projet, ...
3. Le titre de la fenêtre : "Exemple d'un compteur". Ce titre est affiché dans la barre de titre de la fenêtre. Ce titre permet de renseigner l'utilisateur final sur les fonctionnalités de la fenêtre.


- ▶ Validez. Le titre apparaît dans la barre de titre.
- ▶ Enregistrez la fenêtre en cliquant sur l'icone  .

Gestion du compteur

Pour gérer le compteur numérique, vous allez créer :

- un champ dans lequel va s'afficher la valeur numérique,
- deux boutons qui vont permettre d'ajouter 1 ou de soustraire 1 au compteur.

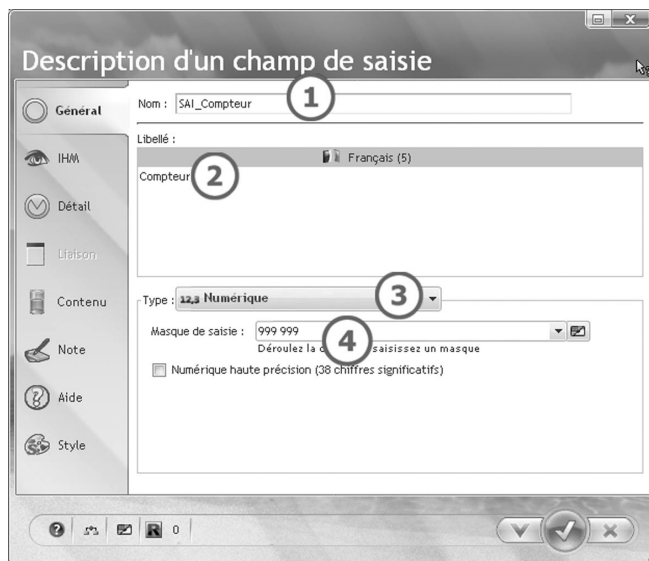
- ▶ Pour créer le champ de saisie :

1. Cliquez sur l'icone  de la barre d'outils.
2. Déplacez la souris vers votre fenêtre.
3. Cliquez ensuite à l'endroit où le champ doit être créé.

- ▶ Double-cliquez sur le champ.

La fenêtre de description du champ apparaît.

Cette fenêtre contient les informations générées par défaut : le nom du champ "SAI_Saisie1" ("SAI_" correspondant au préfixe utilisé pour identifier les champs de saisie dans la charte de programmation), son libellé, et son type. Nous allons modifier ces informations :



1. Saisissez le nom "SAI_Compteur".
2. Entrez le libellé "Compteur".
3. Modifiez le type du champ pour qu'il devienne numérique.
4. Saisissez le format (option "Masque de saisie") : "999 999", cela signifie que :
 - le nombre peut comporter jusqu'à 6 chiffres.
 - le nombre ne comporte aucune partie décimale.
 - le séparateur de milliers est un espace.



Notes

Par défaut, le masque de saisie du champ correspond au masque numérique défini par le projet (option "Projet .. Description du projet", onglet "Langues"). Ainsi, le même masque est utilisé automatiquement dans tous les champs numériques de l'application.
 Cette fonctionnalité est très utile dans les applications multilingues.

Nous voulons donner une valeur initiale au compteur. Pour cela, il suffit d'initialiser le contenu du champ avec cette valeur (onglet "Contenu").



Notes

Beaucoup de développeurs feraient cette opération par programme : il faudrait écrire une ligne de code pour affecter une valeur dans ce champ.
 Avec WinDev, pas besoin de saisir du code. Ce type d'information fait partie des caractéristiques du champ.


- ▶ Cliquez sur l'onglet "Contenu" et saisissez la valeur initiale, par exemple, "100". Bien entendu, comme partout dans WinDev, vous pouvez procéder de manière traditionnelle en saisissant la ligne suivante en WLangage dans le code d'initialisation du champ :

```
SAI_Compteur = 100
```



Notes

Dans cette ligne de code, le signe "=" permet d'affecter le champ "SAI_Compteur" avec la valeur indiquée. La syntaxe utilisée est la suivante :
 <Destination> = <Valeur à affecter>

- ▶ Validez les informations du champ.
 Sous l'éditeur, le champ contient maintenant la valeur 100. Ce concept se nomme "Live Data" : vous voyez les données en temps réel! Vous retrouverez ce concept un peu plus tard, lors de la manipulation des fichiers de données.
- ▶ Nous allons maintenant créer le champ "Spin" qui va nous permettre de modifier la valeur du compteur. Pour créer le champ spin :
 1. Cliquez sur l'icone .
 2. Déplacez la souris vers la fenêtre pour positionner le champ.

3. Cliquez à l'endroit où le champ doit être créé (à côté du champ de saisie).



Notes

Enregistrez votre fenêtre. Ainsi, le nom des champs sera automatiquement proposé dans le code.

- ▶ Affichez le code du champ Spin (option "Code" du menu contextuel du champ). Ce champ contient en plus du code d'initialisation, deux codes spécifiques :
 - Le code d'incrémentement : ce code est exécuté lorsque l'utilisateur manipule le champ spin pour augmenter une valeur.
 - Le code de décrémentement : ce code est exécuté lorsque l'utilisateur manipule le champ spin pour diminuer une valeur.
- ▶ Dans le code d'incrémentement, saisissez le code suivant :

```
SAI_Compteur = SAI_Compteur + 1
```

Cette ligne de code signifie que l'on ajoute 1 à la valeur du champ SAI_COMPTEUR. Le champ SAI_COMPTEUR est alors manipulé comme une variable.



Notes

Les adeptes du langage C peuvent saisir tout simplement :

```
SAI_Compteur++
```

Pour les non initiés au langage C, "++" indique une incrémentement de 1 unité. Vous pouvez également utiliser la syntaxe :

```
SAI_Compteur += 1
```

Remarque : Dès la saisie des trois premiers caractères, WinDev propose tous les mots du vocabulaire WLanguage contenant ces caractères. L'assistance au développement est poussée très loin. Vous ne pouvez plus vous tromper en saisissant le nom d'un élément : les erreurs de syntaxes sont minimisées. Il suffit de sélectionner le mot désiré et de valider par la touche [Entrée]. Vous pourrez ainsi vous concentrer sur l'algorithme.



Notes


C'est pour cela que la charte de programmation est très importante. Tous les éléments manipulés dans le code de l'application utilisent la même norme et peuvent ainsi être facilement retrouvés lors de la saisie du code.

Le principe est le même pour le code de décrémentement. Saisissez son code :

```
SAI_Compteur = SAI_Compteur - 1
```

Ce code signifie que l'on enlève 1 à la valeur du champ SAI_Compteur.

- ▶ Testons cette fenêtre :

1. Enregistrez les modifications en cliquant sur l'icône  (ou sur l'option "Fichier .. Enregistrer", ou par la combinaison de touches [Ctrl]+[S]).
2. Cliquez sur l'icône "GO" (ou [F9]).
3. Testez les différents champs, et observez les changements du champ de saisie.

Tout développeur sait que tester un programme est souvent long, voire fastidieux. Avec WinDev, en UN CLIC, vous testez la fenêtre, l'état, ou la procédure que vous êtes en train de réaliser. C'est à la fois simple et rapide!

- ▶ Cliquez sur le bouton "x" situé dans la barre de titre pour fermer la fenêtre. L'éditeur de WinDev réapparaît.

Améliorez le look de la fenêtre et créez un bouton pour sortir de la fenêtre

Lors de ce premier test, vous avez remarqué que votre fenêtre est trop grande, et qu'elle peut être redimensionnée alors qu'elle ne contient que quelques champs.

Nous allons revoir rapidement l'interface de cette fenêtre.

- ▶ Pour réduire la taille de la fenêtre sous l'éditeur :
 1. Cliquez sur la fenêtre : des poignées noires apparaissent autour de la fenêtre.
 2. Cliquez sur la poignée située en bas à droite, et réduisez la taille de la fenêtre en maintenant le bouton de la souris enfoncé.
- ▶ Pour que la fenêtre ne soit pas redimensionnable :
 1. Double-cliquez sur la fenêtre. La fenêtre de description s'affiche.
 2. Dans l'onglet "Style", décochez l'option "Icône d'agrandissement".
 3. Validez.



Notes

Nous approfondirons les conseils d'interface et d'ergonomie dans un prochain chapitre de ce cours.

Dans une fenêtre, il y a toujours un bouton ou une option pour quitter la fenêtre. Bien entendu, si vous n'ajoutez pas de bouton ou d'option, il reste toujours la possibilité de fermer la fenêtre en cliquant sur l'icône "X" de la barre de titre, mais ce n'est pas très convivial.

- ▶ Pour créer le bouton de fermeture :
 1. Cliquez sur l'icône **OK** dans la barre d'outils.
 2. Déplacez la souris dans votre fenêtre pour positionner le champ.
 3. Cliquez à la position où le bouton doit être créé.
- ▶ Saisissez le libellé du bouton :
 1. Cliquez deux fois sur le bouton. Le texte affiché dans le bouton passe en édition.
 2. Saisissez directement le libellé "&Fermer".
Le "&" permet de définir la lettre d'appel du bouton : la lettre qui suit est automatiquement soulignée et considérée comme lettre d'appel. L'utilisation de la touche [ALT] et de la lettre soulignée entraîne la même action qu'un clic de la souris sur ce bouton.
 3. Validez en appuyant sur la touche "Entrée".
- ▶ Pour définir le type du bouton et l'action associée au bouton :
 1. Affichez la description du champ (option "Description" du menu contextuel ou double-clic sur le bouton).
 2. Dans la combo "Action", sélectionnez l'action "Fermer la fenêtre (annuler)".

3. Dans l'onglet "IHM", sélectionnez le type "Abandon".



Notes

WinDev propose plusieurs types pour les boutons. Nous les verrons en détail dans le chapitre consacré à l'ergonomie.

Pour fermer une fenêtre, le bouton utilisé doit être de type "Abandon".

4. Fermez la fenêtre de description du bouton.

Astuce : Pour améliorer le look de votre bouton, associez un style prédéfini :

1. Sélectionnez le bouton "Fermer" que vous avez créé.
2. Sélectionnez l'option "Choisir un style" du menu contextuel.
3. Choisissez le style voulu (par exemple le libellé avec une croix rouge positionnée à côté), appelé "Btn_std_close". Validez.
4. Le look du bouton est automatiquement mis à jour.



Vous pouvez facilement déplacer le bouton de fermeture : cliquez sur le bouton de fermeture avec le bouton gauche de la souris. Tout en maintenant le bouton de la souris enfoncé, faites glisser la souris (et le champ) à l'emplacement voulu.

- ▶ Relancez le test de cette fenêtre, pour visualiser le résultat en exécution.


Nous en avons terminé avec notre compteur!

Gérer la saisie d'un texte


Nous allons maintenant voir comment manipuler un champ de saisie texte.

Dans une nouvelle fenêtre, vous allez maintenant créer un champ de saisie. Ce champ sera utilisé pour saisir un prénom. Un bouton permettra d'afficher ce prénom.

- ▶ Pour créer une nouvelle fenêtre :

1. Cliquez sur l'icone .
2. Sélectionnez "Fenêtre".
3. Sélectionnez "Vierge". Le gabarit "Media Center" est proposé automatiquement.
4. Validez
5. Faites un clic droit sur la fenêtre puis sélectionnez l'option "Description".
6. Donnez à la fenêtre le nom "FEN_Saisie", le titre et la description "Exemple de saisie". Validez la fenêtre de description.
7. Enregistrez la fenêtre.

- ▶ Pour créer le champ de saisie :

1. Cliquez sur l'icone  puis positionnez le champ dans la fenêtre.
2. Faites un clic droit sur le champ et sélectionnez l'option "Description".

3. Modifiez les caractéristiques du champ.

- Ce champ a pour nom : "SAI_Prénom".
- Ce champ a pour libellé : "Prénom"
- Ce champ est de type texte.



Notes

Par défaut, les caractères seront saisis en majuscules ou en minuscules selon la position de la touche [CAPSLOCK] du clavier. Ce genre de saisie peut être très gênant lors d'une recherche par exemple.

WinDev propose la gestion d'un masque de saisie pour un champ. Le masque de saisie permet de mettre en forme la valeur saisie, automatiquement et sans aucune ligne de code.

- ▶ Modifiez le format du champ de saisie (masque de saisie) et sélectionnez "1^{ère} lettre en majuscule" (la première lettre saisie sera en majuscule et les autres seront en minuscules). Ce masque de saisie va nous être utile pour afficher le prénom saisi avec une majuscule comme première lettre. Validez.
- ▶ Pour créer le bouton "Afficher" :
 1. Cliquez sur l'icone **OK** et positionnez le champ dans la fenêtre.
 2. Modifiez le libellé par "Afficher". Le nom du bouton devient automatiquement "BTN_Afficher".

Pour afficher le résultat de la saisie, vous auriez peut-être envie de créer une nouvelle fenêtre et d'afficher le résultat dans un champ "Libellé". Or, il existe en WLangage des fonctions permettant d'afficher un résultat dans des fenêtres système. Ces fonctions se nomment **Info** ou **Erreur**.

Pour notre exemple, nous utiliserons la fonction **Info**.

- ▶ Saisissez le code suivant dans le code de clic du bouton "Afficher" (option "Code" du menu contextuel du champ) :

```
Info("Vous avez saisi " + SAI_Prénom)
```



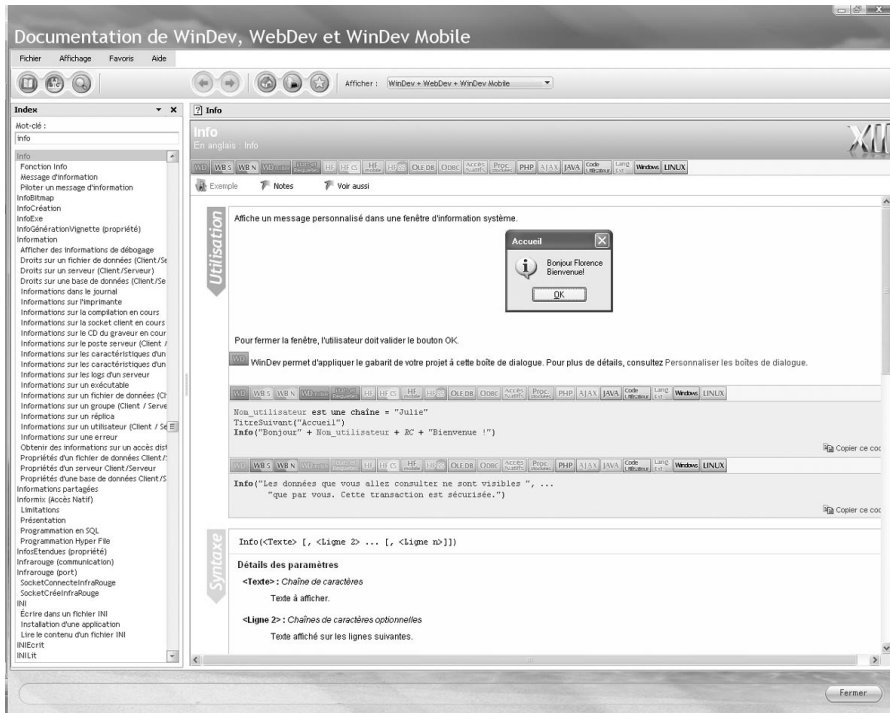
Notes

En saisissant ce code sous l'éditeur de code, vous constatez que les différents éléments saisis utilisent des couleurs différentes. En effet, l'éditeur de code permet ainsi d'identifier rapidement les différents éléments manipulés par le code :

- les fonctions du WLangage apparaissent en bleu,
- les chaînes de caractères (entre guillemets) apparaissent en violet,
- les noms de champs apparaissent en cyan.

La fonction **Info** affiche dans une boîte de dialogue le message passé en paramètre. Notre message est construit à partir du texte "Vous avez saisi " et de la valeur du champ "SAI_Prénom". Le signe "+" indique une opération dite de "concaténation" entre deux chaînes.



- ▶ Pour avoir plus d'informations sur cette fonction, affichons l'aide en ligne : il suffit de se positionner sur le nom de la fonction, et d'utiliser la touche [F1].



Notes

L'aide en ligne de WinDev permet de consulter les informations détaillées concernant les 2000 fonctions du WLangage. L'aide en ligne contient également l'aide des éditeurs, des champs, des astuces, ...

L'aide en ligne est commune à WinDev, WebDev et WinDev Mobile. Vous utilisez uniquement WinDev ? Pour afficher l'aide concernant uniquement ce produit, il suffit de sélectionner "WinDev" dans la liste "Afficher".

- ▶ Enregistrez la fenêtre et testez-la (en cliquant sur l'icône  puis sur .
- ▶ Pendant le test :
 1. Cliquez sur le bouton "Afficher".
 2. Validez.
 3. Pour sortir du test et revenir sous l'éditeur de fenêtres, cliquez sur l'icône "X" de la barre de titre.

Gérer la saisie d'un numérique pour faire un calcul


Dans la même fenêtre, nous allons maintenant :

- Créer deux champs de saisie numériques.
- Calculer et afficher la valeur TTC du montant HT saisi.


Le résultat du calcul sera affiché dans le champ "Prix TTC".

Nous allons séparer les deux parties de la fenêtre grâce à un séparateur.


► Pour créer un séparateur :

1. Cliquez sur l'icône .
2. Cliquez dans la fenêtre à la position où le séparateur doit être créé.
3. Dans l'assistant qui se lance, indiquez le sens du séparateur (horizontal) et son nom.
4. Validez l'assistant.

► Pour créer le champ de saisie du prix HT :

1. Cliquez sur l'icône  et créez le champ dans la fenêtre.
2. Faites un clic droit sur le champ et sélectionnez l'option "Description".
3. Dans la fenêtre de description :
 - Saisissez le nom du champ "SAI_PrixHT".
 - Saisissez le libellé "Prix HT".
 - Sélectionnez le type numérique.
4. Validez.

► Pour créer le champ dans lequel sera affiché le résultat :

1. Cliquez sur l'icône  et créez le champ dans la fenêtre.
2. Saisissez les informations du champ : faites un clic droit sur le champ et sélectionnez l'option "Description".
 - Indiquez le nom du champ "SAI_TTC".
 - Modifiez le libellé par "Prix TTC".
 - Sélectionnez le type numérique.
3. Le résultat affiché dans ce champ ne doit pas être modifiable. Cliquez sur l'onglet "IHM" et choisissez l'état initial "Affichage seul".
4. Validez.
5. Sauvez la fenêtre.


Il existe également des champs de saisie "Monétaires" et "Monétaires + Euro" (pour plus de détails, consultez l'aide en ligne).

Le calcul du montant TTC va être réalisé à chaque modification du champ contenant le montant hors taxe.

► Pour réaliser le calcul du montant TTC :


1. Affichez le code du champ "SAI_PrixHT" (option "Code" du menu contextuel du champ).
2. Dans le code "A chaque modification", saisissez le code suivant :

```
// On fige la TVA à 19.6% pour l'exemple
// cela aurait pu être une variable quelconque provenant
// d'une base de données
SAI_TTC = SAI_PrixHT * 1.196
```

3. Fermez la fenêtre de code.
4. Enregistrez la fenêtre.
5. Testez la fenêtre (bouton ). Saisissez une valeur dans le champ "Prix HT". Le résultat apparaît immédiatement dans le champ "Prix TTC".
6. Fermez la fenêtre de test.

Ce calcul prend en compte un taux fixe de TVA. Pour compliquer un peu notre exemple, nous allons donner la possibilité de sélectionner le taux de TVA dans une liste déroulante (appelée "combo").

► Pour créer le champ combo de sélection de la TVA :

1. Cliquez sur l'icône  puis cliquez dans la fenêtre à la position où le champ doit être créé (à côté du champ "Prix HT").
2. L'assistant de création de combo s'affiche. Cet assistant permet de définir simplement les principales caractéristiques du champ.
3. Sélectionnez l'option "Je veux saisir directement une liste de valeurs ou remplir la combo par programmation". Affichez l'écran suivant.
4. Laissez les options proposées par défaut. Affichez l'écran suivant.
5. Saisissez la liste des valeurs de TVA possibles :
 - 5.5
 - Appuyez sur la touche ENTREE
 - 18.6
 - Appuyez sur la touche ENTREE
 - 26.
6. Affichez l'écran suivant, et donnez un nom au champ (COMBO_TVA) et un libellé (TVA).
7. Validez.

Nous allons maintenant modifier le code du champ "Prix HT" pour prendre en compte le montant de la TVA sélectionné dans la combo.

► Pour prendre en compte le montant de la TVA sélectionné :

1. Affichez le code du champ "Prix HT" (option "Code" du menu contextuel).
2. Modifiez le code de la façon suivante :

```
SAI_TTC = SAI_PrixHT*(1+Val (COMBO_TVA..ValeurAffichée)/100)
```

Ce code calcule le montant TTC en utilisant la valeur sélectionnée dans la combo (donnée obtenue avec la propriété **..ValeurAffichée**). Le contenu de la combo étant une chaîne de caractères, la fonction **Val** permet d'obtenir une valeur numérique afin de calculer le montant TTC.



Notes

Le WLanguage est composé de fonctions et de propriétés. Les fonctions peuvent attendre des paramètres et renvoient des résultats. Les propriétés sont directement appliquées aux champs grâce à la syntaxe :

<Nom du champ>..**<Nom de la propriété>**.

3. Testez votre fenêtre. Saisissez une valeur dans le champ "Prix HT". Le résultat apparaît immédiatement dans le champ "Prix TTC" en fonction de la TVA sélectionnée. Cependant si vous changez le taux de TVA, la valeur du champ "Prix TTC" n'est pas modifiée. Nous allons maintenant corriger ce problème.

4. Fermez la fenêtre. L'éditeur de code est de nouveau affiché.

► Pour prendre en compte le montant de la TVA, à chaque modification de la valeur de la TVA dans la combo, il suffit de recopier le code "A chaque modification" de "Prix HT" dans le code "Sélection d'une ligne" de la combo "COMBO_TVA" :

1. Affichez le code du champ "Prix HT" (option "Code" du menu contextuel par exemple).
2. Sélectionnez le code présent dans le traitement "A chaque modification" de "Prix HT".

3. Utilisez les touches "CTRL" + "C".
4. Affichez le code du champ combo "TVA".
5. Placez-vous dans le traitement "Sélection d'une ligne", et utilisez les touches "CTRL"+"V".
6. Testez votre fenêtre et le fonctionnement de la combo.



Maintenant, notre fenêtre fonctionne correctement. **Mais le même code est utilisé à 2 endroits différents! Comment gérer les modifications et leur report? Le plus simple est d'utiliser une procédure. C'est ce que nous allons faire maintenant.**

Utiliser une procédure

La création d'une procédure à partir d'un code existant est très simple avec WinDev : une option de menu se charge de tout.

- Pour créer la procédure de calcul de la TVA :
 1. Sélectionnez le code présent dans le traitement "Sélection d'une ligne" de la combo "TVA".
 2. Sélectionnez l'option de menu "Code .. Refactoring .. Créer une procédure locale contenant le code sélectionné".
 3. Donnez le nom de la procédure à créer : CalculTVA. Validez.
 4. La procédure locale "CalculTVA" est automatiquement créée avec le code sélectionné. Votre code a été remplacé par l'appel à la procédure "CalculTVA()".
 5. Pour terminer, il suffit de remplacer le second code de calcul de la TVA (présent dans le code "A chaque modification" du champ "Prix HT") par l'appel à la procédure "CalculTVA()".
 6. Voilà, c'est terminé, vous pouvez enregistrer et tester votre fenêtre pour vérifier si tout fonctionne correctement.

Création d'une fenêtre avec un menu


Voilà, nous venons de réaliser quelques fenêtres. Nous allons maintenant créer la fenêtre principale de cet exemple. Cette fenêtre contiendra le menu organisant l'accès aux différentes fenêtres de l'application.

WinDev permet de créer des menus déroulants.

Un menu est toujours associé à une fenêtre. Ainsi, pour créer un menu, il faut en premier créer la fenêtre affichant ce menu.

Décrire le menu

- Pour créer une fenêtre comportant un menu :

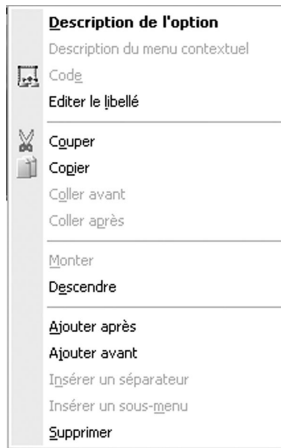
1. Cliquez sur l'icone  pour créer une nouvelle fenêtre.
2. Sélectionnez "Fenêtre".
3. Dans l'assistant qui s'affiche, sélectionnez "Vierge" et validez.
4. Donnez un nom et un titre et une description à cette fenêtre. Cette fenêtre a pour nom "FEN_Menu" et pour titre et description "Menu principal".
5. Enregistrez la fenêtre.
6. Sélectionnez l'option de menu "Fenêtres .. Menu Principal .. Ajouter le menu principal". Un menu s'insère dans la fenêtre sous la barre de titre. Ce menu contient par défaut une option qui se nomme "Menu".

Remarque : Chaque option de menu est saisie directement dans le menu.

Chaque option comporte une lettre d'appel. Cette lettre d'appel permet d'accéder directement à l'option avec la combinaison de touches [Alt]+Lettre. Le caractère '&' doit alors précéder la lettre d'appel (par exemple "&Fenêtres").

- Pour créer une option de menu et pour la modifier, il suffit d'utiliser le menu contextuel du menu ou de l'option de menu. Pour afficher le menu contextuel d'une option de menu :

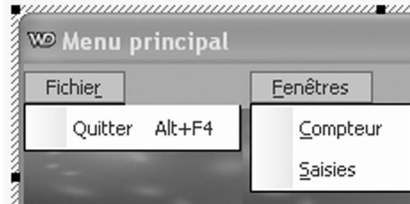
1. Sélectionnez le menu.
2. Cliquez avec le bouton droit de la souris.
3. Le menu contextuel suivant s'affiche :



Plusieurs choix sont possibles. Les principales options de gestion de menu sont :

- "Description de l'option" pour modifier le libellé de l'option et la lettre d'appel.
- "Code" pour saisir le code source correspondant au traitement à exécuter lors du clic sur cette option.
- "Ajouter après" pour ajouter une option de menu **après** l'option en cours.
- "Ajouter avant" pour ajouter une option de menu **avant** l'option en cours.
- "Insérer un sous-menu" pour insérer dans l'option en cours un sous-menu (arborescence).

► Réalisez le menu suivant :



Pour le menu "Fichier .. Quitter" :

1. Affichez la description de l'option "Menu" insérée par défaut, et changez le libellé en "Fichier&r". Validez la fenêtre de description.



Notes

Rappel : Le "&" permet de définir la lettre d'appel de l'option de menu.

2. Dans le menu contextuel de l'option "Fichier", sélectionnez l'option "Insérer un sous-menu" et saisissez "Quitter".

Pour le menu "Fenêtres" :

1. Sélectionnez l'option "Fichier".
2. Dans le menu contextuel de l'option "Fichier", sélectionnez l'option "Ajouter après" et saisissez "&Fenêtres".
3. Dans le menu contextuel de l'option "Fenêtres", sélectionnez "Insérer un sous-menu" et saisissez "&Compteur".
4. Dans le menu contextuel de l'option "Compteur", sélectionnez "Ajouter après" et saisissez "&Saisies".

Nous allons maintenant ajouter le raccourci [ALT + F4] pour l'option "Quitter".

1. Sélectionnez l'option "Quitter".
2. Affichez la description de l'option (option "Description de l'option" du menu contextuel).
3. Dans la zone "Raccourci clavier", sélectionnez "F4" et cochez la case "Alt".
4. Validez.

Lorsque toutes les options de menu sont saisies, visualisez la fenêtre (option "Affichage .. Afficher la fenêtre" ou la combinaison de touches [SHIFT]+[F9]). Ce mode de visualisation permet de visualiser l'aspect final du menu. Pour fermer cette fenêtre, utilisez le bouton "Fermer" de la télécommande de l'aperçu.



Astuce

Dans vos applications, nous vous conseillons de suivre le standard des menus Windows. Veuillez en particulier aux points suivants :

- Chaque option du menu principal doit proposer au moins une sous-option.
- La première lettre de chaque option doit être en majuscules, même si ce n'est pas la lettre d'appel. La lettre d'appel est soulignée (utilisez le caractère "&").
- Si un choix de menu appelle une fenêtre de traitement, terminez le libellé par trois points (...).
- L'option "Quitter" doit être la dernière sous-option de la première option de la barre de menu (exemple : "Fichier .. Quitter").
- Si une option d'aide est présente, elle doit se trouver à la fin de la première ligne de menu (l'option la plus à droite).

Associer un code aux options de menu

Chaque option terminale de menu doit comporter un traitement associé. Classiquement, ce traitement ouvre une fenêtre, mais n'importe quel type de traitement peut être exécuté.

- ▶ Sélectionnez une option du menu, effectuez un clic droit dessus et sélectionnez l'option "Code".
- ▶ Dans l'éditeur de code, saisissez le code pour chaque option de menu.

```

FEN_Menu - Sélection du menu de _Menu.OPT_Fenêtres.O...
Sélection du menu de _Menu.OPT_Fichier.OPT_Quitter Si Err...
  // Ferme la fenêtre
  Ferme

Sélection du menu de _Menu.OPT_Fenêtres.OPT_Compteur S
  Ouvre(FEN_Compteur)

Sélection du menu de _Menu.OPT_Fenêtres.OPT_Saisies Si...
  Ouvre(FEN_Saisie)
  
```

Vous l'aurez deviné : la fonction **Ouvre** permet d'afficher une fenêtre pour effectuer une saisie. La fonction **Ferme** quant à elle, ferme la fenêtre en cours.

Si la fenêtre contenant le menu est la première fenêtre affichée par l'application, la fermeture de la fenêtre contenant le menu (par l'option "Fichier .. Quitter" dans notre exemple) équivaut à quitter l'application.

- ▶ Enregistrez et testez la fenêtre.
- ▶ Sélectionnez une des options de menu.
- ▶ Cliquez sur l'option "Fichier .. Quitter" pour quitter le programme.

LEÇON 1.3. BASES DE PROGRAMMATION

Ce que vous allez apprendre dans cette leçon ...

- Les différents types de variables
- Instructions de base du WLangage
- Procédures et fonctions
- Traitement des chaînes, des numériques et des monétaires
- Traitement des dates et des heures



Durée estimée : 30 min

Introduction

Dans la leçon précédente, nous avons un peu manipulé l'éditeur de code de WinDev et son langage, le WLangage. Cette leçon va vous permettre de revenir sur les bases de la programmation en WLangage, en présentant les sujets suivants :

- Déclaration des différents types de variables
- Instructions de base du WLangage
- Procédures et fonctions
- Traitements des chaînes
- Traitements des numériques
- Traitements des dates et des heures

Déclaration des différents types de variables

Les différents types de variables

WinDev propose une grande variété de types de variables (booléen, entier, réel, monétaire, chaîne, date, heure, durée, dateheure, variant, tableau, structure, ...).

La syntaxe à utiliser pour déclarer une variable est très simple : il suffit d'indiquer le nom de la variable et son type :

```
NomVariable EST UN(e) TypeVariable
```

Quelques exemples :

```
Indice est un entier
NomFournisseur est une chaîne
TablePrix est un tableau de 10 monétaires
I, J, K sont des entiers
Compteur est un entier = 120
B1 est un booléen = Faux
```

Pour plus d'informations tous les types de variables disponibles (types simples, types avancés, ...), consultez l'aide en ligne (mot-clé: "Types de données").

La déclaration des variables et leur portée

Le WLangage permet de manipuler deux types de variables :

- Les variables globales
- Les variables locales

Les variables globales peuvent être globales au projet ou à une fenêtre. Il suffit de déclarer ces variables :

- dans le code d'initialisation du projet, pour déclarer les variables globales au projet. Ces variables pourront être utilisées dans tous les traitements du projet et des éléments du projet (fenêtres, états, ...)
- dans le code de déclarations des globales de la fenêtre, pour déclarer les variables globales de la

fenêtre. Ces variables pourront être utilisées dans tous les traitements de la fenêtre et des éléments de la fenêtre.

Toutes les variables déclarées ailleurs sont des variables locales.



Notes

Si votre projet utilise une charte de programmation, le nom de toutes les variables sera automatiquement préfixé. Ce préfixage permet d'identifier simplement le type et la portée de vos variables :

- préfixe "g" pour les variables globales
- préfixe "n" pour le type entier
- préfixe "s" pour le type chaîne
- préfixe "r" pour le type réel, ...



Attention !

La plupart des développeurs est tentée de déclarer toutes ses variables en "global" dans un projet. Ce type de programmation est en effet "facile". Les variables étant toutes globales, elles sont alors manipulables depuis n'importe quel traitement.

Mais ce type de programmation est souvent la cause d'écrasement de variables et d'effets de bord indésirables.

Il est donc préférable de déclarer un nombre limité de variables globales et ensuite de déclarer des variables locales.

Si vous souhaitez partager des valeurs entre 2 fenêtres, il est vivement conseillé d'utiliser la méthode du "passage de paramètres", méthode que nous verrons un peu plus tard.

Instructions de base du WLangage

Le WLangage est un L5G riche, composé de :

- Fonctions WLangage
- Propriétés WLangage
- Mots-clés WLangage
- Instructions WLangage

Au fur et à mesure de votre formation, vous découvrirez les instructions nécessaires.

Dans cette leçon, nous allons simplement vous présenter les instructions de base, permettant de réaliser des conditions et des boucles ou simplement de saisir des commentaires.

Instructions conditionnelles

Le WLangage permet de gérer des instructions conditionnelles du type :

- **SI, SINON, FIN** pour effectuer un test sur une condition.
- **SELON, CAS, FIN** pour exécuter une ou plusieurs actions selon les différents résultats d'un test sur une condition.

Quelques exemples

```
SI Maximum > Hasard(1, 999) ALORS
  Info("Bravo, vous avez gagné !")
SINON
  Info("Dommage, vous avez perdu !")
FIN
```

```
SELON Jour
  CAS "Lundi"
    // Premier jour de la semaine
  CAS "Mercredi"
    // Troisième jour de la semaine
  CAS "Vendredi"
    // Cinquième jour de la semaine
  CAS "Samedi", "Dimanche"
    // C'est le week-end
  AUTRES CAS
    // c'est un autre jour
FIN
```

Instructions de boucle

Le WLangage propose plusieurs possibilités pour gérer des boucles :

- **POUR, FIN** pour un nombre d'itérations déterminé.
- **TANTQUE, FIN** pour un nombre d'itérations indéterminé et dont la condition de sortie de la boucle est testée au début de la boucle.
- **BOUCLE, FIN** pour un nombre d'itérations indéterminé et dont la condition de sortie de la boucle est testée dans la boucle. L'instruction **SORTIR** permet de sortir de cette boucle.



Notes

Le WLangage propose également des boucles de type **POUR TOUT, FIN** permettant de parcourir les éléments d'un champ, les chaînes de caractères, les enregistrements d'un fichier, ... Ces boucles avancées seront étudiées dans la suite de ce cours d'auto-formation.

Quelques exemples

```
Ind est un entier = 0
TANTQUE Ind<Maximum
  Ind ++
FIN
```

```
Ind est un entier = 0
BOUCLE
  Ind ++
  SI Ind>Maximum ALORS SORTIR
FIN
```

```
POUR Ind = 1 à 100    // Il n'est pas nécessaire de déclarer Ind
  Cpt = Cpt + 1
FIN
```

Les commentaires

Pour saisir des commentaires dans le code, il suffit de commencer le code par // (deux caractères /).

Exemple :

```
// Ceci est une ligne de commentaire
```



Astuce

Vous pouvez mettre en commentaires plusieurs lignes de code en sélectionnant au clavier (ou à la souris) les lignes à mettre en commentaires puis en appuyant sur les touches [Ctrl] / (du pavé numérique).

Vous réalisez l'opération inverse (enlever les commentaires) en sélectionnant au clavier (ou à la souris) les lignes à enlever des commentaires puis en appuyant sur les touches [Ctrl] [Shift] / (du pavé numérique).

Le mot-clé MoiMême

MoiMême est un mot-clé qui représente le nom du champ en cours. **MoiMême** permet de rendre un code local (traitement d'un champ, ...) ou global (procédure globale, classe, ...) indépendant du champ en cours.

```
MoiMême.Libellé = "Nouveau libellé"
```

Un exemple pratique sur les différentes instructions de base

Après avoir présenté les principales instructions de base, rien ne vaut un petit test réel!

- ▶ Lancez WinDev 12 si ce n'est déjà fait.
- ▶ Ouvrez le projet "WD Decouverte.WDP" (option "Fichier .. Ouvrir un projet"). Ce projet se trouve dans le sous-répertoire "AutoFormation\Corrigés\WD Decouverte" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_ProcedureBoucle.wdw" (option "Fichier .. Ouvrir" ou double-clic sur le nom de la fenêtre dans le volet du projet). Cette fenêtre présente différents exemples.



Astuce

Pour rechercher rapidement une fenêtre dans le projet en cours, utilisez la combinaison de touches [CTRL] + [E]. Une fenêtre s'affiche permettant de réaliser une recherche sur toutes les fenêtres contenant la suite de lettres saisies dans le champ de recherche.

- ▶ Lancez le test de cette fenêtre (icone "GO").
 1. Cliquez sur "Boucles".
 2. Testez les différents exemples.
 3. Cliquez sur "SI / SELON".
 4. Testez les différents exemples.

Procédures et fonctions

Définition

Comme nous l'avons vu dans la leçon précédente, lorsqu'un traitement est appelé plusieurs fois dans un projet ou dans une fenêtre, il est souvent intéressant de créer une procédure contenant ce traitement. Il suffit alors d'appeler la procédure à chaque fois que cela est nécessaire.

Côté programmation, il existe les procédures et les fonctions :

- Les **fonctions** renvoient un résultat.
- Les **procédures** servent à exécuter un traitement spécifique.

WinDev permet de gérer simplement les deux types de traitements, de manière identique : sous WinDev, il n'y a pas de différence entre une procédure et une fonction. C'est pourquoi dans la suite de ce cours, nous utiliserons uniquement le terme "procédure".

Vous avez la possibilité de créer des procédures "locales" et des procédures "globales".

Procédure locale

Une procédure "locale" est liée à une fenêtre et uniquement à cette fenêtre.

Une procédure, lorsqu'elle est locale, peut être utilisée uniquement dans les traitements de la fenêtre et des champs de la fenêtre dans laquelle elle a été déclarée. Elle fait partie de la fenêtre.

Procédure globale et collection de procédures

Les procédures "globales" sont contenues dans des "collections de procédures". Chaque "collection de procédures" est un fichier qui contient toutes les procédures globales qui lui sont associées. Ce fichier possède l'extension ".WDG".

Par exemple, une collection permet de regrouper les procédures selon un thème : CalculsTVA, CalculsFraisDePorts, ...

Par défaut, lors de la création du projet, une collection de procédures est créée. Cette collection de procédures a pour nom "Procédures globales de <nom_projet>".

Vous pouvez créer autant de collections de procédures que vous le désirez dans un projet.

Une collection de procédures permet de :

- partager des procédures globales entre plusieurs développeurs, pour un même projet.
- partager des procédures globales entre plusieurs projets. En effet, une même collection de procédures peut être utilisée dans plusieurs projets.

Comment choisir si une procédure est globale ou locale ?

Pour choisir si une procédure doit être globale ou locale, posez-vous la question suivante : "La procédure va-t-elle être utilisée uniquement dans cette fenêtre, ou peut-elle être appelée depuis une autre fenêtre ?"

- Si la procédure est appelée "uniquement depuis cette fenêtre", la procédure peut être "locale".
- Si la procédure peut être appelée "depuis plusieurs fenêtres", la procédure doit être "globale".

A propos du passage des paramètres

Dans le projet "Mes_Premières_fenêtres", vous avez pu voir qu'une procédure pouvait gérer des paramètres. Les paramètres peuvent être obligatoires ou optionnels.

Les paramètres "obligatoires" sont toujours définis avant les paramètres "optionnels". La déclaration d'un paramètre "optionnel" est effectuée en affectant une valeur par défaut lors de la déclaration du paramètre.

Exemple :

```
PROCEDURE MaProcédure (Param1, ParamOption = "Valeur par défaut")
```

Une fenêtre peut également renvoyer une valeur au traitement qui l'a appelée. Pour plus de détails, consultez l'aide en ligne sur la propriété **ValeurRenvoyée**.

Appel d'une procédure

Pour appeler une procédure dans un traitement, il suffit d'écrire son nom dans l'éditeur de code et si nécessaire d'indiquer des paramètres.

WinDev sait bien évidemment gérer les appels de procédures imbriquées.



Astuce

Dans l'éditeur de code, lorsque vous êtes positionné sur un nom de procédure, l'appui sur la touche de fonction [F2] permet de visualiser le code de cette procédure. L'appui simultané sur les touches [CTRL]+[F2] permet de revenir dans le code précédent sur le nom de la procédure.

Création d'une procédure

Une procédure peut être créée directement depuis l'éditeur principal de WinDev. Il suffit d'utiliser le volet "Code" en bas à gauche de l'éditeur.

- ▶ Pour créer une procédure locale :
 1. Sélectionnez le volet "Code" (si nécessaire affichez ce volet par l'option "Affichage .. Barre d'outils .. Code").
 2. Sélectionnez "Procédures locales".
 3. Cliquez avec le bouton droit dans la liste et sélectionnez l'option "Nouvelle procédure locale".
 4. Saisissez le nom de la procédure et validez. Ensuite, saisissez le code de la procédure.

- ▶ Pour créer une procédure globale :
 1. Sélectionnez le volet "Code" (si nécessaire affichez ce volet par l'option "Affichage .. Barre d'outils .. Code").
 2. Sélectionnez "Procédures globales".
 3. Sélectionnez la collection de procédures dans laquelle la procédure globale doit être créée.
 4. Cliquez avec le bouton droit dans la liste et sélectionnez l'option "Nouvelle procédure globale".
 5. Saisissez le nom de la procédure et validez. Ensuite, saisissez le code de la procédure.

Rappel : Il est également possible de créer directement une procédure à partir du code sélectionné dans l'éditeur de code (option "Créer une procédure .. Créer une procédure contenant le code sélectionné" du menu contextuel). C'est ce que nous avons fait dans la leçon précédente.



Astuce

Quand utiliser les procédures ?

- Lorsqu'un traitement est utilisé plusieurs fois dans une même fenêtre, il est conseillé d'utiliser une procédure locale à la fenêtre qui contiendra ce traitement.
- Lorsqu'un traitement est utilisé plusieurs fois dans un ensemble de fenêtres, il est conseillé d'utiliser une procédure globale au projet qui contiendra ce traitement.

Un exemple pratique sur les procédures et fonctions

Après avoir vu la théorie, un peu de pratique !

- ▶ Ouvrez si nécessaire le projet "WD Decouverte.WDP" (option "Fichier .. Ouvrir un projet"). Ce projet se trouve dans le sous-répertoire "AutoFormation\Corrigés\WD Decouverte" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_ProcedureBoucle.wdw" (option "Fichier .. Ouvrir"). Cette fenêtre présente différents exemples.



Astuce

Pour rechercher rapidement une fenêtre dans le projet en cours, utilisez la combinaison de touches [CTRL] + [E]. Une fenêtre s'affiche permettant de réaliser une recherche sur toutes les fenêtres contenant la suite de lettres saisies dans le champ de recherche.

- ▶ Lancez le test de cette fenêtre (icone "GO").
 1. Cliquez sur "Fonction" pour tester le fonctionnement d'une fonction.
 2. Cliquez sur "Procédure" pour tester le fonctionnement d'une procédure.
 3. Fermez la fenêtre (bouton "Fermer").
- ▶ Consultez la liste des différentes procédures locales (accessibles depuis le volet "Code"). En double-cliquant sur le nom de la procédure, le code de cette procédure est affiché dans l'éditeur de code.

Traitements des chaînes

Dans les fonctionnalités de base d'un langage de programmation, la manipulation des chaînes de caractères est une des fonctionnalités les plus importantes.

Le WLangage offre une palette impressionnante de fonctionnalités pour manipuler les chaînes de caractères : fonctions WLangage, opérateurs d'extraction, de concaténation, ...

Nous présentons ici les fonctions les plus courantes pour manipuler les chaînes de caractères. Pour plus de détails, consultez l'aide en ligne (mot-clé : "Chaîne de caractères").

Exemple pratique

- ▶ Ouvrez si nécessaire le projet "WD Decouverte.WDP" (option "Fichier .. Ouvrir le projet"). Ce projet se trouve dans le sous-répertoire "AutoFormation\Corrigés\WD Decouverte" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_ManipulationChaines.WDW" (option "Fichier .. Ouvrir").
- ▶ Testez la fenêtre.

Détails

Un champ de type texte (par exemple un libellé) peut être initialisé :

- soit directement avec la chaîne :

```
SAI_TEXT1 = "WinDev est un outil formidable"
```

- soit avec une variable de type chaîne :

```
ch est une chaîne
ch = "J'apprends à travailler avec WinDev"
SAI_TEXT2 = ch
```

Une chaîne peut être construite à partir de plusieurs autres chaînes. On parle alors de **concaténation de chaînes**.

Pour concaténer deux chaînes, il suffit d'utiliser l'opérateur "+".

```
// La fonction Info permet d'afficher le résultat à l'écran
Info(SAI_TEXT2 + SAI_TEXT1)
```

L'extraction d'une partie d'une chaîne peut être réalisée avec :

- les opérateurs [[et]] (attention il ne faut pas d'espace entre les crochets [[et]]).

```
Info(SAI_TEXT1[[1 à 6]]) //affiche "WinDev"
```

- la fonction **ExtraitChaîne** :

```
Info(ExtraitChaîne(SAI_TEXT1,1," ")) //affiche "WinDev"
```

- la fonction **Milieu**, qui extrait un morceau de chaîne depuis une chaîne :

```
Info(Milieu(SAI_TEXT2,29)) //affiche "WinDev"
```

- la fonction **Gauche**, qui retourne la partie gauche d'une chaîne :

```
Info(Gauche(SAI_TEXT2,10)) // affiche "J'apprends"
```

- la fonction **Droite**, qui retourne la partie droite d'une chaîne :

```
Info(Droite(SAI_TEXT1,10)) //affiche "formidable"
```

La taille d'une chaîne peut être connue avec la fonction **Taille** :

```
Info(Taille(SAI_TEXT2)) //affiche 35
```

Une chaîne peut être transformée en majuscules avec la fonction **Majuscule** ou en minuscules avec la fonction **Minuscule** :

```
Info(Majuscule(SAI_TEXT2))
Info(Minuscule(SAI_TEXT2))
```

Une chaîne peut être recherchée dans une autre avec la fonction **Position** :

```
ChaineRecherche est une chaîne = "WinDev"
Pos est un entier
Pos = Position(SAI_TEXT2, ChaineRecherche)
SI Pos=0 ALORS
    Info(ChaineRecherche + " non trouvé dans le texte 2")
SINON
    Info(ChaineRecherche + " trouvé dans texte 2")
FIN
```



Astuce

Vous pouvez également effectuer une recherche de position d'une chaîne de caractères dans une autre sans tenir compte de la casse. Il suffit d'indiquer à la fonction **Position** la constante **SansCasse**.

Exemple :

```
Pos = Position(TEXT2, ChaineRecherche, 1, SansCasse)
```

Pour connaître le nombre d'occurrences d'une chaîne de caractères donnée dans une autre chaîne de caractères, utilisez la fonction **ChaîneOccurrence** :

```
NbOccurrences est un entier
NbOccurrences = ChaîneOccurrence("anastasia", "a") // Renvoie 4
```

Traitements des numériques

Les calculs sur les numériques peuvent être effectués à partir des champs de saisie de type numérique ou directement en manipulant les variables typées (entier, réel, numérique, monétaire, ...)

Exemple pratique

- ▶ Ouvrez si nécessaire le projet "WD Decouverte.WDP" (option "Fichier .. Ouvrir le projet"). Ce projet se trouve dans le sous-répertoire "AutoFormation\Corrigés\WD Decouverte" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_CalculsNumeriques.WDW" (option "Fichier .. Ouvrir").
- ▶ Testez la fenêtre. Cette fenêtre présente un aperçu des manipulations sur les "numériques".

Détails

Un champ de saisie de type numérique peut être initialisé :

- soit directement avec la valeur numérique :

```
SAI_NUM1 = 3.14
```

- soit avec une variable numérique :

```
ent1 est un entier
ent1 = 1234
SAI_NUM1 = ent1
```

Il est possible de concaténer une chaîne et un numérique avec l'opérateur "+" :

```
Info("Le champ SAI_NUM1 contient la valeur : "+ SAI_NUM1)
```



Astuce

Si vous exécutez le code suivant :

```
Info("Un calcul : " + 1 + 2)
```

La boîte de dialogue affichera "Un calcul : 12".

Si vous souhaitez afficher le résultat du calcul, il faut utiliser le code suivant :

```
Info("Un calcul : " + (1 + 2))
```

La boîte de dialogue affichera "Un calcul : 3".

Quelques exemples de manipulations de numériques réalisés en WLangage :

- La partie entière d'un nombre est connue par la fonction **PartieEntière**, la partie décimale par la fonction **PartieDécimale** :

```
Info("Partie entière de "+SAI_NUM1+" : "+...
      PartieEntière(SAI_NUM1), ...
      "Partie décimale de "+SAI_NUM1+" : "+ ...
      PartieDécimale(SAI_NUM1))
```

- La valeur absolue d'un nombre est retournée par la fonction **Abs** :

```
Info("Valeur absolue de "+SAISIE2+" : "+Abs(SAISIE2))
```

- La valeur arrondie d'un nombre est retournée par la fonction **Arrondi** :

```
Info("Arrondi de "+SAISIE2 +" : " + Arrondi(SAISIE2,1))
```

- La fonction **Racine** permet de calculer la racine N^{ième} d'un nombre :

```
SAI_Racine = Racine(SAI_Puissance, 2)
```

- La fonction **Puissance** permet d'élever un nombre à la puissance N :

```
SAI_Puissance = Puissance(SAI_Racine, 2)
```

Remarque : le type des champs de saisie numériques

Lorsqu'un champ est défini comme numérique, on ne connaît pas a priori son type (entier, réel, réel double, ...). Son type est défini automatiquement en fonction du masque sélectionné pour le champ.

Pour forcer le type d'un champ, il suffit d'utiliser une variable typée. Par exemple :

```
Val1 est un entier
Val1 = 123456789 //affectation du champ
SAI_NUM1 = Val1
Val1 = SAI_NUM1 //récupération du champ
```

Traitement des monétaires

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_RéelsMonétaires.wdw".
- ▶ Testez la fenêtre. Cette fenêtre permet de tester les différents codes présentés dans ce paragraphe.

Détails

Le type "Monétaire" est un réel codé sur 10 octets. Il permet de donner la solution à deux problèmes non résolus par les réels :

- avoir plus de 15 chiffres significatifs (les réels doubles gèrent 15 chiffres significatifs maximum), le type "Monétaire" gère 23 chiffres significatifs.
- éviter une erreur induite par le codage binaire des réels :
 - Un réel gère jusqu'à 15 chiffres significatifs. Or, en réalité, le codage binaire des réels ne permet pas de coder tous les réels de 15 chiffres significatifs.
 - Le type "Monétaire" a un codage différent des réels qui n'induit pas d'erreur d'arrondi. En effet, le code suivant donne un résultat faux :

```
// Avec une variable de type Réel, le calcul est faux
x est un Réel
x = 18.6 - 8.6 - 10
Erreur ("18.6-8.6-10="+x)
```

Par contre avec le programme suivant, le calcul est correct :

```
x est un Monétaire
x = 18.6-8.6-10
Info(x)
```

Pour effectuer des divisions sur des monétaires, il est conseillé de passer par des variables intermédiaires de type "Monétaire".

```
x, y sont des Monétaires
x = 12345678901234567.123456
y = 12345678901234567.123456
x = x/y
Info ("21345678901234567.123456", ...
      "/", "12345678901234567.123456", "= "+x)
```



Remarque

Pour faire des calculs avancés, le WLangage met également à votre disposition le type **Numérique**. Par défaut, le type Numérique correspond à 32 chiffres pour la partie entière et 6 chiffres pour la partie décimale (comme le type monétaire). Mais l'avantage du type numérique est de pouvoir configurer le nombre de chiffres à utiliser pour la partie entière et le nombre de chiffres à utiliser pour la partie décimale.

Pour plus de détails, consultez l'aide en ligne (mot-clé : "Numérique").

Mélange de chaînes et de numériques

Les numériques et les chaînes font bon ménage. WinDev est très souple dans les affectations de variables. Ainsi il est possible d'affecter une chaîne de chiffres dans une variable numérique et inversement.

Par exemple :

```
i est un entier
c est une chaîne
i = 123
c = i //la variable c contient la chaîne "123"
c = "456"
i = c //la variable i contient la valeur 456
```

Si vous voulez transformer un nombre en une chaîne de caractères en respectant un format bien précis, il suffit d'utiliser la fonction **NumériqueVersChaîne**.

Par exemple :

```
NumériqueVersChaîne(1234.567, "12,3f") // renvoie "000001234,567"
```

- ▶ Ouvrez la fenêtre "FEN_NumeriqueVersChaine.wdw". Cliquez par exemple sur son nom dans le volet WinDev (onglet "Projet", option "Fenêtres").
- ▶ Testez la fenêtre.

A chaque fois que vous devez utiliser la fonction **NumériqueVersChaîne** et que vous ne savez pas exactement quels paramètres utiliser, vous pouvez exécuter cette fenêtre : elle vous donnera la syntaxe à utiliser en fonction du résultat souhaité.



Notes

La fonction "inverse" de *NumériqueVersChaîne*, est la fonction *Val*. Cette fonction permet de convertir une chaîne en un numérique.

Traitements des dates et des heures

Présentation

Pour gérer simplement les dates et les heures dans vos applications, WinDev met à votre disposition :

- **un champ de saisie de type Date, Heure ou Durée.** Avec ce champ, plus aucun problème pour saisir une date ou une heure valide.
- **des variables de type Date, Heure, DateHeure et Durée.** Ces variables simplifient la manipulation des dates et des heures par programmation et permettent de multiples calculs.

Exemple pratique

- ▶ Ouvrez si nécessaire le projet "WD Decouverte.WDP" (option "Fichier .. Ouvrir le projet"). Ce projet se trouve dans le sous-répertoire "AutoFormation\Corrigés\WD Decouverte" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_DateHeure.WDW". Cette fenêtre illustre les explications données dans la suite de cette leçon.

Masque de saisie et valeur retournée

Les dates et les heures sont des chaînes qui ont un format prédéfini.

Dans un champ de saisie de type "Date" ou de type "Heure", il faut distinguer :

- Le **masque de saisie** : c'est le format avec lequel la date et l'heure seront affichées.
- La **valeur retournée** (ou mémorisée) : c'est la valeur retournée par le champ vers le programme.

Ces informations sont saisies dans la fenêtre de description du champ de saisie :

Par exemple, pour un champ de type "Date" :

- le masque de saisie sera "JJ/MM/AAAA". La date saisie par l'utilisateur sera de la forme "01/11/2007".
- la valeur retournée sera par défaut "AAAAMMJJ" (Exemple : la valeur saisie sous la forme "23/04/2007" retournera au programme "20070423").



Remarque

Il existe la possibilité de choisir un masque de saisie de type "Date système". Dans ce cas, le format affiché sera celui défini dans les "Paramètres régionaux du panneau de configuration de Windows" de la machine en cours d'exécution.

Pour comprendre l'importance et les relations entre le masque de saisie et la valeur retournée, regardons les exemples suivants :

- Si le masque de saisie d'un champ date est "JJ/MM/AA" et la valeur retournée est "AAAAMMJJ" :

```
DATEJ="19981225" //affiche la date sous la forme "25/12/98"
DATEJ="981225" //affiche la date sous une forme incorrecte
```

La date affichée sera incorrecte.

- Si le masque de saisie d'un champ date est "MM/JJ/AAAA" et la valeur retournée est "AAMMJJ" :

```
DATEJ="981225" //affiche la date sous la forme "12/25/1998"
DATEJ="19981225" //affiche la date sous une forme incorrecte
```

La date affichée est incorrecte.

Il faut donc faire attention au format de la valeur retournée dans un champ de type date ou heure.

Les fonctions du WLangage qui manipulent :

- des dates utilisent le format "AAAAMMJJ".
- des heures utilisent le format "HHMMSSCC" (certaines utilisent le format "HHMMSSCCMMM", pour la gestion des millisecondes).




Notes

Rappel : Par défaut, le format (masque de saisie) du champ correspond au masque date défini par le projet (option "Projet .. Description du projet", onglet "Langues"). Ainsi, le même masque est utilisé automatiquement dans tous les champs de type date de l'application.

Cette fonctionnalité est également très utile dans les applications multilingues.

Les dates

Exemple pratique

- ▶ Ouvrez si nécessaire le projet "WD Decouverte.WDP" (option "Fichier .. Ouvrir le projet"). Ce projet se trouve dans le sous-répertoire "AutoFormation\Corrigés\WD Decouverte" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_ManipDates.WDW". Cette fenêtre illustre les explications données dans la suite de cette leçon.
- ▶ Testez cette fenêtre (bouton )

Quel jour sommes-nous?

Pour connaître la date du jour, il suffit d'utiliser la fonction **DateDuJour** (ou **DateSys**). La fonction **DateDuJour** retourne la date système de votre ordinateur sous la forme d'une chaîne de caractères au format "AAAAMMJJ".

Exemple :

```
Info("Nous sommes le " + DateDuJour())
```

- Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Quel jour sommes-nous ?".

Comme vous pouvez le constater, la date affichée est au format "AAAAMMJJ".

Pour afficher la date sous un format plus explicite, il suffit d'utiliser la fonction **DateVersChaîne** :

```
Info("Nous sommes le " + DateVersChaîne(DateDuJour()))
```

La fonction **DateVersChaîne** transforme une chaîne au format "AAAAMMJJ" en chaîne au format "JJ/MM/AAAA" (à la norme française et beaucoup plus lisible !).

- Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Quel jour sommes-nous (bis, mais en clair) ?".

Quel jour sommes-nous, mais avec le jour et le mois en lettres ?

Pour afficher la date en toute lettre, il faut utiliser les fonctions **EntierVersJourEnLettre** et **EntierVersMoisEnLettre** :

```
varDate, varDateJ sont des Chaînes
varDateJ = DateDuJour()
varDate = EntierVersJourEnLettre(DateVersEntier(varDateJ)) + ...
           " " + varDateJ[[7 à]] + " " + ...
           EntierVersMoisEnLettre(DateVersEntier(varDateJ))
Info("Nous sommes le "+varDate)
```

Les fonctions **EntierVersJourEnLettre** et **EntierVersMoisEnLettre** donnent en toutes lettres le jour et le mois correspondant à la date passée en paramètre.

Les fonctions **EntierVersJourEnLettre** et **EntierVersMoisEnLettre** attendent un entier. Il faut donc utiliser la fonction **DateVersEntier** pour convertir une date "chaîne" en un entier.

- Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Quel jour sommes-nous (ter, mais avec des lettres) ?".

Quel est le nombre de jours entre 2 dates ?

Vous voulez savoir combien de jours se sont écoulés entre deux dates ? C'est très simple : il suffit d'utiliser la fonction **DateDifférence** :

```
NombreJour est un entier
NombreJour = DateDifférence(DateSys(), "20100101")
Info("Le nombre de jours entre", "le " + ...
      DateVersChaîne(DateSys()) + " et le 01/01/2010 est " + ...
      NombreJour)
```



Astuce

Pour calculer l'âge d'une personne, utilisez la fonction **WLangage Age**.

- Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Nombre de jours entre deux dates".

Quel est le jour de la semaine d'une date donnée ?

Toujours aussi simplement, vous pouvez connaître le jour de n'importe quelle date avec la fonction **EntierVersJourEnLettre** :

```
J est une chaîne
J = EntierVersJourEnLettre(DateVersEntier("17890714"))
Info("Le 14 juillet 1789 était un " + J)
```

Remarque : Il est également possible d'utiliser directement la fonction **DateVersJourEnLettre**.

```
J = DateVersJourEnLettre("17890714")
```

- ▶ Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Jour de la semaine".

Petit exercice pratique

Maintenant que vous maîtrisez les dates, une colle : que fait le code suivant ?

```
DateVersChaîne(DateDuJour(),"JJJJ JJ MMM AAAA")
```

Réponse : Affiche la date littérale (le jour et la date en clair).

Les heures

Exemple pratique

- ▶ Ouvrez si nécessaire le projet "WD Decouverte.WDP" (option "Fichier .. Ouvrir le projet"). Ce projet se trouve dans le sous-répertoire "AutoFormation\Corrigés\WD Decouverte" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_ManipHeures.WDW". Cette fenêtre illustre les explications données dans la suite de cette leçon.
- ▶ Testez cette fenêtre (bouton "GO").

Quelle heure est-il ?

Pour connaître l'heure, il suffit d'utiliser la fonction **Maintenant** (ou **HeureSys**). La fonction **Maintenant** retourne l'heure actuelle de votre système sous la forme d'une chaîne de caractères au format "HHMMSSCC" :

```
Info("Il est " + Maintenant())
```

- ▶ Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Quelle heure est-il ?".

Vous souhaitez afficher l'heure au format standard "HH:MM:SS:CC" ? Le code devient alors :

```
Info("Il est " + HeureVersChaîne(Maintenant()))
```

La fonction **HeureVersChaîne** transforme une heure au format "HHMMSSCC" en une chaîne au format "HH:MM:SS:CC".

- ▶ Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Quelle heure est-il (bis mais en clair)?".

Pour ne pas afficher les centièmes de seconde, le code est :

```
Heure est une chaîne = Gauche(HeureVersChaîne(Maintenant()),8)
Info("Il est "+Heure)
```

La fonction **Gauche** retourne les n premiers caractères d'une chaîne (8 dans notre exemple).

- Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Quelle heure est-il (ter sans les centièmes)?".

Combien de temps s'est écoulé entre deux heures ?

Il est 17h25. Combien de temps s'est écoulé depuis 12h15 ? Le code est le suivant :

```
// Affiche le temps écoulé depuis 12h15
Diff est un entier = HeureDifférence("1215", Maintenant())
Signe_temps est une chaîne
Résultat est une chaîne = EntierVersHeure(Abs(Diff))

SI Diff < 0 ALORS Signe_temps = "-"
Info("Temps écoulé : " + Signe_Temps + Gauche(Résultat, 2) + ...
    "heures"+ Milieu(Résultat, 3, 2) + "minutes"+...
    Milieu(Résultat, 5, 2) + "secondes"+...
    Milieu(Résultat, 7, 2) + "centièmes de seconde")
```

La fonction **EntierVersHeure** transforme un entier (qui correspond au nombre de centièmes de secondes écoulées depuis minuit (ou 00h00)) vers une heure sous la forme "HHMMSSCC".

La fonction **HeureVersEntier** effectue l'opération inverse.



Attention !

- Ne mettez pas de ":" dans l'heure passée en paramètre à la fonction **HeureVersEntier**, cela fausserait le résultat.
- Pour calculer des durées de plus de 24 heures, utilisez les types Date, Heure, ...

- Pour tester cette fonctionnalité dans notre exemple, cliquez sur le bouton "Temps (en secondes) écoulé depuis 12H15".

Variable de type Date, Heure, DateHeure, Durée

Le WLangage propose des types de variables spécifiques pour les valeurs de type Date, Heure, DateHeure ou de Durée.

Ces variables sont assimilables à des "chaînes". Chacune contient une valeur selon le tableau suivant :

Type	Format par défaut
Date	AAAAMMJJ
Heure	HHMMSSCC
DateHeure	AAAAMMJJHHMMSSCC
Durée	AAAAMMJJHHMMSSCC

Le type "Date" permet de gérer des dates comprises entre le 01/01/0001 et le 31/12/9999 (de quoi être tranquille pour un bon moment !).

Plus sérieusement, pour manipuler la valeur de ces variables, vous pouvez utiliser les syntaxes suivantes :

```
MaDate est une Date = "20021021"  
Info(DateVersChaine(MaDate)) //Affiche "21/10/2002"  
MaDate..Année = MaDate..Année + 1  
MaDate..Mois = MaDate..Mois + 1  
MaDate..Jour = MaDate..Jour + 1  
Info(DateVersChaine(MaDate)) //Affiche "22/11/2003"
```

Dans ce code, **Année**, **Mois** et **Jour** sont des propriétés WLangage.

Calculs avec des dates et des heures

Le WLangage possède de très nombreuses fonctions pour gérer les dates et les heures et effectuer différents calculs.

Pour plus de détails, consultez l'aide en ligne (mot-clé : "Date, Fonctions WLangage").

LEÇON 1.4. QUESTIONS / RÉPONSES

Ce que vous allez apprendre dans cette leçon ...


- Questions / Réponses




Durée estimée : 5 mn

Questions / Réponses

Question Comment visualiser l'élément auquel appartient le traitement en cours ?

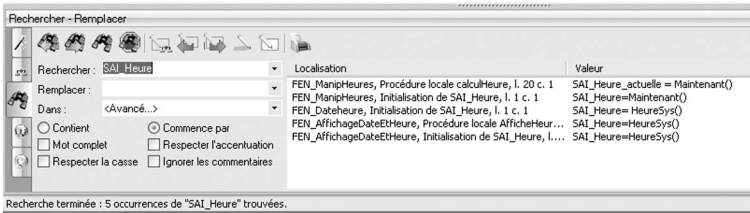
Pour visualiser l'élément correspondant au traitement en cours, cliquez sur l'icone . La fenêtre contenant l'élément voulu s'affiche.

Question Comment imprimer le code source ?

Le code source en cours peut être imprimé directement en cliquant sur l'icone  dans la barre d'icônes de l'éditeur ou en sélectionnant l'option "Fichier .. Imprimer le dossier", ou la combinaison de touches [CTRL] + [P].

Question Comment rechercher et/ou remplacer une variable dans le code ?

Les fonctions de recherche ou de remplacement dans le code sont accessibles depuis le menu de WinDev (option "Edition .. Rechercher" ou "Edition .. Remplacer") ou dans le volet "Rechercher - Remplacer" en cliquant sur :



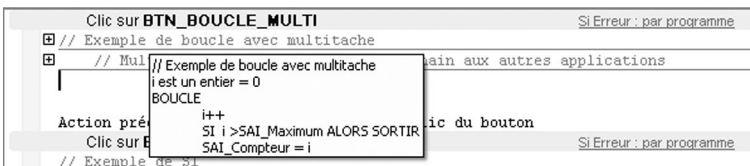
Vous pouvez accéder à la recherche à tout moment en appuyant sur les touches [CTRL]+[F].

Question Que signifie les signes "+" ou "-" dans l'éditeur de code ?

L'éditeur de code permet d'enrouler ou de dérouler le code WLangage. Cette fonctionnalité est très utile si vos traitements utilisent de nombreuses instructions structurées (boucles, condition, parcours, ...).

Pour replier un code, sélectionnez l'option de menu "Code .. Code repliable .. Replier tout" (ou le raccourci [CTRL] + [SHIFT] + * (du clavier numérique)).

Seuls les commentaires restent visibles. Le survol de chaque ligne de commentaire permet de visualiser le code associé dans une bulle :



La combinaison de touches [CTRL] + * (du clavier numérique) permet de déplier tout le code. Le clic sur les symboles "-" ou "+" permettent d'enrouler ou de dérouler uniquement la partie de code correspondante.

Question Est-il possible d'identifier la personne qui a écrit un code ?

L'appui de la touche [F6] permet d'afficher les informations (nom et date de la création/modification) de chaque ligne de code.

Question Existe-t-il un moyen d'obtenir facilement la syntaxe ou de l'aide sur une fonction ?

Lors de la frappe d'une fonction, la syntaxe de la fonction s'affiche :

- dans une bulle d'aide au-dessous de la ligne en cours de saisie. Pour chaque paramètre (y compris le résultat de la fonction), une bulle explicative est affichée. Si plusieurs syntaxes sont disponibles, il est possible de passer d'une syntaxe à l'autre grâce aux touches [ALT] + flèche droite ou flèche gauche.
- dans la barre de message de l'éditeur.

Dans l'aide affichée, les paramètres entourés par [et] sont optionnels.

Pour les fonctions utilisant des noms de fichiers, de champs, de fenêtres, ou d'états, la saisie assistée permet d'afficher la liste des éléments du projet correspondant au paramètre de la fonction en cours de saisie.

Exemple de saisie assistée pour la fonction **HLitPremier** :

L'option <Assistant> permet de lancer un assistant de code. Cet assistant vous pose différentes questions sur l'utilisation de la fonction et génère automatiquement le code correspondant.

Toutes les fonctions et les propriétés du WLangage ont une aide associée. Cette aide est directement accessible depuis l'éditeur en appuyant sur la touche [F1] sur le nom de la fonction ou de la propriété voulue.

Question Comment utiliser un glossaire sous l'éditeur de code ?

Le glossaire de fonctions proposé par WinDev permet d'associer un raccourci aux fonctions du WLangage le plus souvent utilisées. Il suffit ensuite de saisir ce raccourci sous l'éditeur de code.

Par exemple au lieu de taper "EcranVersFichier", il suffit de taper le code abrégé "EVF".

- ▶ Pour utiliser le glossaire, il suffit de :
 - taper les trois lettres identifiant la fonction,
 - appuyer sur la touche [F5], le nom de la fonction est automatiquement inscrit.

Le contenu du glossaire peut être modifié, par exemple pour ajouter le nom des variables que vous utilisez fréquemment.

- ▶ Pour modifier le contenu du glossaire, appuyez sur les touches [SHIFT]+[F5].

Question **Quels sont les raccourcis clavier à connaître sous l'éditeur de code ?**

- [CTRL]+[L] supprime la ligne en cours.
- [CTRL]+[D] duplique la ligne en cours sur la ligne du dessous.
- [TAB] et [SHIFT]+[TAB] permettent de gérer l'indentation de l'ensemble des lignes sélectionnées.
- [CTRL]+[/] met en commentaires les lignes sélectionnées, [CTRL]+[SHIFT]+[/] enlève les commentaires (Attention touche [/] du pavé numérique).
- [F2] sur le nom d'un champ, d'une méthode, d'une classe, d'une procédure ou d'un bloc d'état affiche le traitement de cet objet.
- [CTRL]+[F2] permet de revenir au traitement initial.
Il est possible par l'appui répété sur la touche [F2] de se déplacer d'un traitement à un autre. Pour revenir sur le traitement initial, il suffit d'appuyer autant de fois sur les touches [CTRL]+[F2].
- [CTRL]+[R] permet de réindenter automatiquement le code affiché.



DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 2

Application avec données

XIII ***Express***

LEÇON 2.1. PRÉSENTATION

Ce que vous allez apprendre dans cette leçon ...

- Présentation de l'application réalisée dans cette partie



Durée estimée : 20mn

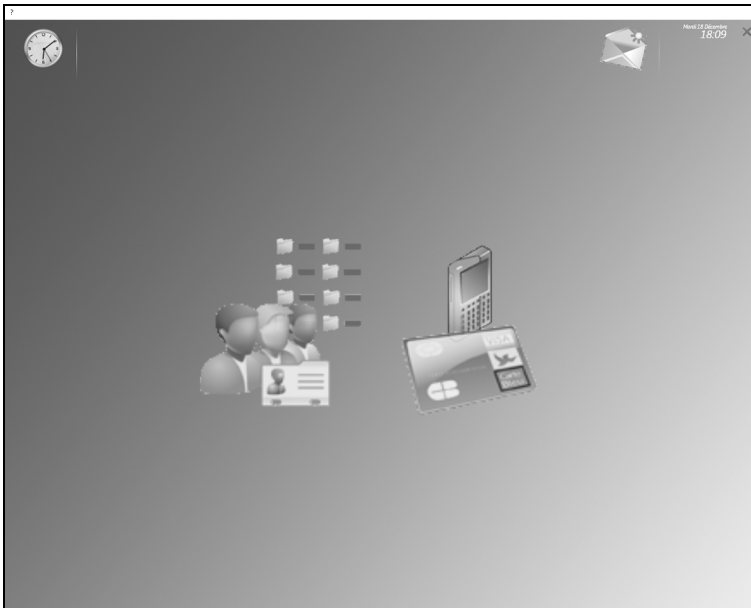
Présentation de l'application réalisée dans cette partie

Dans cette partie, nous allons étudier les bases de données pouvant être manipulées avec WinDev. Dans un but pédagogique, nous vous proposons de développer une application de A à Z, de la conception de l'analyse à la distribution de l'application.

Vous verrez ainsi tous les points importants du développement d'une application.

L'application que nous allons réaliser ensemble est une gestion de comptes bancaires : vous allez pouvoir suivre en temps réel l'état de vos comptes ... quand l'application sera réalisée. La base de données utilisée est Hyper File Classic, la base de données gratuite fournie avec WinDev. Plus loin dans ce cours, nous manipulerons la base de données Hyper File Client / Serveur.

Mais avant de se mettre au travail, voyons ce que vous allez faire. Le projet complet de l'application est disponible dans le répertoire "\\Autoformation\\Corrigés\\Mes Comptes". Vous pouvez lancer directement l'application par l'option "? .. Autoformation .. Application avec données (mes comptes)" du menu de WinDev.



LEÇON 2.2. PROJET ET ANALYSE

Ce que vous allez apprendre dans cette leçon ...

- Création d'un projet
- Création d'une analyse



Durée estimée : 20mn

Présentation

Pour créer une application avec une base de données, il faut :

- Créer le projet lié à l'application. Ce projet regroupera tous les éléments de l'application (fenêtres, codes, requêtes, états, ...).
- Créer l'analyse liée au projet. L'analyse permet de décrire tous les fichiers de données manipulés par l'application.

Nous créerons ensuite l'application grâce au module RAD (Rapid Application Development).

Création du projet

► Pour créer le projet :

1. Lancez WinDev si ce n'est pas déjà fait.
2. Sélectionnez l'option "Fichier .. Nouveau", survolez le "P" puis cliquez sur "Projet". L'assistant de création de projet se lance. Les différents plans de l'assistant vont vous aider à créer votre projet. Tous les renseignements indiqués dans cet assistant pourront être modifiés par la suite.
3. Le premier plan de l'assistant permet de saisir le nom du projet, son emplacement et sa description. Dans notre cas, ce projet va simplement s'appeler "Mes Comptes". Par défaut, WinDev propose de créer ce projet dans le répertoire "\\Mes projets\Mes Comptes". Vous pouvez conserver cet emplacement ou le modifier grâce au bouton [...]. Pour le résumé du projet, saisissez "Le projet a pour but de gérer des comptes bancaires".



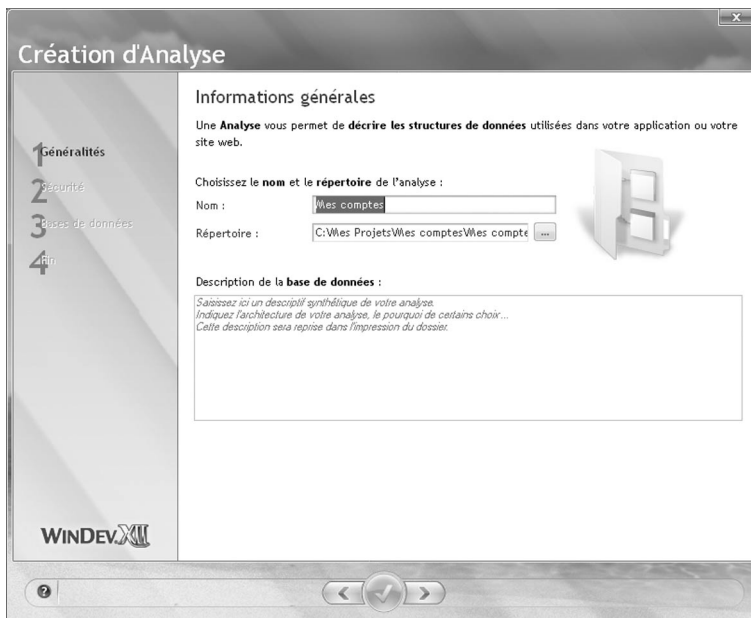
4. Les différentes étapes de l'assistant sont indiquées à gauche de l'assistant. Ces étapes sont directement clicables. Les autres écrans de l'étape 1 ("Description") n'étant pas fondamentaux, cliquez directement sur "2-Equipe".

5. Dans ce projet, nous travaillerons seul (le travail en équipe sera présenté plus loin).

6. Cliquez sur "3-Chartes". Cette étape permet de définir la charte de programmation et la charte graphique. Nous avons déjà vu l'intérêt de la charte de programmation, nous l'utilisons dans ce projet. Sélectionnez l'option "Oui, utiliser la charte de programmation ci-dessous" et passez à l'écran suivant.
7. Nous allons maintenant choisir la charte graphique. Sélectionnez "ActivVista" et passez à l'écran suivant.
8. Nous choisissons une taille de 800x600 pour les écrans, car notre application n'aura pas de grandes fenêtres, et sera ainsi adaptée à la plupart des résolutions utilisées. Le choix serait différent pour une application de gestion d'images par exemple.
9. Nous allons maintenant donner les renseignements concernant la base de données. Cliquez directement sur "5-Base de données".
10. Sélectionnez l'option "Oui, créer une nouvelle base de données" et validez. L'assistant de création de l'analyse se lance.

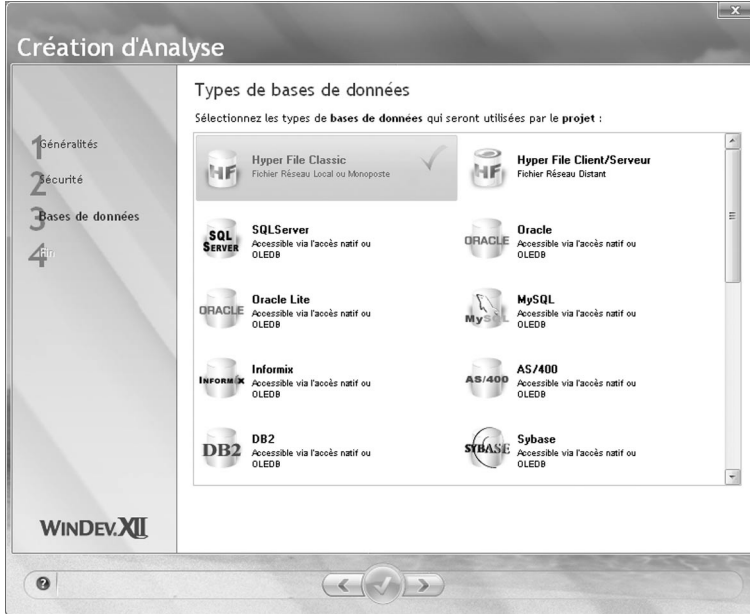
Création de l'analyse

- Les étapes de l'assistant de création d'analyse sont les suivantes :
 1. Indiquez le nom et le répertoire de l'analyse. Par défaut, le nom de l'analyse correspond au nom du projet et le répertoire de l'analyse est un répertoire ".ana" dans le répertoire du projet. Nous allons conserver ces paramètres par défaut. Passez à l'écran suivant de l'assistant.



2. L'écran suivant concerne la sécurité de l'analyse, avec l'utilisation d'un mot de passe. Nous verrons cet aspect plus loin dans ce cours. Passez à l'écran suivant.
3. Vous pouvez ensuite choisir le ou les types de bases de données manipulées par le projet. Sélectionnez Hyper File Classic (la base de données proposée par défaut avec WinDev). Passez

à l'écran suivant de l'assistant.

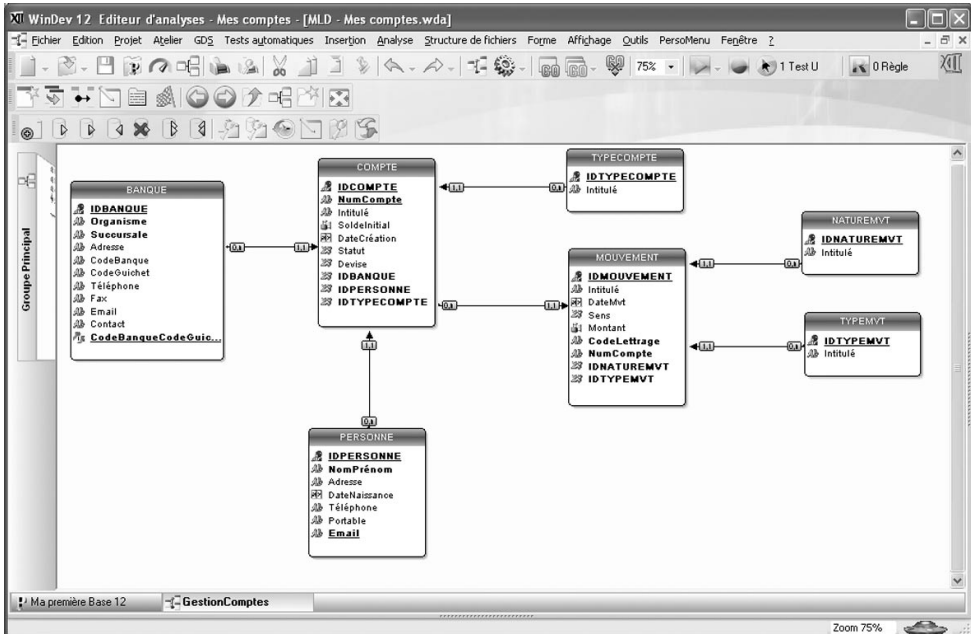


4. Validez. L'assistant de création de fichier se lance automatiquement.

Création de la description des fichiers de données

Notre application de gestion de compte va être associée à l'analyse suivante. Cette analyse comporte sept fichiers différents :

- BANQUE :
- COMPTE
- TYPECOMPTE
- PERSONNE
- MOUVEMENT
- NATUREMVT
- TPEMVT



Pour créer les différents fichiers de cette analyse, nous allons utiliser différentes méthodes disponibles dans WinDev.

Création d'un fichier de données et de ses rubriques sous l'éditeur

- ▶ Les étapes de l'assistant de création de fichier sont les suivantes :
 1. Sélectionnez dans l'assistant l'option "Créer une nouvelle description de fichier". Passez à l'écran suivant de l'assistant.
 2. Le fichier que nous allons créer est le fichier "COMPTE". Son nom est donc "COMPTE". Ce nom sera utilisé :
 - pour manipuler le fichier en programmation. La variable associée au fichier sera compte.
 - pour construire le nom du fichier de données physique associé (fichier COMPTE.fic).
 Automatiquement, le libellé et la représentation du fichier apparaissent.



Notes

"Un enregistrement représente" indique l'information qui sera utilisée pour décrire les liaisons entre fichiers. Il est important de renseigner avec soin cette information !

3. Conservez l'option "Le fichier possède un identifiant automatique". Cette option indique si le fichier doit posséder une clé unique, gérée automatiquement par WinDev.



Notes

Pour créer l'identifiant (un identifiant est une clé unique), vous pouvez créer une rubrique numérique de type "Identifiant automatique".

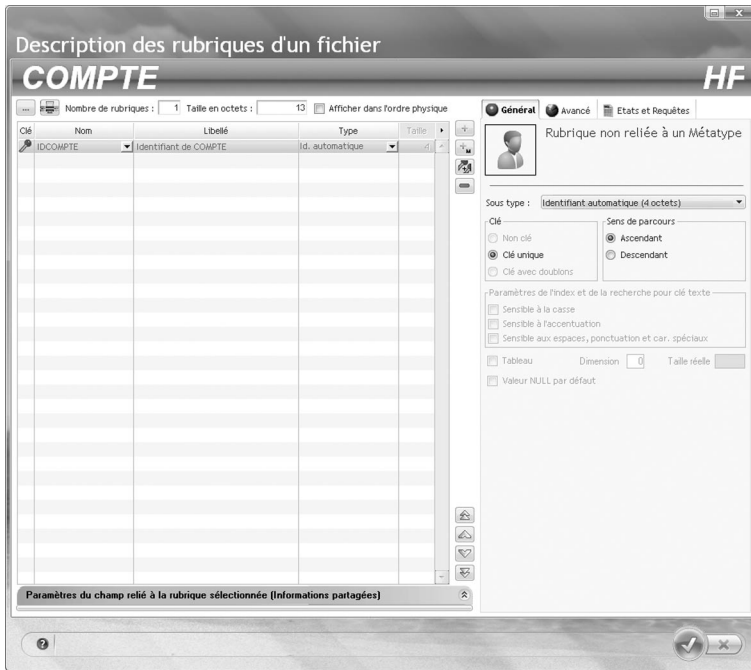
Cet identifiant est automatiquement géré par WinDev. Chaque fois qu'un enregistrement est ajouté dans le fichier, WinDev affecte automatiquement une valeur à l'identifiant du fichier. Cette valeur est unique.

Vous pouvez décocher ce choix si vous n'avez pas besoin d'identifiant automatique (si aucune clé unique n'est nécessaire, ou si une clé unique existe déjà dans le fichier).

4. Passez à l'écran suivant et sélectionnez le type de la base associée au fichier. Nous allons travailler sur des fichiers Hyper File Classic. Passez à l'écran suivant.

5. Conservez les options par défaut et passez à l'écran suivant.

6. Cliquez sur le bouton vert pour valider. Le fichier est automatiquement créé dans l'analyse. La fenêtre de description des rubriques s'ouvre.

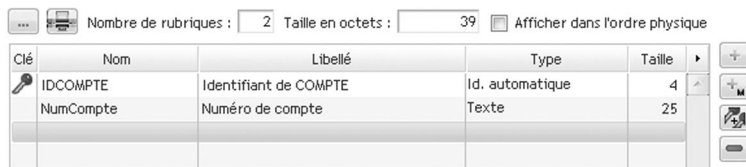


Nous allons saisir les rubriques du fichier COMPTE. Dans la fenêtre de description du fichier, vous pouvez déjà voir qu'une rubrique a été automatiquement créée : IDCOMPTE. Cette rubrique correspond à l'identifiant automatique du fichier. Cette rubrique est composée des lettres "ID" et du nom du fichier.

Nous allons créer les autres rubriques de ce fichier.

- ▶ Tout d'abord, nous allons créer la rubrique "NumCompte". Cette rubrique contiendra le numéro de compte.

1. Dans la fenêtre de description des rubriques, cliquez dans la colonne "Nom" de la première ligne vide. Cette colonne passe automatiquement en saisie. Saisissez "NumCompte".
2. Cliquez dans la colonne "Libellé". Automatiquement, le nom de la rubrique apparaît. Nous allons modifier le libellé de la rubrique en saisissant "Numéro de compte". Dans la colonne "Type", le type "Texte" est automatiquement sélectionné. Nous n'y touchons pas.
3. Nous allons modifier la taille de la rubrique. Cliquez sur la case "50" et remplacez "50" par "25". Cliquez sur la ligne suivante. Les valeurs sont automatiquement mises à jour.



4. Cette rubrique va être une clé de notre fichier : les clés permettent d'accélérer les accès aux données ou de faciliter les parcours de fichiers. Il sera possible de faire des recherches ou des

parcours sur cette rubrique.



Notes

La notion de clé fait partie des caractéristiques d'une rubrique. En fait, lors de la création d'une rubrique, vous avez la possibilité d'indiquer si elle est :

- non clé,
- clé unique : la valeur de cette clé sera unique dans l'ensemble du fichier (c'est-à-dire dans tous les enregistrements du fichier)
- clé avec doublons : la valeur de cette clé pourra être retrouvée plusieurs fois dans le fichier.

5. La définition de la clé se fait de la manière suivante : re-sélectionnez la ligne de la rubrique "NumCompte" pour activer les champs de description présents sur la droite de l'écran. Il suffit alors de préciser le type de clé utilisé. Dans notre cas, le numéro de compte est une clé unique.

► De la même manière, créez les rubriques suivantes (ces rubriques ne sont pas des clés) :

Nom	Libellé	Type	Taille
Intitulé	Intitulé	Texte	50
SoldeInitial	Solde initial	Monétaire	---

► Nous allons maintenant créer les rubriques "Statut" et "Devise".

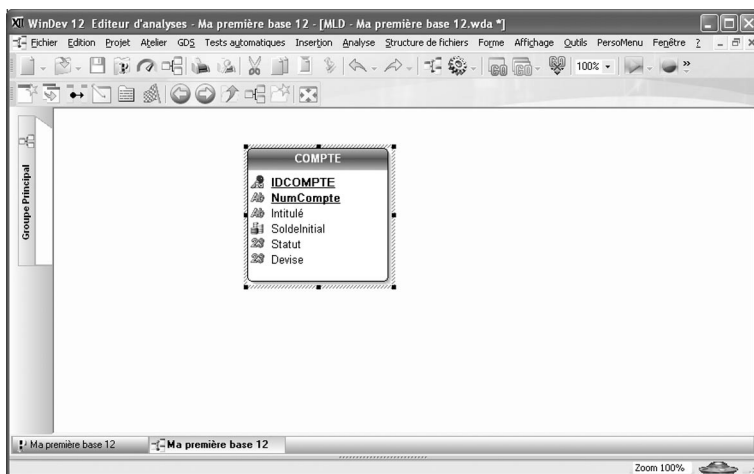
1. Positionnez-vous sur une nouvelle ligne de la table. Saisissez :

- le nom : Statut
- le libellé : Statut du compte
- le type : Sélecteur, liste, Combo. Dans la fenêtre qui s'ouvre, vous pouvez sélectionner le type de champ utilisé par défaut pour cette rubrique. Ici, ce sera un sélecteur.

2. Dans la partie basse de l'écran, cliquez sur la double-flèche pour afficher les paramètres du champ lié à la rubrique sélectionnée. Les informations saisies ici seront automatiquement utilisées lors de la création des fenêtres liées au fichier. Vous retrouvez ici le type du champ et le libellé. Nous allons saisir les options suivantes :

- Actif (remplace Option1)
- Suspendu (remplace Option2)
- Clôturé (utilisez la touche ENTREE pour créer cette option).

3. Cliquez à nouveau sur la double-flèche.
4. De la même manière, saisissez la rubrique "Devise". Cette rubrique est liée à une combo et contient les informations suivantes :
 - EUR - Euro
 - USD - Dollar Américain
 - JPY - Yen Japonais
 - AUD - Dollar Australien
 - CAD - Dollar Canadien
5. Voilà, la description du fichier COMPTE est terminée. Fermez la fenêtre de description des rubriques. Le fichier COMPTE apparaît sous l'éditeur d'analyse.



Maintenant que vous devenez "expert" en création de fichier, créez le fichier "PERSONNE". Pour lancer l'assistant de création de fichier, il suffit de sélectionner l'option "Insertion .. Fichier". Le fichier PERSONNE a un identifiant automatique et contient les rubriques suivantes :

Nom	Libellé	Type et taille
NomPrénom	Nom et Prénom de la personne	Texte, 50, clé avec doublons
Adresse	Adresse	Texte, type mémo
DateNaissance	Date de Naissance	Date, 8
Téléphone	Téléphone	Texte, 20
Portable	Portable	Texte, 20

Nous allons ajouter une petite nouveauté dans ce fichier. Nous allons créer une rubrique email.



Facile ! Mais pour cela, nous allons utiliser les métatypes.



Notes


Métatype

Un métatype est une combinaison de caractéristiques d'une rubrique et de son champ relié. Par exemple, un métatype "Fax" contiendra le type et la longueur de la rubrique, le masque de saisie, le cadrage, ...

Vous pouvez utiliser les rubriques métatypes fournies avec WinDev, ou créer vos propres métatypes.

Pour utiliser une rubrique métatype, il suffit de cliquer sur le bouton "Métatypes" dans la description des rubriques de l'analyse.

► Pour ajouter une rubrique créée à partir d'un métatype :

1. Cliquez sur le bouton .
2. Sélectionnez "Adresse email".
3. Validez.
4. Cette rubrique est une clé unique.
5. Fermez la fenêtre de description de fichier.

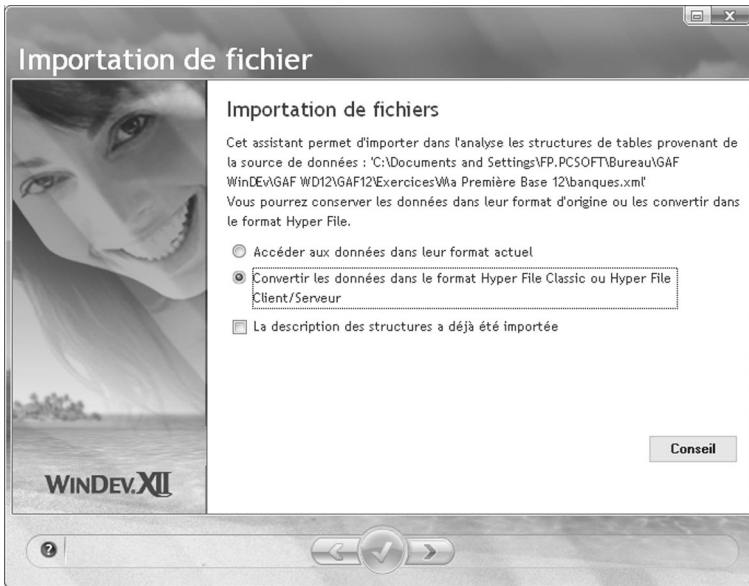
Importation d'un fichier XML

Pour créer le fichier BANQUE, contenant les caractéristiques des banques, nous allons utiliser une autre méthode : l'import d'un fichier XML. À partir du fichier XML contenant les données, WinDev va créer la description de fichier dans l'analyse ainsi que le fichier Hyper File correspondant avec les données présentes dans le fichier XML.

► Pour importer un fichier XML dans l'analyse :

1. Dans l'explorateur de fichiers de Windows, ouvrez le sous-répertoire de WinDev suivant : "`\autoformation\exercices\Mes Comptes`".
2. Sélectionnez le fichier "Banques.xml".
3. Réalisez un "Drag and Drop" du fichier "Banques.xml" vers l'éditeur d'analyse de WinDev.

L'assistant d'importation de fichier se lance.



4. Nous allons convertir le contenu du fichier XML au format Hyper File. Cochez l'option "Convertir les données dans le format Hyper File Classic ou Hyper File Client / Serveur". Passez à l'écran suivant de l'assistant.

5. WinDev sélectionne automatiquement l'accès natif à utiliser ("Accès Natif XML pour WinDev") et indique le chemin du fichier à importer. Passez à l'écran suivant de l'assistant.

6. Sélectionnez la table "Banque" à gauche et cliquez sur la flèche ">". "Banque" apparaît dans la colonne "Tables à importer". Il est possible d'effectuer un aperçu du fichier. Passez à l'écran suivant de l'assistant.

7. Validez l'assistant. L'assistant de conversion des données se lance.


8. Validez le répertoire de création des fichiers de données Hyper File. Passez à l'écran suivant de l'assistant.

9. Décochez "Enregistrer les paramètres de la conversion" et validez. Le fichier apparaît dans l'analyse.

► Voyons la description du fichier importé :

1. Sélectionnez le fichier "Banque" et sélectionnez l'option "Description du fichier" du menu contextuel.

2. Ce fichier ne contient pas d'identifiant automatique : cochez l'option "Identifiant automatique".

3. Cliquez sur l'icone  pour afficher la description des rubriques du fichier.

4. Toutes les rubriques ont été importées sous forme de chaîne de 255 caractères, nous allons modifier ces informations :

Rubrique	Taille
Organisme	20
Succursale	50
Adresse	Sous type : Memo texte
Code Banque	5
Code guichet	5
Telephone	20
Fax	20
Email	260
Contact	50

Nous obtenons les données suivantes :

The screenshot shows a window titled 'Banque' with a table of fields. At the top, there are controls for 'Nombre de rubriques : 10' and 'Taille en octets : 226', along with a checkbox 'Afficher dans l'ordre physique'. The table has columns for 'Clé', 'Nom', 'Libellé', 'Type', and 'Taille'.

Clé	Nom	Libellé	Type	Taille
IDBanque	Identifiant de Banque	Identifiant de Banque	Id. automatique	4
	Organisme	Organisme	Texte	20
	Succursale	Succursale	Texte	50
	Adresse	Adresse	Texte	8
	CodeBanque	CodeBanque	Texte	5
	CodeGuichet	CodeGuichet	Texte	5
	Telephone	Telephone	Texte	20
	Fax	Fax	Texte	20
	Email	Email	Texte	260
	Contact	Contact	Texte	50

► Pour simplifier les recherches dans le fichier, nous allons créer une clé composée.



Notes

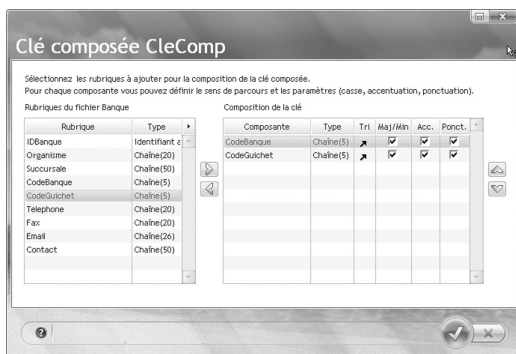
Une clé composée est une clé qui est composée de plusieurs rubriques du même fichier.

Ce type de clé permet de parcourir le fichier sur des critères plus complexes ou de faire des recherches plus précises.

Cette clé sera composée du code banque et du code guichet.

1. Cliquez sur
2. Sélectionnez "Code Banque" et cliquez sur ">".

3. Sélectionnez "Code Guichet" et cliquez sur ">".



4. Validez. La clé composée apparaît dans la description de fichier.

5. Validez la description du fichier "BANQUE".

6. Validez la description des fichiers.

Importation de fichiers du dictionnaire

Pour simplifier la création des fichiers de l'analyse, nous avons préparé une description de fichiers que nous avons placée dans un dictionnaire. Nous allons voir comment importer rapidement ce fichier.

WinDev propose un volet spécifique, le volet "Dictionnaire", afin de manipuler simplement et facilement les dictionnaires.

► Pour importer des éléments depuis un dictionnaire :

1. Affichez si nécessaire le volet "Dictionnaire" (option "Affichage .. Barre d'outils .. Dictionnaire").
2. Cliquez dans le volet et sélectionnez l'option "Ouvrir" du menu contextuel.
3. Sélectionnez le fichier "Dico_GAF" présent dans le sous-répertoire de WinDev suivant : "\autoformation\exercices\Mes comptes". Le dictionnaire apparaît dans le volet. Ce dictionnaire contient le fichier "Mouvement".
4. Effectuez un "Drag and Drop" du fichier "Mouvement" vers l'éditeur d'analyse.
5. L'éditeur demande si un abonnement est nécessaire. Répondez "Non". Nous verrons plus en détail le dictionnaire dans la suite de ce cours.
6. Validez. La description du fichier "Mouvement" est intégrée à l'analyse.

Importation directe de fichiers de données existant

La dernière méthode que nous allons voir pour créer des fichiers de données est la simple importation de fichiers Hyper File existants. Nous avons préparé les derniers fichiers sous cette forme.

► Pour importer des fichiers de données Hyper File :

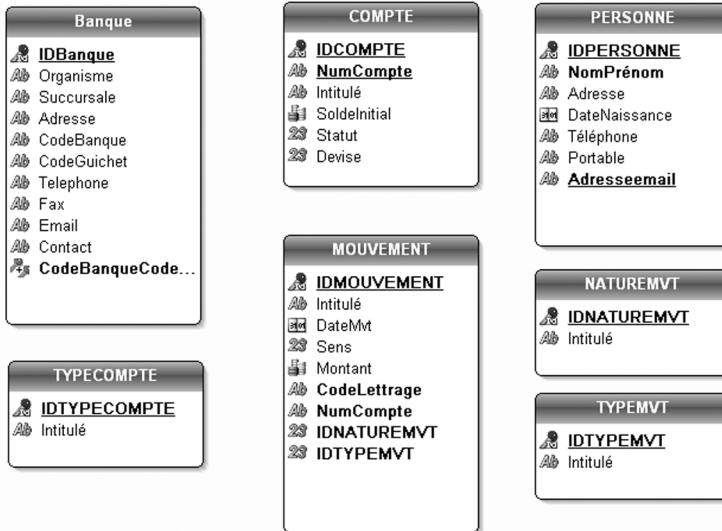
1. Dans l'explorateur de fichiers de Windows, ouvrez le sous-répertoire de WinDev suivant : "\autoformation\exercices\Mes comptes".
2. Sélectionnez le fichier "TypeMvt.fic".
3. Réalisez un "Drag and Drop" du fichier "TypeMvt" vers l'éditeur d'analyse de WinDev.

4. L'assistant d'importation se lance. Validez les différents plans. Le fichier apparaît sous l'éditeur d'analyse.
5. Répétez cette opération avec les fichiers Typecompte.fic et Naturemvt.fic.

Tous les fichiers nécessaires sont maintenant présents dans l'éditeur d'analyse.

Création des liaisons

Nous avons créé toutes les descriptions de fichier nécessaire à l'application de gestion de compte.



Maintenant nous allons créer les liaisons entre les fichiers.

- ▶ Créons tout d'abord la liaison entre le fichier BANQUE et le fichier COMPTE.
 1. Sélectionnez l'option "Insertion .. Liaison". Le curseur de la souris se transforme en crayon.
 2. Cliquez sur le fichier "BANQUE" puis sur le fichier "COMPTE".
 3. L'assistant de création de liaison se lance.

4. Répondez aux questions posées par l'assistant:



- Chaque Banque a au moins un compte : Non
- Chaque Banque peut avoir plusieurs comptes : Oui
- Chaque Compte a au moins une banque : Oui
- Chaque Compte peut avoir plusieurs banques : Non



Notes

Vous pouvez également saisir directement les cardinalités de la liaison dans l'assistant.

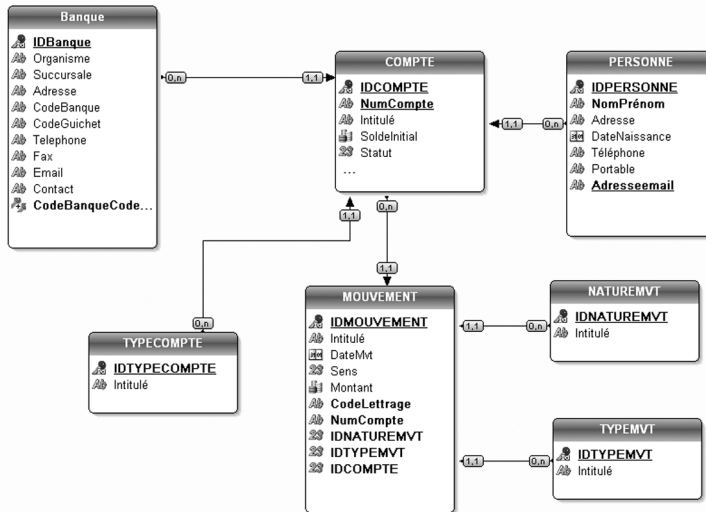
5. Passez à l'écran suivant. L'assistant propose automatiquement la clé utilisée par la liaison. Affichez l'écran suivant de l'assistant.
6. L'assistant propose de créer une nouvelle clé dans le fichier Compte. Acceptez cette option en passant à l'écran suivant.
7. Validez les règles d'intégrité en passant à l'écran suivant.
8. Cliquez sur la flèche verte. La liaison est automatiquement créée dans l'éditeur d'analyse.

► Vous pouvez créer les liaisons suivantes :

Fichier source	Fichier relié	Cardinalités
Personne	Compte	0,n - 1,1
Compte	Mouvement	0,n - 1,1
TypeCompte	Compte	0,n - 1,1
NatureMvt	Mouvement	0,n - 1,1
TypeMvt	Mouvement	0,n - 1,1



L'analyse est alors la suivante :



Configuration de l'analyse pour le RAD

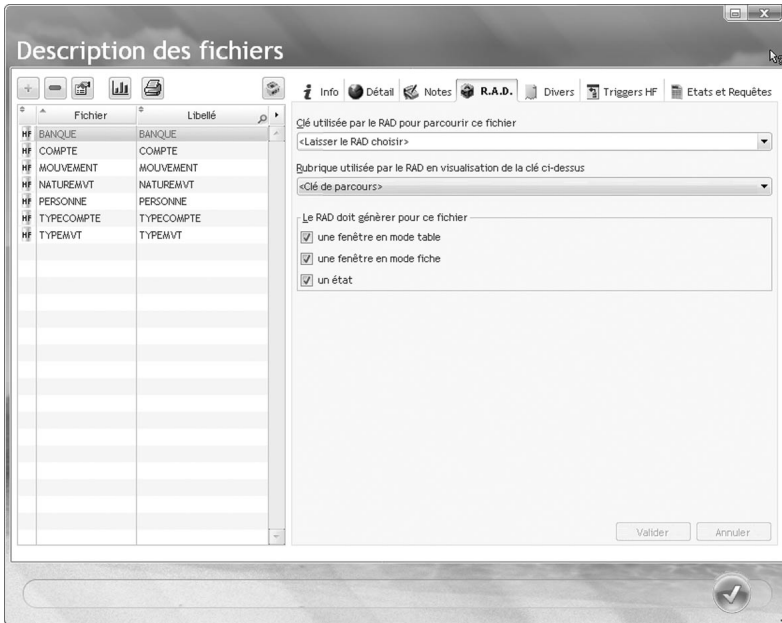
Lors de la génération de l'application, de fenêtres, d'états, le module RAD (Rapid Application Development) utilise des options spécifiées directement dans l'analyse. Il est nécessaire de consacrer un peu de temps à la configuration de ces options pour obtenir le meilleur résultat possible automatiquement. Les éléments utilisés par le RAD sont définis :

- dans la description des fichiers (onglet RAD)
- dans la description des rubriques (onglet RAD)
- dans les informations partagées spécifiées pour chaque rubrique.

Configuration du RAD dans la description de fichiers

- Pour configurer les options RAD des fichiers :
 1. Sélectionnez l'option "Structures de fichiers .. Description".

2. Affichez l'onglet "RAD".



Les informations à configurer pour chaque fichier sont :

- La rubrique à afficher pour la visualisation
- Les éléments générés par le RAD pour le fichier.

Les informations à configurer pour chaque fichier sont les suivantes :

<i>Fichier</i>	<i>Rubrique à visualiser</i>	<i>Élément à générer</i>
BANQUE	Organisme	Table, Fiche
COMPTE	Intitulé	Table, Fiche, Etat
MOVEMENT	Intitulé	Table, Fiche, Etat
NATUREMVT	Intitulé	Table, Fiche, Etat
PERSONNE	NomPrénom	Table, Fiche
TYPECOMPTE	Intitulé	Table, Fiche, Etat
TYPEMVT	Intitulé	Table, Fiche, Etat

- Pour réaliser ces modifications :
 1. Sélectionnez le fichier dans le tableau.
 2. Réalisez les modifications dans l'onglet RAD.
 3. Cliquez sur le bouton "Valider".
 4. Passez au fichier suivant.

Configuration du RAD dans la description des rubriques

- Pour configurer les options RAD des rubriques :
 1. Pour chaque fichier, affichez la description des rubriques (option "Description des rubriques" du menu contextuel).
 2. Affichez l'onglet "Avancé".
 3. Configurez les options RAD.
 Pour chaque fichier, les identifiants des fichiers doivent avoir les caractéristiques suivantes :

Le RAD doit générer pour cette rubrique : _____

une colonne en mode table

un champ en mode fiche

un champ pour les états

En effet, l'identifiant ne doit pas être affiché sur les fenêtres en mode fiche et les états. Laissez tout coché pour les autres rubriques.

4. Validez la description des rubriques du fichier.

L'analyse est prête à être utilisée par le RAD.

Génération de l'analyse

La génération de l'analyse consiste à rendre disponible aux autres modules du projet, les informations sur les fichiers. Ces fichiers pourront être manipulés dans les programmes.

Sans la génération, bien que la description du fichier existe, vous ne pourrez pas utiliser le fichier dans vos programmes.

Lors de la génération de l'analyse, toutes les modifications effectuées sur l'analyse et les fichiers de données seront automatiquement propagées dans tout le projet (fenêtres, champs reliés, états, ...).



Attention!

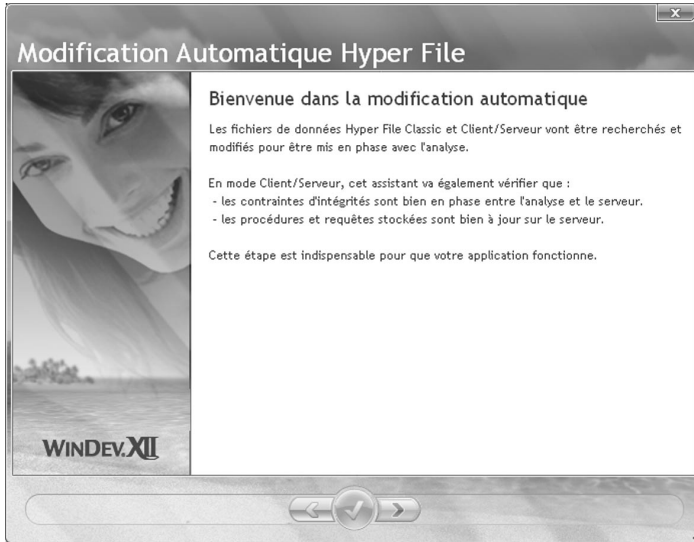
La génération doit être effectuée à chaque fois que vous voulez que les modifications effectuées dans l'analyse soient prises en compte dans les programmes utilisant cette analyse.

Si vous modifiez l'analyse plusieurs fois, sans programmer entre chaque modification, il n'est pas nécessaire de générer l'analyse à chaque modification. Générez uniquement l'analyse lorsque toutes les modifications sont faites et que vous passez à la programmation.

- Pour générer l'analyse :
 1. Sous l'éditeur d'analyse, sélectionnez l'option "Analyse .. Génération".
 2. La génération de l'analyse est automatiquement lancée.

Nous avons modifié les descriptions des fichiers de données présents dans l'analyse (fichier BANQUE). Pour mettre à jour les fichiers de données de l'application, WinDev lance automatiquement la procédure de modification des fichiers de données. Cette opération permet de mettre à jour les fichiers de données (fichiers .fic) en fonction de leur description dans l'analyse.

► L'assistant de modification automatique se lance :



Validez les différents plans jusqu'à l'exécution de la modification automatique des données. Fermez l'éditeur d'analyse. Maintenant, nous pouvons commencer à programmer l'application.

LEÇON 2.3. LE RAD COMPLET

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce le RAD ?
- Génération du RAD
- Test du projet



Durée estimée : 20mn

Qu'est-ce que le RAD ?

R.A.D. sont les initiales de "Rapid Application Développement".

Le RAD permet de construire automatiquement une application, c'est-à-dire construire toutes les fenêtres, les états et les requêtes nécessaires.

Comme nous l'avons vu précédemment, pour développer une application en WinDev, il faut commencer par créer un projet et éventuellement une analyse. L'analyse contient la définition des structures des fichiers manipulés dans les traitements.

Le module RAD de WinDev s'appuie sur cette analyse. Le module RAD est constitué d'un assistant dans lequel vous pouvez détailler les différents paramètres nécessaires au développement du projet :

- La présentation (modale ou MDI).
- Le code généré (procédural ou objet).
- Le type de programmation utilisé (Hyper File ou SQL).

Toutes les fenêtres, les états, les requêtes et le code générés sont entièrement personnalisables. Vous avez également la possibilité de modifier les types de champs, les valeurs par défaut, ...

Le RAD permet également de générer des fenêtres de plusieurs types, c'est le RAD fenêtre.

Nous allons ici découvrir comment utiliser le module RAD Projet.



Note

RID (Rapid graphical Interface Design)

WinDev permet également de générer des fenêtres contenant uniquement les champs liés aux rubriques de l'analyse. Tout le code nécessaire au fonctionnement de ces fenêtres reste à la charge du développeur.

Pour plus de détails, consultez l'aide en ligne (mot-clé : "RID").

Génération du RAD

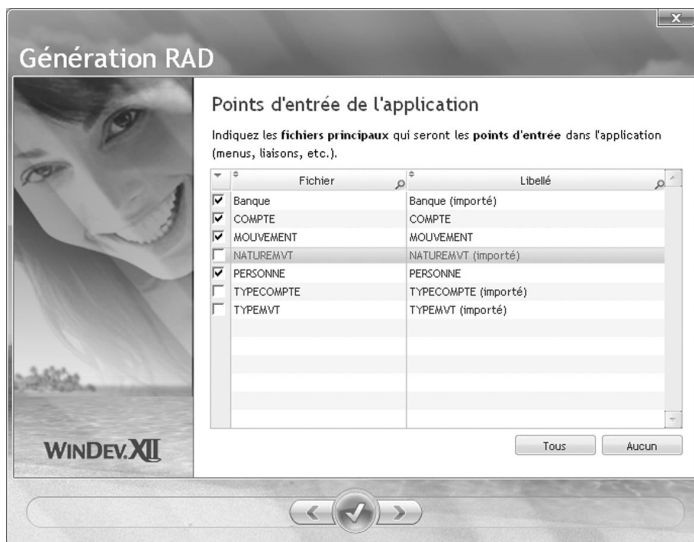
- ▶ Pour lancer la génération du RAD :
 1. Sélectionnez l'option "Atelier .. RAD application complète". L'assistant de génération de l'application RAD se lance.
 2. Sélectionnez le modèle à utiliser pour la génération du RAD : par exemple "Pattern RAD Jet Tames". Passez à l'écran suivant de l'assistant.

3. Tous les fichiers de l'analyse vont être pris en compte :



Affichez l'écran suivant.

4. Les points d'entrée de l'application correspondent aux entrées disponibles dans le menu.



Conservez uniquement les fichiers Banque, Compte, Mouvement et Personne. Affichez l'écran suivant.

5. Pour chaque fichier, il est possible d'associer une image. Cette image sera par exemple utilisée dans le menu, ... Cette image peut être une image sélectionnée dans le catalogue d'images de WinDev.



Note

Le catalogue d'images est une bibliothèque de plus de 4500 images pouvant être intégrées dans vos applications. Nous verrons plus loin dans ce cours comment utiliser le catalogue d'images.

Passez à l'écran suivant.

6. Indiquez si le groupware utilisateur doit être utilisé dans l'application. Nous l'intégrerons plus tard dans l'application. Passez à l'écran suivant.

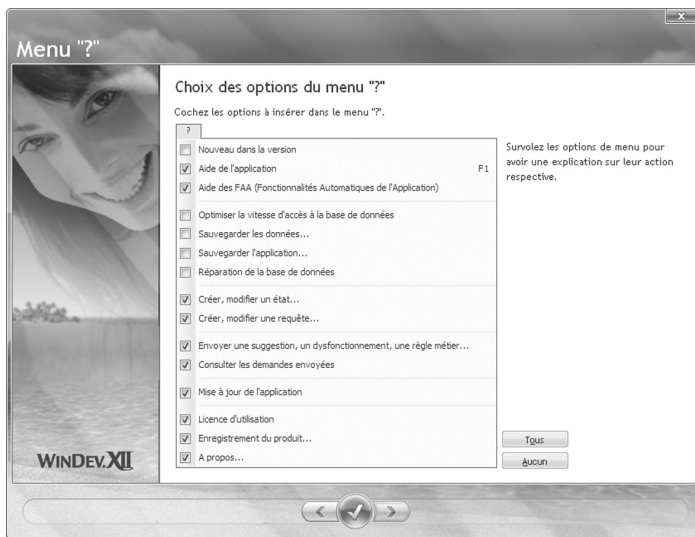
7. Indiquez si le menu automatique doit être intégré à l'application. Répondez "OUI". Passez à l'écran suivant. L'assistant de génération du menu automatique se lance.



Note

Le menu automatique est un menu d'aide adapté à vos applications. Ce menu permettra aux utilisateurs de vos applications d'accéder directement à plusieurs fonctionnalités.

► Les différentes étapes de l'assistant de génération du menu automatique sont les suivantes :



1. Conservez les options "Aide des FAA (Fonctionnalités automatiques de l'application)" et "Envoyer une suggestion...".

WinDev 12 version "Express" : La fonctionnalité "Envoyer une suggestion..." n'est pas disponible dans cette version d'évaluation.

2. Répondez aux diverses questions et terminez l'assistant.



Notes

Lors de la création du menu "? " :

- l'option "? .. Aide des fonctionnalités automatiques" a ajouté automatiquement le composant "CCMenu" dans votre application. En effet, cette option utilise une procédure de ce composant pour fonctionner.
- l'option "? .. Suggestion, dysfonctionnement..." a ajouté automatiquement le composant "FeedBack" dans votre application. Ce composant permet à l'utilisateur d'envoyer ses retours concernant une application. Ces retours client sont automatiquement récupérés par le centre de Suivi de Projets et le centre de Suivi de Qualité et des Evolutions.

Cette option fonctionne uniquement à partir de l'exécutable (elle ne fonctionne pas en mode test).


WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

L'application est générée et peut être testée.

Test de l'application

Testons immédiatement l'application générée.

► Pour lancer le test de l'application :

1. Cliquez sur le bouton . L'application se lance.
2. Sélectionnez dans le menu "Personne".
3. La fiche concernant les personnes s'affiche.
4. Cliquez sur le lien "Ajouter" pour ajouter une nouvelle personne (vous par exemple !).
5. Validez l'enregistrement.
6. Créez de la même manière un compte et plusieurs mouvements.

Modification rapide : verrouillage de l'application

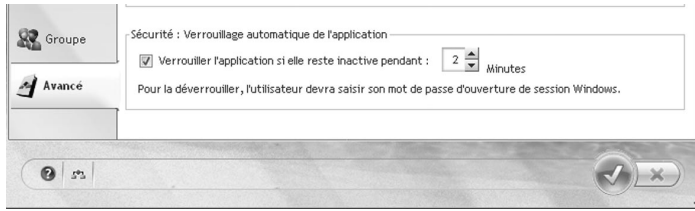
Bien souvent, lors de l'utilisation d'une application, celle-ci reste en cours sur le poste de l'utilisateur alors que celui-ci n'est pas présent (réunion, repas, ...). Pour éviter l'utilisation de l'application par des personnes non autorisées, il est possible de verrouiller automatiquement l'application. Pour se reconnecter, l'utilisateur devra utiliser son identifiant et son mot de passe Windows.

Pour utiliser cette fonctionnalité, quelques clics de souris suffisent.

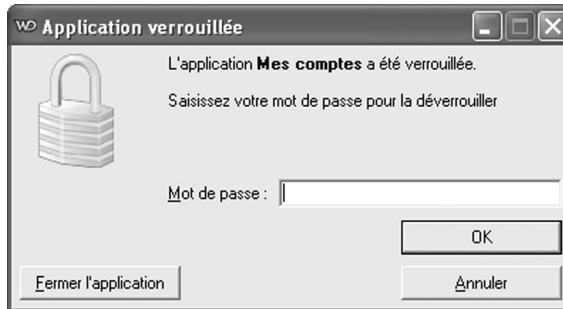
► Pour verrouiller une application :

1. Affichez la description du projet (option "Projet .. Description du projet").
2. Affichez l'onglet "Avancé".

3. Spécifiez les options de verrouillage (par exemple 2 minutes).



Si l'application n'est pas utilisée pendant 2 minutes, la fenêtre suivante sera affichée :



LEÇON 2.4. FENÊTRE DE SAISIE

Ce que vous allez apprendre dans cette leçon ...

- Création d'une fenêtre de saisie
- Gestion de l'ajout d'un enregistrement
- Gestion de la recherche puis de la modification

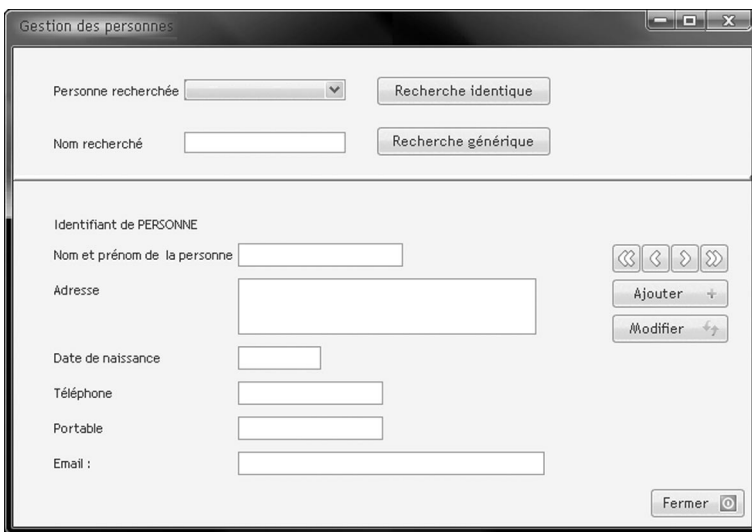


Durée estimée : 20mn

Présentation

Nous avons vu les différentes fenêtres créées automatiquement par le RAD. Maintenant, nous allons créer nous-même les fenêtres permettant de gérer l'ajout et la modification d'enregistrements. Ces manipulations vous feront découvrir plusieurs aspects de la gestion des fichiers de données et vous feront également utiliser certaines fonctionnalités bien utiles de WinDev.

La fenêtre que nous allons créer va permettre de gérer la saisie, la recherche et la modification des personnes. Cette fenêtre est la suivante :



Création d'une fenêtre de saisie

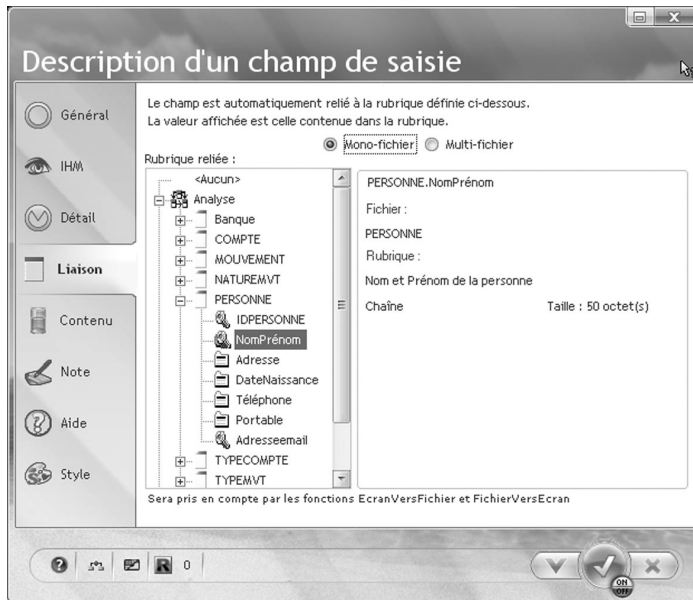
Pour créer la fenêtre de saisie sur le fichier Personne, nous pourrions utiliser l'assistant de création de fenêtre RAD et créer une fenêtre de type "Fiche".

Mais nous avons choisi de créer cette fenêtre à partir d'une fenêtre vierge : nous verrons ainsi toutes les étapes nécessaires à la programmation d'une telle fenêtre.

► Pour créer une fenêtre de saisie :


1. Créez une nouvelle fenêtre vierge (option "Fichier .. Nouveau .. Fenêtre", type de fenêtre "Vierge").
2. Dans la fenêtre de description, indiquez :
 - le nom de la fenêtre "FEN_PersonneX".
 - le titre et la description de la fenêtre "Gestion des personnes".
3. Validez la fenêtre de description.
4. Affichez le volet "Analyse" (option "Affichage .. Barre d'outils .. Analyse"). Les différents fichiers décrits dans l'analyse apparaissent dans le volet.
5. Double-cliquez sur le fichier "PERSONNE" : les rubriques du fichier sont listées.
6. Sélectionnez à l'aide de la souris l'ensemble des rubriques et effectuez un "Drag and Drop" (glisser / déplacer) de ces rubriques vers la fenêtre que vous venez de créer.
7. Différents champs sont automatiquement créés dans la fenêtre. Ces champs sont automati-

quement liés à la rubrique dans le fichier. Pour le vérifier, il suffit d'afficher l'onglet "Liaison" de la description d'un des champs (option "Description" du menu contextuel).



Fermez la description.

8. Enregistrez la fenêtre.

- ▶ Testez la fenêtre (icone .
- ▶ Nous allons modifier cette fenêtre pour gérer le cas des personnes âgées de moins de 18 ans. En effet, si une personne de moins de 18 ans est associée à un compte, des conditions spécifiques doivent être prises en compte.
 1. Créez un champ libellé et positionnez-le à côté du champ "Date naissance". Ce libellé a pour nom "LIB_Libellé1".
 2. Dans le code de sortie du champ "Date naissance", saisissez le code suivant :

```
dAge est une Date = Age(SAI_DateNaissance)
SI dAge..Année < 18 ALORS
    //On indique que le personne n'est pas majeure
    LIB_Libellé1 = "Personne non majeure" + RC + ...
                "Conditions spécifiques"
    LIB_Libellé1..Couleur = RougeFoncé
SINON
    //On réinitialise le libellé
    LIB_Libellé1 = ""
FIN
```

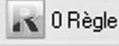
Ce code permet de calculer l'âge de la personne, et si celui-ci est inférieur à 18, le libellé affiche en rouge la chaîne "Personne non majeure".

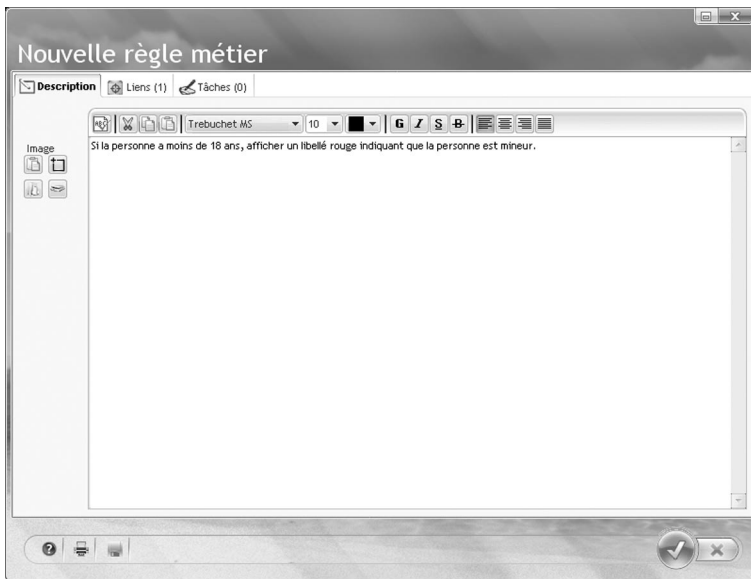
- Nous allons maintenant créer une règle métier sur cette fenêtre.

**Note**

Une règle métier permet de définir un mode opératoire précis ou de préciser un traitement particulier. Par exemple : le calcul d'un taux de TVA avec ses particularités, les règles de changement de statut d'un client, la formule de calcul d'un montant de frais de port, ...

Dans notre cas, la règle métier va s'appliquer à toutes les personnes ayant moins de 18 ans. Dans ce cas, un traitement spécifique doit être fait : il faut afficher un libellé indiquant que la personne est non majeure.

1. Affichez le volet des règles métier (par exemple en cliquant sur le bouton  0 Règle).
2. Par défaut, la nouvelle règle métier sera créée sur l'élément sélectionné : sélectionnez le libellé "LIB_Libellé1".
3. Cliquez sur le bouton "+": l'écran de saisie de la règle métier s'affiche.
4. La description de la règle est la suivante : "Si la personne a moins de 18 ans, afficher un libellé rouge indiquant que la personne est mineure".



5. Nous allons maintenant définir sur quels éléments la règle métier doit s'appliquer. Cliquez sur l'onglet "Liens".
6. Nous allons définir un lien automatique : la règle s'appliquera dès qu'un élément respectant les conditions de la règle sera créé. Cliquez sur le bouton "Conditions des liens automatiques".
7. Cette règle métier s'appliquera sur le projet courant. Sélectionnez l'option "Le projet courant uniquement".
8. Nous allons définir les conditions de sélection de la règle métier. Cliquez sur "+" et indiquez

les informations suivantes :



La règle va concerner tous les champs des fenêtres liés à la rubrique "Personne.DateNaissance". Validez. La condition de sélection est affichée.

Automatiquement, les fenêtres existantes du projet correspondant à la règle sont listées.

9. Validez.

10. Les liens de la règle métier sont mis à jour.

11. Validez la création de la règle métier.

12. Vous pouvez passer en "réalisé" la règle métier pour la fenêtre "FEN_Personne_X".

13. Enregistrez votre fenêtre.

Gestion de l'ajout d'un enregistrement

Nous allons maintenant gérer l'ajout d'un enregistrement dans notre fenêtre. En effet, la fenêtre actuellement contient des champs reliés au fichier de données, mais nous voulons maintenant pouvoir saisir des informations dans ces champs et les enregistrer dans le fichier de données "Personne".

Nous allons ajouter deux boutons dans notre fenêtre : un bouton "Ajouter" qui va permettre d'enregistrer les données et un bouton "Fermer" pour sortir de la fenêtre sans enregistrer.

Création du bouton d'ajout

► Pour créer le bouton d'ajout :

1. Créez un nouveau bouton (icone **OK**).

2. Ce bouton a pour libellé "Ajouter" et pour nom "BTN_Ajouter".

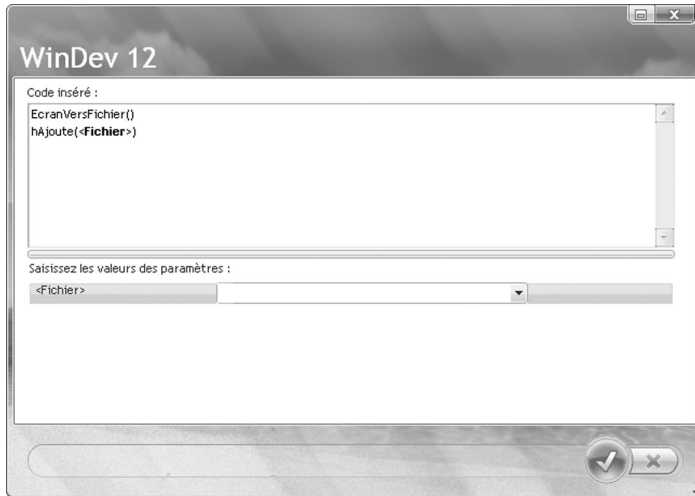
► Pour saisir le code du bouton, nous allons utiliser les briques de code :

1. Affichez le gestionnaire de briques de code (option "Affichage .. Barre d'outils .. Modules additionnels .. Gestionnaire de briques de code").

2. Afficher le code du bouton "BTN_Ajouter" (par exemple, sélectionnez le bouton et utilisez la touche F2).

3. Dans le gestionnaire de briques de code, sélectionnez la ligne "Ajout d'un enregistrement" et

faites glisser la souris vers le code de clic du bouton. La fenêtre suivante s'affiche :



Nous travaillons sur le fichier PERSONNE, sélectionnez le fichier dans la combo.

4. Validez. Le code suivant est automatiquement inséré :

```
EcranVersFichier ()
HAjoute (PERSONNE)
```


Détaillons ce code :


- La fonction **EcranVersFichier** permet d'initialiser les rubriques avec les valeurs des champs liés. Cette fonction est donc équivalente aux lignes suivantes :

```
Personne.NomPrenom = SAI_NomPrénom
Personne.Adresse = SAI_Adresse
Personne.DateNaissance = SAI_DateNaissance
...
```

Notre fenêtre utilise moins de 10 champs et déjà l'avantage se fait sentir; pensez simplement aux fenêtres qui contiennent plusieurs dizaines de champs : 1 seule ligne de code effectue toutes les affectations !

- La fonction **HAjoute** ajoute l'enregistrement dans le fichier. Cette fonction prend les valeurs en mémoire et écrit le contenu des rubriques du fichier dans le fichier lui-même. Les index sont immédiatement et automatiquement mis à jour.

5. Enregistrez votre fenêtre (icone .

► Testez la fenêtre (icone ). Remplissez les champs et cliquez sur le bouton "Ajouter".

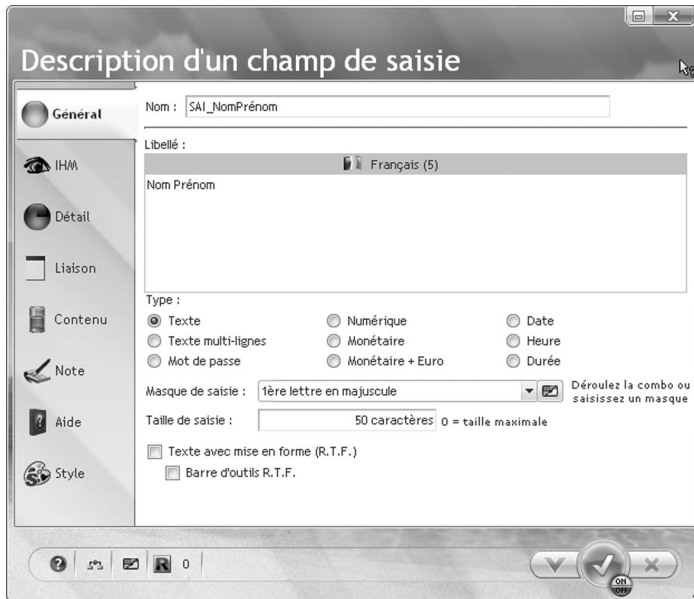
Plusieurs remarques :

- Aucun masque n'est utilisé. Vous pouvez saisir des majuscules et des minuscules dans les différents champs.
- Après l'ajout, les champs ne sont pas remis à vide.
- Aucun bouton de fermeture : il faut utiliser la croix en haut de la fenêtre.

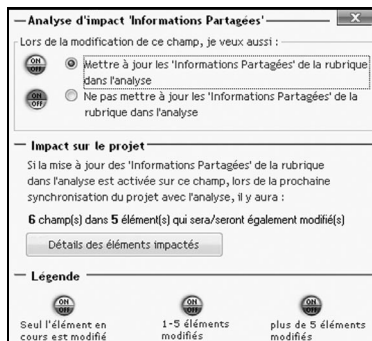
Nous allons faire les modifications nécessaires.

Ajout d'un masque de saisie

- ▶ Tout d'abord, le masque de saisie. Nous allons mettre un masque de saisie sur le champ "Nom Prénom". Nous allons sélectionner le masque "1ère lettre en majuscule" dans la description du champ.



Après avoir modifié le masque, le bouton Valider de la fenêtre de description change d'aspect. Un bouton "On / Off" apparaît. Si vous cliquez sur ce bouton, l'écran suivant apparaît :



Cet écran permet de gérer la modification des informations partagées dans l'analyse (les informations concernant le champ associé à la rubrique NOMPrenom). Si les informations partagées sont modifiées dans l'analyse, à la prochaine génération de l'analyse, tous les champs liés à la rubrique auront cette information également modifiée.

Validez la fenêtre de description du champ.

- ▶ Ajoutez de la même façon un masque de saisie sur le champ "SAI_DateNaissance". Ce masque est de la forme "JJ/MM/AAAA". La valeur retournée est de la forme "AAAAMMJJ".

Effacement des données après l'ajout

- Nous allons maintenant gérer l'effacement des données dans les champs, après l'utilisation du bouton "Ajouter". Seules deux nouvelles fonctions du WLangage vont nous permettre de réaliser cette opération sur tous les champs de la fenêtre.

1. Affichez le code du bouton "Ajouter" (F2 sur le bouton par exemple).
2. Complétez le code de la façon suivante :


```
EcranVersFichier ()
HAjoute (PERSONNE)
RAZ ()
HRAZ ()
```


La fonction **Raz** remet à blanc tous les champs pour la prochaine saisie. Cela permet ainsi de signaler à l'utilisateur que l'enregistrement a été ajouté.

La fonction **HRaz** remet à blanc toutes les variables de fichier (évite de conserver le buffer précédent en cas d'ajout partiel de rubriques dans un enregistrement).

3. Enregistrez la fenêtre.

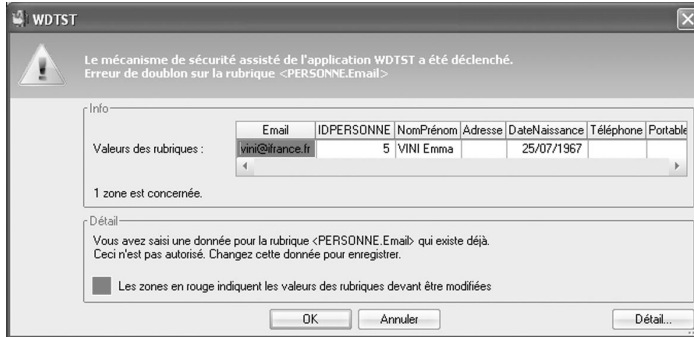
- Pour ajouter un bouton de fermeture :

1. Créez un bouton (icone ).
2. Saisissez les informations suivantes dans la fenêtre de description du bouton :
 - Libellé : "Ferme"
 - Nom : "BTN_Ferme".
 - Action : "Fermer la fenêtre".
3. Enregistrez la fenêtre.

- Testez la fenêtre (bouton ). Remplissez les champs et cliquez sur le bouton "Ajouter". Vous pouvez désormais saisir plusieurs enregistrements à la suite. Quelques exemples :

Nom Prénom	Date de naissance	Email
MOULAIN Françoise	03/11/1945	moulain@free.fr
TESTE Florence	19/07/1970	test@orange.fr
VINI Hervé	01/12/1965	vini@ifrance.fr
VINI Emma	25/07/1967	vini@ifrance.fr

Lors de la validation du dernier enregistrement, une fenêtre spécifique apparaît :



Cette fenêtre signale à l'utilisateur qu'il existe un doublon : en effet, l'adresse email (qui est une clé unique) est identique pour deux personnes. Cette fenêtre permet de modifier la valeur de l'adresse email : saisissez par exemple "vini2@ifrance.fr".

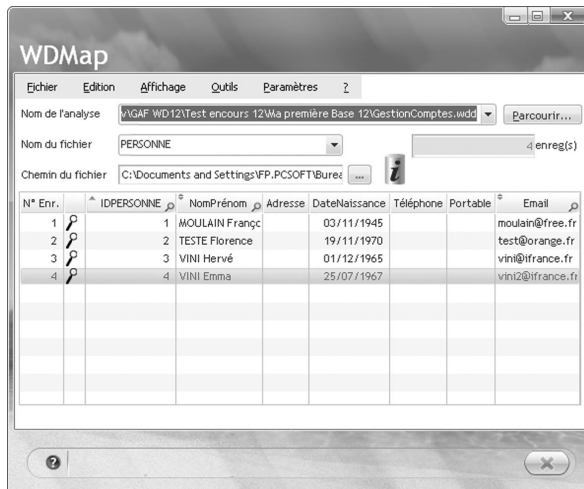
Cette fenêtre est une des fenêtres de gestion automatique des erreurs Hyper File, disponible en standard. Nous verrons dans la suite de ce cours comment gérer par programmation les différents types d'erreur pouvant survenir sur les bases de données.

Visualiser les enregistrements saisis

Depuis le début de l'utilisation de notre fenêtre, nous avons ajouté plusieurs enregistrements. Comment voir leur contenu ? WinDev propose un outil simple permettant de voir le contenu des fichiers pendant le développement de l'application (lorsque par exemple, les fenêtres de visualisation n'ont pas encore été créées). Cet outil s'appelle WDMAP. Nous allons l'utiliser pour voir le contenu du fichier Personne.

► Pour lancer WDMAP :

1. Sélectionnez l'option "Outils .. WDMAP - Visionneur de données".
2. Sélectionnez le fichier "Personne". Le contenu du fichier s'affiche.



Gestion de la recherche puis de la modification

Nous avons géré l'ajout d'enregistrements dans le fichier Personne. C'est bien. Mais ce serait encore mieux de pouvoir rechercher et modifier un enregistrement. C'est ce que nous allons faire maintenant, sur la même fenêtre.

Cette fenêtre va nous permettre de réaliser une recherche sur le nom. Nous allons mettre en place deux modes de recherche : une recherche à l'identique et une recherche générique.


Ensuite, il sera possible de modifier l'enregistrement affiché.

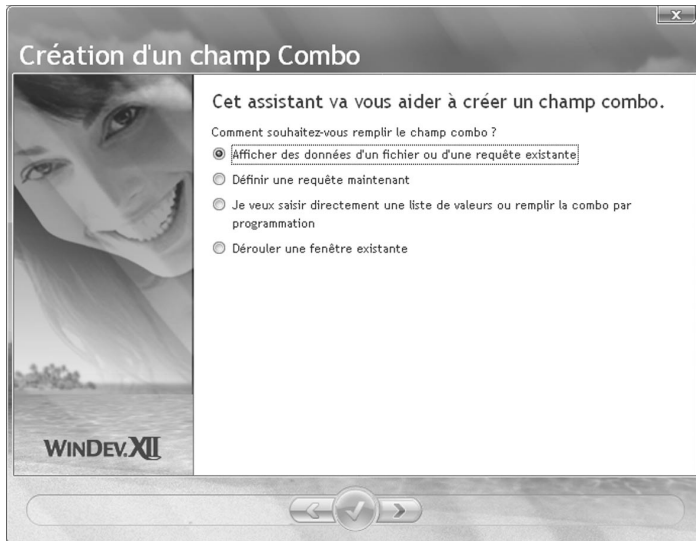
Recherche à l'identique

Pour effectuer la recherche à l'identique, nous allons sélectionner le nom et le prénom de l'utilisateur dans une combo. Le bouton "Rechercher" permettra d'afficher la fiche de la personne correspondante. Une seule personne correspond au nom et prénom sélectionnés.

Dans ce premier cas, le champ de recherche est une combo.

► Pour créer le champ de recherche :

1. Créez un champ de type combo (icone ).
2. L'assistant de création des combos s'affiche. Nous allons créer une combo basée sur le fichier "PERSONNE". Sélectionnez l'option "Afficher les données d'un fichier ou d'une requête existante".



Passez à l'écran suivant de l'assistant.

3. Sélectionnez le fichier "Personne". Passez à l'écran suivant.
4. Nous allons afficher le nom et le prénom de la personne. Sélectionnez la rubrique "NomPrénom". Passez à l'écran suivant.
5. La rubrique "NomPrénom" sera également utilisée pour trier la liste. Sélectionnez la rubrique "NomPrénom". Passez à l'écran suivant.
6. La rubrique retournée par la combo sera l'identifiant "IDPersonne". C'est cette valeur qui

sera recherchée dans le fichier *Personne*. Passez à l'écran suivant.

7. La combo ne sera liée à aucune rubrique. Passez à l'écran suivant.

8. Validez l'écran suivant avec la flèche jaune.

9. Donnez un nom à la combo, par exemple "Combo_Personne", et comme libellé "Personne recherchée".

10. Validez. Positionnez la combo dans la fenêtre (par exemple en haut à gauche).

► Pour créer le bouton de recherche :

1. Créez un bouton (icone ).

2. Positionnez le bouton à côté de la combo qui vient d'être créée.

3. Ce champ a pour nom "BTN_Identique", et pour libellé "Recherche identique".

4. Saisissez le code suivant :

```
HLitRecherchePremier (PERSONNE, IDPersonne, Combo_Personne)
SI HTrouve (PERSONNE) ALORS
    FichierVersEcran ()
FIN
```

La fonction **HLitRecherchePremier** permet de réaliser une recherche à l'identique. Dans cet exemple, la recherche se fait sur le fichier *PERSONNE* et sur la rubrique *IDPersonne*. La valeur recherchée correspond au dernier paramètre de la fonction. Ici, la valeur recherchée correspond à la valeur sélectionnée dans la combo. Cette valeur est obtenue en utilisant directement le nom de la combo (*Combo_Personne*).

La fonction **HTrouve** permet de tester le résultat de la recherche. Si **HTrouve** vaut Vrai, une valeur a été trouvée, si **HTrouve** vaut Faux, aucune valeur n'a été trouvée. Lorsqu'un enregistrement a été trouvé, cet enregistrement est lu : il devient l'enregistrement courant du fichier.

Dans ce code, si l'enregistrement a été trouvé, il est affiché grâce à la fonction **FichierVersEcran**.



Note

La fonction **FichierVersEcran** effectue l'opération inverse de **EcranVersFichier** : les données présentes dans les rubriques du fichier sont affichées dans les champs correspondants.


5. Enregistrez la fenêtre.


► Testez la fenêtre. Saisissez une valeur dans la combo et cliquez sur le bouton de recherche. Le résultat est immédiat.

Recherche générique

Nous allons maintenant effectuer une recherche générique. Au lieu de rechercher exactement la valeur saisie, nous allons rechercher tous les éléments qui commencent par la valeur saisie.

Pour effectuer cette recherche, nous allons créer un champ de saisie afin de saisir le nom recherché et un bouton pour lancer la recherche.

- ▶ Pour créer le champ de recherche :
 1. Créez un champ de saisie (icone ).
 2. Ce champ a pour nom "SAI_Nom_recherché" et pour libellé "Nom recherché" .

- ▶ Pour créer le bouton de recherche :
 1. Créez un bouton (icone ).
 2. Positionnez le bouton à côté du champ de saisie qui vient d'être créé.
 3. Ce champ a pour nom "BTN_Generique", et pour libellé "Recherche générique".
 4. Saisissez le code suivant :

```
HLitRecherche (PERSONNE, NomPrénom, SAI_Nom_Recherché)
SI HTrouve (PERSONNE) ALORS
    RAZ ()
    FichierVersEcran ()
SINON
    Erreur ("Aucune personne ne correspond")
FIN
```

La fonction **HLitRecherche** permet de réaliser une recherche générique. Dans cet exemple, la recherche se fait sur le fichier PERSONNE et sur la rubrique "NomPrénom". La valeur recherchée correspond à la valeur saisie dans le champ SAI_Nom_Recherché. Cette valeur est obtenue en utilisant directement le nom du champ.




Note

Il est possible de faire une recherche à l'identique avec la fonction **HLitRecherche** : il suffit d'utiliser la constante `hidentique`.

5. Enregistrez la fenêtre.
- ▶ Testez la fenêtre. Saisissez une valeur dans le champ de saisie et cliquez sur le bouton de recherche. Le résultat est immédiat. Cependant, si plusieurs enregistrements correspondent, seul le premier est affiché.

Modification de la fiche affichée

Lorsque le résultat de la recherche est affiché, il peut être intéressant de modifier les informations affichées. Modifier les valeurs des champs de saisie est simple, mais il faut ensuite les prendre en compte dans le fichier de données. Pour cela, nous allons créer un bouton de modification.

- ▶ Pour créer le bouton de modification :
 1. Créez un bouton (icone ).
 2. Positionnez le bouton sous le bouton "Ajouter".
 3. Ce champ a pour nom "BTN_Modifier", et pour libellé "Modifier".
 4. Saisissez le code suivant :

```
EcranVersFichier ()
HModifie (PERSONNE)
ListeAffiche (COMBO_PERSONNE, taCourantBandeau)
```

Dans ce code, la fonction **HModifie** permet de modifier l'enregistrement en cours avec les données présentes dans l'écran.

La fonction **ListeAffiche** permet de mettre à jour la combo de recherche (par exemple en cas de modification du nom).



Note

Lors de la modification d'un enregistrement, des erreurs d'intégrité, de doublons, .. peuvent survenir. Par défaut, le mécanisme de la gestion automatique des erreurs est activé (comme nous l'avons déjà vu lors de l'ajout).

5. Enregistrez et testez votre fenêtre.

Parcours des fiches

Nous allons maintenant ajouter des boutons pour parcourir les différents enregistrements.

- Créez quatre boutons qui ont pour nom par exemple "BTN_PREMIER", "BTN_PRECEDENT", "BTN_SUIVANT", et "BTN_DERNIER".

Le code de ces boutons sera :

```
//Bouton btn_premier : appeler le premier
HLitPremier(Personne, IDPersonne)
SI HEnDehors(Personne) = Vrai ALORS
    Info("Aucune fiche à visualiser")
SINON
    FichierVersEcran()
FIN
```

```
//Bouton btn_précédent : appeler le précédent
HLitPrécédent(Personne, IDPersonne)
SI HEnDehors(Personne) = Vrai ALORS
    Info("Début du fichier atteint")
SINON
    FichierVersEcran()
FIN
```

```
//bouton btn_Suivant : appeler le suivant
HLitSuivant(Personne, IDPersonne)
SI HEnDehors(Personne) = Vrai ALORS
    Info("Fin du fichier atteinte")
SINON
    FichierVersEcran()
FIN
```

```
//Bouton btn_dernier : appeler le dernier
HLitDernier(Personne, IDPersonne)
SI HEnDehors(Personne) = Vrai ALORS
    Info("Aucune fiche à visualiser")
SINON
    FichierVersEcran()
FIN
```

La fonction **HLitPremier** permet de lire le premier enregistrement du fichier de données, selon la clé spécifiée. Ici, nous lisons le fichier PERSONNE selon la clé IDPersonne.

La fonction **HLitDernier** est basée sur le même principe, cette fonction va lire l'enregistrement qui a la valeur de clé la plus grande.

La fonction **HLitSuivant** va lire l'enregistrement qui a la valeur de clé immédiatement supérieure à celle de l'enregistrement en cours.

La fonction **HLitPrécédent** va lire l'enregistrement qui a la valeur de clé immédiatement inférieure à celle de l'enregistrement en cours.

Dans tous les cas :

- la fonction **HEnDehors** permet de savoir si le fichier est vide.
- la fonction **FichierVersEcran** permet d'afficher l'enregistrement à l'écran.

► Enregistrez la fenêtre et testez-la. Parcourez le fichier en cliquant sur chacun des boutons.

LEÇON 2.5. TABLE SUR UNE REQUÊTE

Ce que vous allez apprendre dans cette leçon ...

- Création d'une requête paramétrée
- Création d'une fenêtre avec une table basée sur la requête
- Création d'un état automatique sur la table



Durée estimée : 20mn

Présentation

Dans cette leçon, nous allons apprendre à manipuler les tables. Dans notre exemple, nous allons créer une fenêtre contenant une table listant les mouvements effectués dans une période donnée. Cette table est basée sur une requête que nous allons créer. La requête permet de sélectionner les enregistrements affichés dans la table.

La fenêtre est la suivante :

Numéro de compte	Date du mouvement	Mouvement	Nature du mouvement	Montant
0123456L030	01/10/2007	Loyer octobre	Loyer	737,28 €
0123456L030	02/10/2007	Salaire septembre	Divers	2 350,52 €
0123456L030	10/10/2007	Essence	Carburant	26,00 €

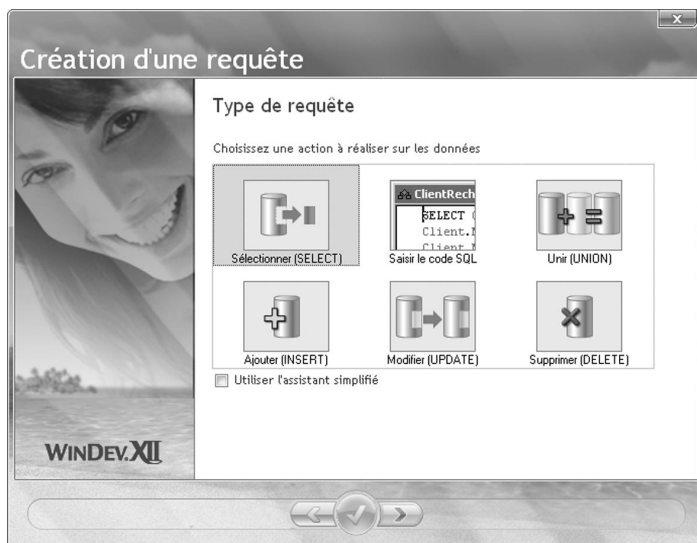


Note

Cette leçon "survole" la création de requête. Une partie entière de ce cours d'auto-formation est consacrée à la gestion des requêtes ("Créez une requête de sélection", page 305).

Création d'une requête paramétrée

- Pour créer une requête, nous allons utiliser l'éditeur de requêtes.
 1. Sélectionnez l'option "Fichier .. Nouveau .. Requête". L'assistant de création de requête se lance.



2. Sélectionnez l'option "Sélectionner (SELECT)". En effet, la requête que nous allons créer va nous permettre de sélectionner des enregistrements. Passez à l'écran suivant.

3. La fenêtre de description de la requête apparaît. Nous allons construire la requête en sélectionnant les éléments que nous voulons dans le résultat.

4. Double-cliquez sur les rubriques présentes dans l'analyse à gauche de la fenêtre de description. Les rubriques prises en compte apparaissent alors au centre de l'écran.

Nous voulons afficher :

- le numéro de compte (rubrique NumCompte du fichier COMPTE),
- la date du mouvement qui a été réalisé (rubrique DateMvt du fichier MOUVEMENT),
- son intitulé (rubrique Intitulé du fichier MOUVEMENT),
- la nature du mouvement (rubrique Intitulé du fichier NATUREMVT)
- son montant (rubrique Montant du fichier MOUVEMENT).

La fenêtre de description de la requête est la suivante :




5. Nous allons trier les données par date. Sélectionnez la rubrique "MOUVEMENT.DateMvt", puis cliquez sur le bouton "Trier" et sélectionnez l'option "Trier sur la rubrique sélectionnée". Indiquez un tri croissant sur la rubrique et validez.

6. Nous allons maintenant indiquer que la date correspond à une période saisie par l'utilisateur. Pour cela, sélectionnez la rubrique "MOUVEMENT.DateMvt", puis cliquez sur le bouton "Entre deux dates". Les deux dates seront fournies par l'utilisateur : indiquez que la valeur est comprise entre deux paramètres et validez.




7. Notre requête est terminée. Donnez un nom à la requête (par exemple REQ_Mouvements, en haut à gauche de l'écran). Validez la fenêtre de description de la requête.

8. Nous allons tester la requête : cliquez sur . La fenêtre qui s'affiche demande les paramètres de la requête. Saisissez les dates correspondant aux mouvements saisis lors du test. Le résultat s'affiche automatiquement.

Nous allons maintenant créer la fenêtre contenant la table associée à cette requête.

Création d'une fenêtre avec une table basée sur la requête

La fenêtre que nous allons créer va permettre à l'utilisateur de saisir l'intervalle de dates pour lequel la liste des mouvements effectués doit être affichée.

- ▶ Pour créer la fenêtre avec le champ table :
 1. Créez une nouvelle fenêtre (option "Fichier .. Nouveau .. Fenêtre"). Sélectionnez une fenêtre "vierge" et validez.
 2. Donnez un nom et un titre à votre fenêtre : par exemple : FEN_Mouvements et "Mouvements pour une période donnée".
 3. Créez un champ table (icone ). L'assistant de création de table se lance.
 4. Nous allons créer une table basée sur la requête que nous venons de créer. Sélectionnez l'option "Afficher les données d'un fichier ou d'une requête existante". Passez à l'écran suivant de l'assistant.
 5. Sélectionnez la requête qui sera la source de données de la table : REQ_Mouvements. Passez à l'écran suivant de l'assistant.
 6. Sélectionnez toutes les rubriques proposées. Passez à l'écran suivant de l'assistant.
 7. Sélectionnez la rubrique de parcours : "DateMvt". Passez à l'écran suivant de l'assistant.
 8. Saisissez les différents paramètres de la table : gardez les options proposées par défaut et décochez l'option "Générer le code d'initialisation de la requête".



Attention!

L'option "Générer le code d'initialisation de la requête" permet de générer le code d'initialisation du champ Table. Cette option est à utiliser avec précaution car le code généré utilise les valeurs par défaut données lors du test de la requête.

Passez à l'écran suivant de l'assistant.

9. Donnez un nom et un titre à la table (par exemple "TABLE_Mouvements" et "Mouvements réalisés pour la période donnée". Validez.
10. La table est automatiquement créée. Vous pouvez la redimensionner ou redimensionner la fenêtre.
11. Enregistrez la fenêtre.

- ▶ Pour créer les champs permettant de saisir l'intervalle de dates, nous allons utiliser un super-champ fourni par défaut avec WinDev.
 1. Affichez si nécessaire le volet "Assistants, exemples et composants" (option "Affichage ..

Barre d'outils .. Assistants, exemples et composants").

2. Sélectionnez "Champs".

3. Sélectionnez "Superchamp".

4. Sélectionnez le superchamp "Saisie d'une période" et réalisez un "Glisser / déplacer" vers la fenêtre : positionnez le superchamp par exemple au-dessus de la table.

► Nous allons maintenant saisir le code qui va permettre d'initialiser la table. Ce code va être saisi dans un bouton spécifique.

1. Créez un nouveau bouton. Ce bouton a pour nom "BTN_Mouvement" et pour libellé "Mouvements de la période".

2. Affichez le code du bouton (par exemple avec l'option "Code" du menu contextuel du bouton).

3. Saisissez le code suivant dans le code de clic du bouton :

```
REQ_Mouvements.Param1 = SC_SélectionPériode.SAI_DateDébut  
REQ_Mouvements.Param2 = SC_SélectionPériode.SAI_DateFin  
HExécuteRequête (REQ_Mouvements)  
TableAffiche (TABLE_Mouvements, taCourantPremier)
```

Ce code permet d'initialiser les paramètres de la requête REQ_Mouvements avec les valeurs saisies dans les champs Date.

La table est ensuite affichée.

► Il nous reste plus qu'à enregistrer et tester la fenêtre.

Création d'un état automatique sur la table

La fenêtre affiche correctement les mouvements effectués par compte entre deux dates. Pourquoi ne pas mettre ces informations dans un état pour les imprimer ? Mais comment faire l'état ?

Rien de plus simple avec WinDev ! Les données sont affichées dans une table ? Il suffit de créer un bouton de type "Bouton Etat sur table" et l'état correspondant à la table est créé automatiquement.

► Pour ajouter un bouton "Etat sur table" :

1. Sélectionnez l'option "Insertion .. Spécial .. Bouton Etat sur table".

2. L'état est immédiatement créé.

3. Une fenêtre s'affiche proposant d'ajouter l'état au projet. Validez.

4. Revenez sur la fenêtre "FEN_Mouvements" (par exemple en cliquant sur les boutons).

5. Le bouton "Imprimer" a été positionné en haut à gauche de la fenêtre. Déplacez-le par exemple sous la table.

6. Enregistrer la fenêtre et testez-la.

LEÇON 2.6. ETAT AVEC REQUÊTE INTÉGRÉE

Ce que vous allez apprendre dans cette leçon ...

- Création d'un état
- Modification d'un état
- Test d'un état



Durée estimée : 20mn

Présentation

Dans la leçon précédente, nous avons créé un état très rapidement. Dans cette leçon, nous allons nous arrêter sur la création d'un état basé sur une requête. Mais au lieu de faire d'abord la requête, puis l'état, nous allons créer la requête pendant la description de l'état. Dans ce cas, la requête sera intégrée à l'état : elle ne pourra pas être utilisée ailleurs dans le projet.

L'état que nous allons créer est relativement simple : nous voulons afficher pour chaque compte de chaque banque les opérations réalisées.

Opérations sur compte bancaire				12/10/2007
Organisme :	BNP PARIBAS			
Numéro de compte :	0123456L030			
Date du	Nature du mouvement	Sens	Montant	
10/10/2007	Carburant	Débit	-26 €	
01/10/2007	Loyer	Débit	-737 €	
21/09/2007	Retrait espèces	Débit	-30 €	
02/10/2007	Divers	Crédit	2 351 €	
			1 557 €	
Nombre de lignes : 4				Powered by WinDev



Note

Cette leçon "survole" la création d'un état. Une partie entière de ce cours d'auto-formation est consacrée à la gestion des états ("Votre premier état", page 329).

Création d'un état

- Pour créer un état :
 1. Sélectionnez l'option "Fichier .. Nouveau .. Etat". L'assistant de création d'un état se lance automatiquement.
 2. Sélectionnez un état de type "Tableau". Passez à l'écran suivant de l'assistant.
 3. L'état va être basé sur une requête : sélectionnez l'option "d'une requête". Passez à l'écran suivant de l'assistant.
 4. Sélectionnez l'option "D'une requête que je veux définir maintenant". Passez à l'écran suivant de l'assistant.
 5. L'écran de description de la requête apparaît. Nous allons définir les rubriques qui vont composer la requête :
 - La rubrique "Organisme" du fichier BANQUE : le nom de la banque
 - La rubrique "NumCompte" du fichier COMPTE : le numéro de compte
 - La rubrique "Intitulé" du fichier NATUREMVT : l'intitulé de l'opération
 - La rubrique "DateMvt" du fichier MOUVEMENT : la date de l'opération
 - La rubrique "Montant" du fichier MOUVEMENT : le montant de l'opération

- la rubrique "Sens" du fichier MOUVEMENT : le sens de l'opération

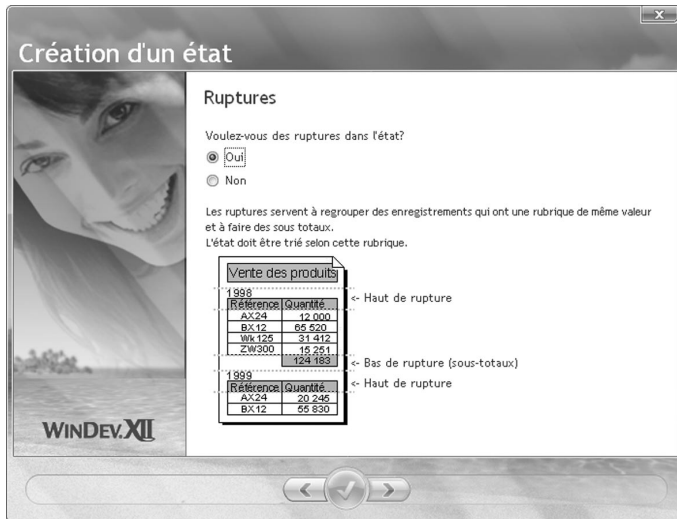


6. Nous allons trier les données :

- sélectionnez la rubrique Organisme et définissez un tri croissant (bouton "Trier").
- sélectionnez la rubrique NumCompte et définissez un tri croissant (bouton "Trier")

7. Validez la requête. L'assistant de création d'état continue.

8. L'état va contenir une rupture. Les ruptures servent à regrouper les données. Dans l'état, les données vont être regroupées par organisme bancaire. Passez à l'écran suivant.



9. L'assistant propose automatiquement la rubrique **Organisme** et la rubrique **NumCompte** comme rubriques de rupture. Ce sont les rubriques pour lesquelles nous avons défini un tri dans la requête. Sélectionnez uniquement la rubrique **Organisme** et passez à l'écran suivant.

10. Cet écran est très important. En effet, il permet d'associer les différentes rubriques associées à l'état aux différentes zones de l'état. Dans cet exemple, l'organisme et le numéro de compte seront affichés en haut de rupture.

Nous allons également changer l'ordre des rubriques : dans le "corps" de l'état (le bloc principal), le tableau affichera la date, l'intitulé, et le montant. Utilisez les flèches à côté du tableau pour réorganiser les rubriques.



Passez à l'écran suivant.

11. L'écran suivant nous propose de faire un total sur la rubrique Montant à chaque fin de rupture. Acceptez en passant à l'écran suivant.

12. Pour les dimensions du papier, choisissez l'affichage en mode paysage. Passez à l'écran suivant.

13. Choisissez un gabarit pour votre état (ActivVista par exemple).

14. Donnez un nom et un libellé à l'état (par exemple ETAT_Opération, et "Opérations par compte bancaire").


15. Validez.

16. L'état est automatiquement créé.

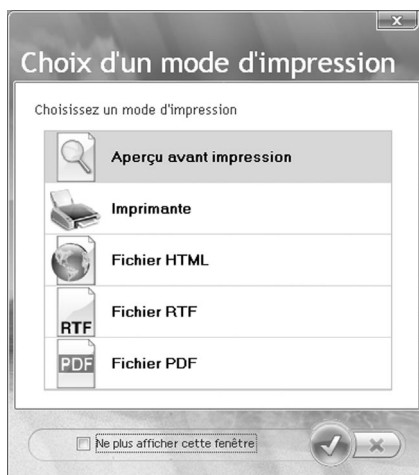
17. Enregistrez l'état.

Test d'un état

► Pour tester un état :

1. Lancez le test grâce à l'icône .

2. Choisissez le mode d'impression "Aperçu avant impression" :



3. L'état apparaît dans une fenêtre d'aperçu.

► Si vous regardez attentivement cet état, vous vous rendrez compte que cet état est faux ! La somme des montants calculés ne tient pas compte du sens du mouvement. Nous allons faire quelques modifications dans l'état pour y remédier.

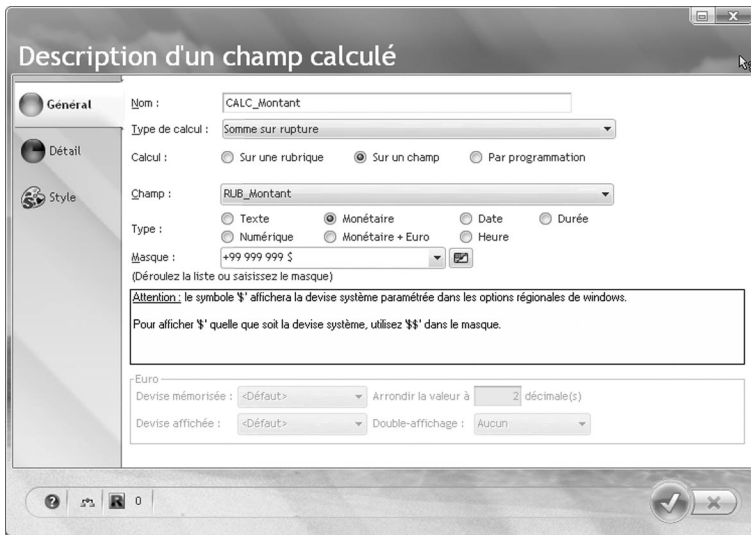
1. Editez le code du bloc CORPS (positionnez le curseur de la souris dans la partie CORPS de l'état et sélectionnez l'option "Code du bloc" du menu contextuel).

2. Dans le code du bloc, ajoutez le code suivant :

```
//Affichage des montants négatifs en cas de débit
SI RUB_Sens = 1 ALORS
    RUB_Montant = RUB_Montant*-1
FIN
```

Ce code permet de prendre en compte les montants négatifs.

3. Fermez l'éditeur de code.
4. Éditez la description du champ "CALC_Montant". Ce champ correspond au montant présent dans le bloc FIN DE RUPTURE. Ce champ affiche le montant total.
5. Modifiez la méthode de calcul du total : le total doit être fait sur le champ "RUB_Montant".



6. Validez.
7. Enregistrez l'état et testez-le.

Modification d'un état

Nous allons maintenant apporter une petite amélioration à l'état que nous venons de réaliser : chaque nouvel organisme bancaire va s'afficher sur une nouvelle page. L'organisme bancaire est la rubrique qui sert de tri et de rupture. Nous allons donc ajouter un saut de page après chaque rupture.

- Pour ajouter un saut de page après chaque rupture :
 1. Affichez l'état sous l'éditeur (si nécessaire).
 2. Cliquez dans la zone "Bas de rupture 1".
 3. Affichez le menu contextuel (clic droit) et sélectionnez l'option "Description du bloc".
 4. Sélectionnez l'onglet "Détail" de la fenêtre de description.
 5. Cochez l'option "Saut de page après le bloc".
 6. Validez.
 7. Enregistrez et testez l'état.

LEÇON 2.7. LE GROUPWARE UTILISATEUR

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce le groupware utilisateur ?
- Intégrer le groupware utilisateur
- Configurer le groupware utilisateur
- Tester le groupware utilisateur



Durée estimée : 20mn

Présentation

Une application peut être utilisée par différents intervenants, ayant tous des profils différents. Il est souvent nécessaire de prévoir des niveaux d'accès selon les fonctions de l'utilisateur.

Prenons un exemple simple : lors de la mise en place d'une gestion commerciale, l'application propose les fonctionnalités suivantes :

- Consultation du tarif
- Modification du tarif
- Saisie des commandes
- Saisie des clients.

Selon l'utilisateur, les accès possibles sont différents. Quelques exemples :

- les secrétaires peuvent consulter le tarif et créer des commandes
- les commerciaux peuvent consulter le tarif et établir les commandes, gérer de nouveaux clients.
- les directeurs commerciaux ont accès à toutes les options.

WinDev permet de gérer ces différents niveaux d'accès en quelques clics dans vos applications grâce au groupware utilisateur.

Nous allons intégrer le groupware utilisateur dans notre application "Mes comptes" et le configurer.

Intégrer le groupware utilisateur

- Pour intégrer le groupware utilisateur dans le projet "Mes comptes" :
 1. Sélectionnez l'option de menu "Atelier .. Paramétrer le groupware utilisateur". La fenêtre de paramétrage du groupware utilisateur s'affiche.



2. Cochez l'option "Activer la gestion du groupware utilisateur".

3. Conservez les options par défaut de l'onglet "Général" :

- Intégration par défaut : la fenêtre de login est automatiquement intégrée à l'application.
- Lancement automatique : le groupware utilisateur est lancé dès le démarrage de l'application.



Note

L'intégration personnalisée permet de personnaliser les fenêtres et le code du groupware utilisateur. Cette option est destinée aux développeurs voulant par exemple traduire les fenêtres du groupware utilisateur, ou les personnaliser.

Si le client final utilise un **annuaire LDAP**, il est possible de l'utiliser pour authentifier les utilisateurs.

Lors de l'installation de l'application, l'utilisateur pourra saisir les paramètres de son annuaire LDAP.

4. Sélectionnez l'onglet "Fichiers". Cet onglet permet de définir le format et l'emplacement des fichiers de données du groupware utilisateur. Dans notre cas, nous utiliserons des fichiers Hyper File Classic, dans l'emplacement spécifié par défaut.



Note

Si vous utilisez (ou si l'utilisateur final utilise) **Windows Vista**, il est conseillé d'utiliser le répertoire des données.

5. Sélectionnez l'option "Activer l'historique des connexions". Cette option permet au superviseur de savoir qui s'est connecté, quand et à quelle heure.

6. Validez. Le groupware utilisateur est intégré dans l'application.

► Nous allons tout de suite tester notre application :

1. Lancez le test du projet (). Une fenêtre de login apparaît.



Note

Par défaut, un seul utilisateur existe, le superviseur. Pour se connecter en tant que superviseur utilisez :

- le nom : SUPERVISEUR
- le mot de passe : SUPERVISEUR

2. Connectez-vous en tant que superviseur.

3. Un nouveau menu apparaît, permettant soit de tester l'application, soit de configurer l'application.

4. Choisissez l'option "Configurer le groupware". Nous allons configurer le groupware utilisateur.

Configurer le groupware utilisateur

La configuration du groupware utilisateur consiste à définir les différents utilisateurs de l'application et leurs droits sur les différentes fenêtres et champs de l'application.



Note

La configuration des utilisateurs peut être réalisée :

- lors du développement de l'application. Les fichiers nécessaires (des utilisateurs et de leurs droits) pourront être installés avec l'application.
- lorsque l'application est installée, par l'administrateur de l'application.

- Pour configurer le groupware utilisateur :



1. Créez les différents utilisateurs qui auront accès à l'application (bouton "Nouvel utilisateur"). Par défaut, seul le superviseur existe.

Créez le nouvel utilisateur TEST. Son mot de passe est "test".

2. Associez si nécessaire les utilisateurs à un groupe. Le bouton "Nouveau groupe" permet de créer des groupes.

3. Paramétrez les droits des utilisateurs grâce au bouton "Modifier les droits". La fenêtre qui s'affiche permet de sélectionner chaque fenêtre de l'application. Pour chaque élément de la fenêtre, il est possible de dire si le champ aura le comportement de l'application (défaut) ou sera inactif, invisible ou grisé.

- Dans la suite de ce cours nous allons voir les tests automatiques : il est nécessaire de configurer le groupware utilisateur afin de ne pas avoir de fenêtre de login.

1. Sélectionnez l'option "Atelier .. Paramétrer le groupware utilisateur".

2. Indiquez les informations suivantes dans la partie "Login automatique en mode test" :

- Login : TEST
- Mot de passe : test

LEÇON 2.8. LE TABLEAU DE BORD

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce que le tableau de bord ?
- Tests automatiques
- Optimisation de requêtes



Durée estimée : 20mn


Présentation

Le tableau de bord du projet est un élément indispensable à la gestion de projets WinDev. Le tableau de bord permet d'avoir une vision globale et synthétique de l'état d'avancement d'un projet.

Le tableau de bord est constitué de jauges, de voyants, de compteurs permettant de voir d'un seul coup d'œil l'état d'un projet.

Dans cette partie, nous nous limiterons à la gestion des tests automatiques et à l'optimisation des requêtes.

Le tableau de bord sera abordé plus en détail dans le chapitre "Tableau de bord", page 429.

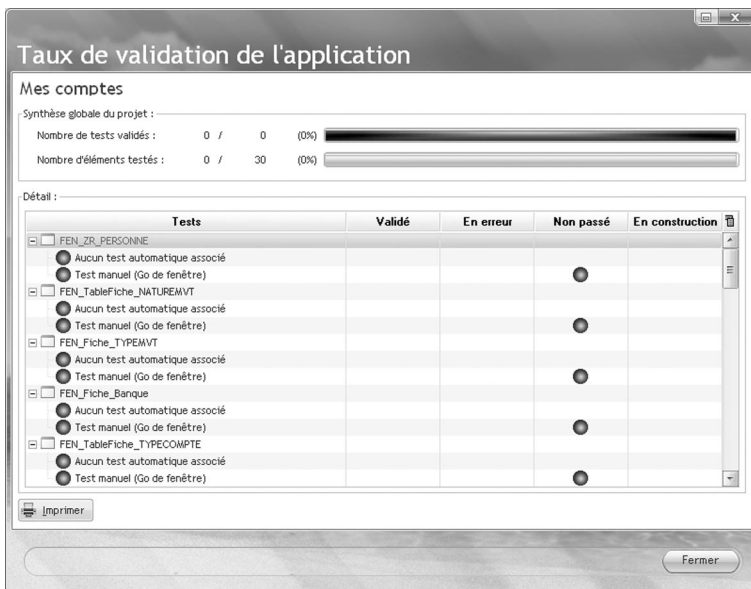
- Pour afficher le tableau de bord du projet (si ce n'est pas déjà fait), cliquez sur l'icone .

Tests automatiques

Une des fonctionnalités intéressante du tableau de bord est de renseigner sur les tests qui ont été réalisés sur l'application.

Nous avons déjà fait de nombreux tests d'utilisation, au fur et à mesure de la conception de notre application. Cependant le voyant Test est encore rouge !

- Cliquez sur le voyant test. La fenêtre de taux de validation de l'application apparaît.



Cette fenêtre recense tous les tests qui ont été effectués sur l'application :

- Tests manuels (c'est les tests que nous avons déjà effectués)
- Tests automatiques.

Les tests automatiques sont une catégorie de tests spécifiques. Les tests automatiques permettent d'automatiser certaines manipulations de vos fenêtres. Ces tests sont enregistrés sous forme

de scénarios WLangage, facilement modifiables sous l'éditeur de code. Une fois enregistré, le test automatique peut être réexécuté autant de fois que vous voulez, pour tester par exemple l'impact d'une modification sur une fenêtre, une procédure, ...

Faisons immédiatement un essai ! Nous allons créer un test automatique sur la fenêtre de saisie que nous avons créée au début de cette leçon.



Note

Les tests automatiques peuvent être réalisés sur des fenêtres, des procédures, des classes.

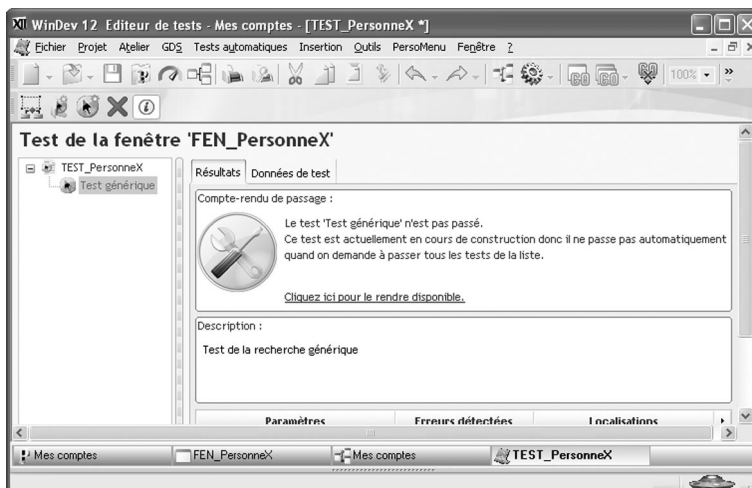
- ▶ Pour créer un test automatique sur une fenêtre :
 1. Ouvrez la fenêtre "FEN_PersonneX" créée précédemment.
 2. Sélectionnez l'option de menu "Tests automatiques .. Enregistrer un nouveau test".
 3. L'éditeur de tests automatiques apparaît, et vous propose de commencer l'enregistrement. Acceptez.
 4. La fenêtre "FEN_PersonneX" se lance.
 5. Saisissez un nom dans la zone "Recherche générique" et cliquez sur le bouton "Recherche générique".
 6. Cliquez sur le bouton "Fermer".
 7. L'éditeur de tests propose d'enregistrer une description pour le test. Acceptez. Le test s'appelle "Test générique" et sa description correspond à "Test de la recherche générique". Validez.
 8. Le code du test en WLangage apparaît sous l'éditeur de code. Fermez cette fenêtre de code.



Note

Les fonctions WLangage permettant de réaliser des tests unitaires sont les fonctions PiloteXXX.

- 9. L'éditeur de tests apparaît :



Le test est actuellement en cours de construction.

10. Nous allons rendre le test disponible (cliquez sur le lien "Cliquez ici pour le rendre disponible"). Maintenant, le test pourra être lancé à tout moment. L'éditeur de tests nous indique que le test n'est jamais passé. Cliquez sur le lien "Cliquez ici pour lancer le test".

11. Le test s'exécute automatiquement et l'éditeur de tests affiche le résultat (le test est passé avec succès dans notre cas).

12. Fermez l'éditeur de tests et enregistrez le test si nécessaire.

Nous allons maintenant faire une modification dans notre fenêtre, puis repasser le test. La modification porte sur le champ de recherche. Nous allons ajouter un code de vérification dans le code "A chaque modification".

1. Ajoutez le code suivant dans le code "A chaque modification" du champ

SAI_Nom_Recherché :

```
SI Taille(SAI_Nom_Recherché) >=2 ALORS  
    BTN_GENERIQUE..Etat = Grisé  
SINON  
    BTN_GENERIQUE..Etat = Actif  
FIN
```

2. Enregistrez la fenêtre.

3. Passez le test associé à la fenêtre en utilisant l'option "Tests automatiques .. Lancer les tests".

4. Le test est désormais en erreur.

5. Modifiez le code du champ de recherche de la façon suivante :

```
SI Taille(SAI_Nom_Recherché) <=2 ALORS
```

6. Relancez le test. Le test passe maintenant correctement.

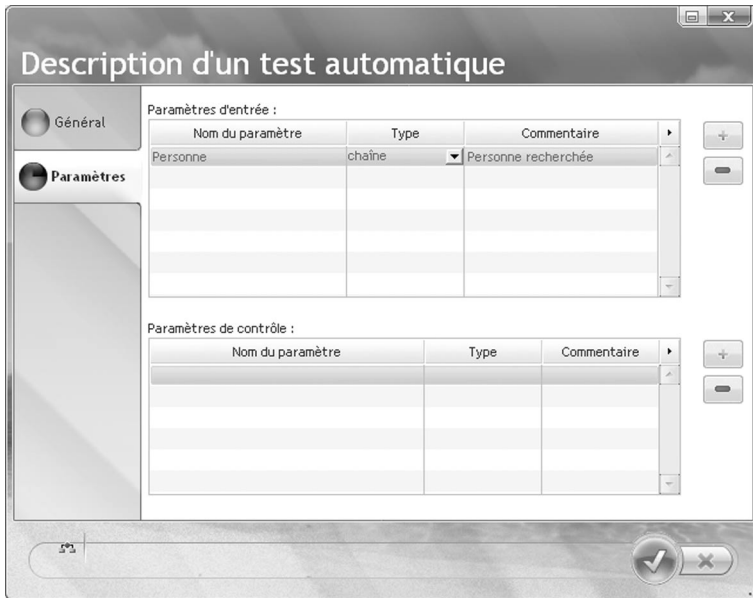
Une autre possibilité de l'éditeur de tests est la possibilité d'utiliser un jeu de tests : c'est le concept des itérations. En effet, dans notre exemple, nous avons testé uniquement la recherche sur une personne. Nous allons modifier notre test pour rechercher les différentes personnes de notre fichier. Ce test n'est pas un cas réel de test, mais il permet de tester la fonctionnalité des itérations.

1. Affichez l'éditeur de tests (par exemple double-cliquez sur "TEST_PersonneX" dans le volet WinDev).

2. Affichez la description du test (Option "Description" du menu contextuel du "Test générique").

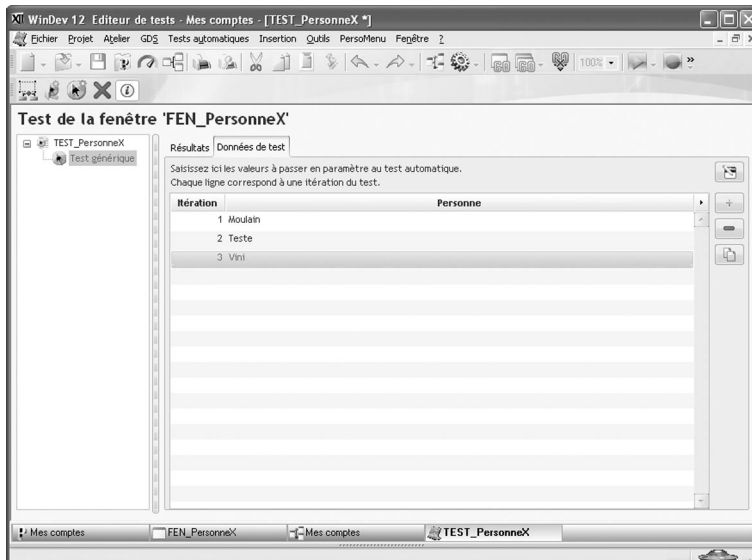
3. L'onglet "Paramètres" permet de saisir les paramètres à prendre en compte dans le test. Ici,

nous aurons uniquement en paramètre d'entrée, le nom recherché :



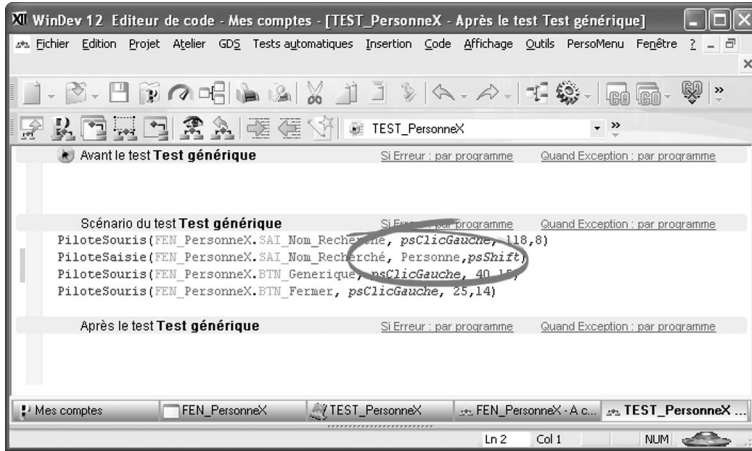
Validez la fenêtre de description.

4. Dans l'onglet "Données de test", nous allons saisir les noms des personnes présentes dans notre fichier, par exemple :



5. Nous allons maintenant éditer le code du test pour prendre en compte le paramètre. Sélectionnez l'option "Code" du menu contextuel du test "Test Générique". Il est nécessaire de rem-

placer le nom utilisé pendant le test par la variable "Personne"




6. Lancez le test.

L'éditeur de tests propose de nombreuses fonctionnalités que nous ne pourrions pas aborder dans ce cours :

- la définition de paramètres d'entrée et de sortie du test.
- la possibilité de créer une bibliothèque de tests pour tester un exécutable par exemple sur une machine différente du poste de développement.

Pour plus de détails sur les tests automatiques, nous vous conseillons de consulter l'aide en ligne (mot-clé : "Test automatique").

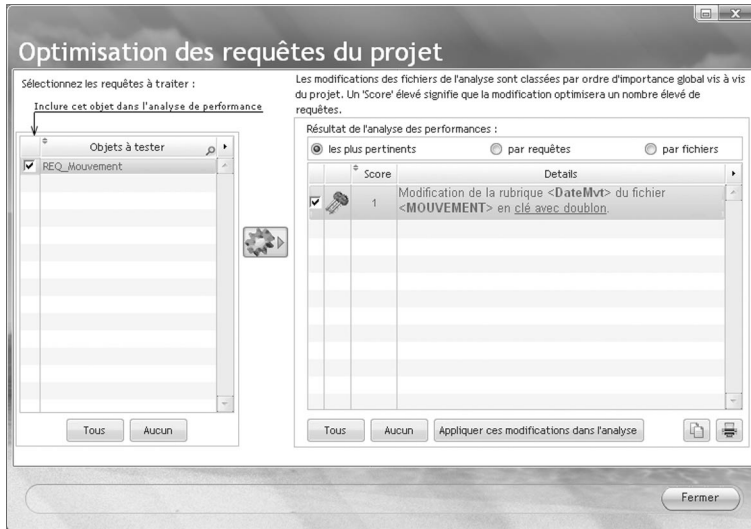
Optimisation des requêtes

Revenons au tableau de bord de notre application (bouton ). Dans la section "Optimisation", un voyant s'intitule "Optimisation des requêtes". Nous allons voir si les requêtes de l'application peuvent être optimisées.

1. Cliquez sur le voyant "Optimisation des requêtes". Une fenêtre listant les différentes requêtes du projet apparaît.
2. Sélectionnez les différentes requêtes du projet (bouton "Tous") et cliquez sur le bouton



3. Le résultat est automatiquement affiché.



Pour optimiser les requêtes du projet, WinDev propose de créer des clés composées. Ces clés composées doivent être créées dans l'analyse. WinDev va réaliser ces modifications automatiquement.

4. Cliquez sur le bouton "Appliquer ces modifications dans l'analyse".

LEÇON 2.9. DÉPLOIEMENT DE L'APPLICATION

Ce que vous allez apprendre dans cette leçon ...

- Création de l'exécutable
- Aide des nouveautés
- Création de l'installation



Durée estimée : 20mn

Présentation

Notre première application est terminée. Maintenant, il faut générer l'exécutable et installer l'application chez les utilisateurs. C'est ce que nous allons faire pour conclure cette partie. Nous aurons ainsi vu la majorité des points importants lors du développement d'une application WinDev.

Création de l'exécutable

La création de l'exécutable est une opération très simple : une option de menu et quelques clics suffisent.

Nous vous proposons ici de détailler toutes les étapes de la création de l'exécutable afin de connaître toutes les possibilités de WinDev. À tout moment, vous pouvez cliquer sur le bouton vert pour valider la totalité des écrans de l'assistant.

► Pour créer l'exécutable :

1. Sélectionnez l'option "Atelier .. Générer l'exécutable Windows 32 bits".



Note

WinDev permet également de créer des exécutables 64 bits et des exécutables Linux.

WinDev 12 version "Express" : Ces fonctionnalités ne sont pas disponibles dans cette version d'évaluation.

2. L'assistant de création d'exécutable se lance. Un premier écran présentant l'état des tests automatiques est affiché. Dans notre cas, tous les tests automatiques ont été passés avec succès. Passez à l'écran suivant.
3. L'écran suivant permet de décrire les options générales de l'exécutable :
 - le nom de l'exécutable : nous conservons le nom par défaut.
 - le nom de l'icone associé à l'exécutable : vous pouvez sélectionner un icone dans le catalogue d'images.
 - l'image d'introduction. WinDev propose plusieurs types de splash screen. Par défaut, un splash screen avec texte animé est sélectionné. Le bouton "Options" permet de le configurer.
WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.
4. Passez à l'écran suivant. Pour le fonctionnement de l'exécutable, nous garderons les options par défaut. Passez à l'écran suivant.
5. L'écran suivant permet de personnaliser le message d'erreur de l'application. Nous gardons le message proposé par défaut. Passez à l'écran suivant.

6. L'écran suivant permet de définir l'utilisation du MCU dans notre application.



Note

Le MCU (Macro Code Utilisateur) donne la possibilité à l'utilisateur final de créer ses propres procédures en WLangage pour modifier le fonctionnement d'un champ, d'une fenêtre, ajouter un raccourci clavier sur une opération. Si le MCU est intégré dans l'application, l'utilisateur disposera d'un bouton supplémentaire dans la barre de titre pour saisir les codes supplémentaires.

Pour plus d'informations, consultez l'aide en ligne.

Nous garderons les options par défaut. Passez à l'écran suivant.

7. L'écran suivant permet de dire si l'exécutable prendra en compte les patches.



Note

Lors d'une modification de l'application, pour éviter de livrer la totalité de l'exécutable, il est possible de livrer des ressources complémentaires (fenêtres, états, ...) sous forme de patch. Ces patches sont en fait des bibliothèques additives.

Si l'exécutable a été créé avec l'option "Prendre en compte les mises à jour effectuées par Patch", lors du lancement de l'application, les éléments présents dans le patch se substitueront aux éléments présents dans la bibliothèque de l'application.

Pour plus d'informations, consultez l'aide en ligne.

Nous garderons les options par défaut. Passez à l'écran suivant.

8. L'écran suivant permet de gérer les langues de l'exécutable. Nous aborderons le sujet du multilingue dans une prochaine leçon. Nous garderons les options par défaut. Passez à l'écran suivant.

9. L'écran suivant récapitule tous les fichiers qui seront intégrés dans la bibliothèque de l'exécutable. Ce sont tous les éléments de votre projet qui peuvent être manipulés par l'utilisateur final. Nous garderons les options par défaut. Passez à l'écran suivant.

10. L'écran suivant concerne le répertoire des fichiers de données. Si votre application doit être déployée sur Windows Vista, il est conseillé de choisir l'option "Répertoire des données de l'application".



Note

Sous Windows Vista, le système de l'UAC implique l'utilisation des normes de programmation Windows.

Pour plus d'informations, consultez l'aide en ligne.

Sélectionnez l'option correspondant à votre système. Passez à l'écran suivant.

11. Nous allons maintenant saisir les informations de version de l'exécutable. Ces informations sont affichées sous l'explorateur Windows en sélectionnant les propriétés du fichier.

Entrez par exemple les informations suivantes :

Création d'exécutable Windows

Infos de version

Les informations de version sont affichées dans l'onglet "Version" de la fenêtre des propriétés de l'exécutable accessible depuis l'explorateur de fichiers.

Société : PC SOFT

Description : Mes comptes

Version : 1.00Aa Exemple : 11.01Cb

Copyright : PC SOFT 2007

URL site Web : www.pcsoft.fr

Email : suggestions@pcsoft.fr

Si vous le souhaitez, la version sera automatiquement incrémentée lors des prochaines créations d'exécutable.

Incrémentation automatique de la version à chaque création d'exécutable

Passez à l'écran suivant.

12. L'écran suivant concerne le framework WinDev.

WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.



Note

Le framework correspond aux différentes bibliothèques nécessaires au fonctionnement de l'exécutable.

L'option "Utiliser uniquement le framework propre à l'application" permet d'utiliser uniquement les bibliothèques nécessaires.

L'option "Utiliser le framework WinDev commun" permet d'utiliser le framework commun à toutes les applications WinDev installées sur le poste. Avec cette option, le framework est installé une seule fois sur le poste (il est possible de le télécharger par Internet) et il est utilisé par toutes les applications WinDev. Cochez cette option.

Passez à l'écran suivant.

13. Un écran concernant Windows Vista apparaît. Cet écran permet d'intégrer un manifeste pour une utilisation sous Windows Vista.

Passez à l'écran suivant.

14. L'écran suivant concerne le centre de réutilisabilité. Ce centre est intéressant si l'application doit être déployée au sein d'une même entreprise.

Passez à l'écran suivant.

15. L'exécutable est créé. Il est possible de le lancer immédiatement, pour vérifier son fonctionnement.

Voilà, la création de l'exécutable est terminée. Beaucoup d'étapes, mais après une première configuration de vos choix, vous pouvez valider dès le début toutes les étapes.



Note

Il est également possible de cliquer directement sur les étapes indiquées dans l'assistant pour parvenir directement à un écran. Les options par défaut des autres écrans seront automatiquement validées.

Création de l'installation

La création de l'installation est une opération très simple : un assistant vous permet de définir les choix principaux. Si les options proposées par l'assistant ne suffisent pas, il est possible d'utiliser l'éditeur d'installation. Nous ne détaillerons pas son utilisation dans cette leçon. Pour plus de détails, consultez "Installation d'une application", page 492.

Nous vous proposons ici de détailler toutes les étapes de la création du programme d'installation afin de connaître toutes les possibilités de WinDev. À tout moment, vous pouvez cliquer sur le bouton vert pour valider la totalité des écrans de l'assistant.

- Pour créer le programme d'installation :

1. Sélectionnez l'option "Atelier .. Créer la procédure d'installation". L'assistant de création d'exécutable et d'installation se lance.

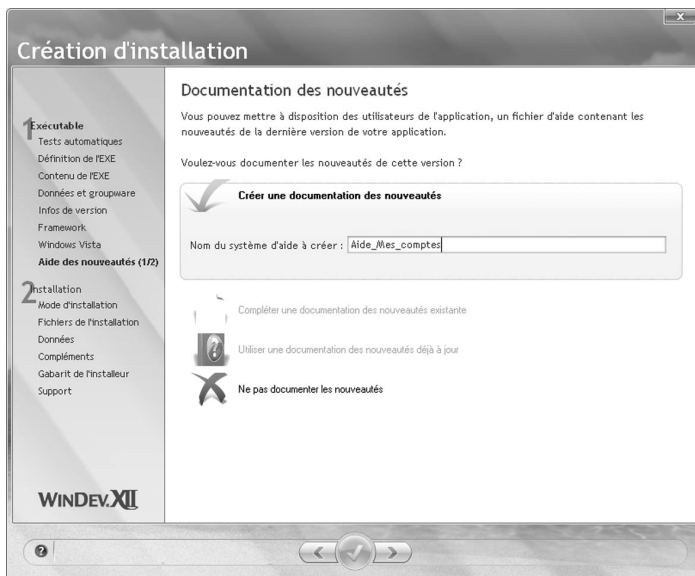


Note

Nous avons déjà créé l'exécutable, cliquez directement dans l'assistant sur l'intitulé "Aide des Nouveautés".
WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

2. Un écran s'affiche proposant de créer la page des nouveautés. Cette option permet de créer un fichier d'aide pour présenter à l'utilisateur final les nouveautés de la version. Lors d'une pre-

mière installation, ce fichier peut correspondre à l'aide du logiciel.



Sélectionnez l'option "Créer une documentation des nouveautés" et passez à l'écran suivant. **WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.**

3. L'écran suivant permet de définir les éléments qui seront automatiquement intégrés au fichier d'aide. Conservez les options par défaut et passez à l'écran suivant.

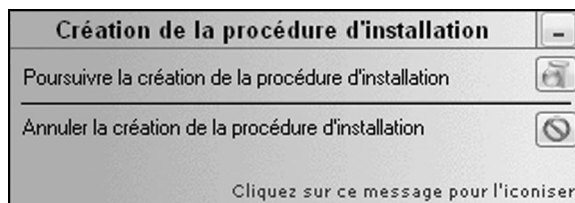


Note

Si l'application possède déjà un système d'aide, il est possible de l'utiliser pour le compléter avec une page de nouveautés.

WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

4. Validez le message. L'exécutable est automatiquement créé (avec les options que nous avons définies lors de la création de l'exécutable), et le système d'aide est créé. Vous pouvez saisir les informations spécifiques dans le système d'aide. Pour l'exemple, laissez les informations créées par défaut. Pour reprendre la création du programme d'installation, cliquez sur l'option "Poursuivre la création de la procédure d'installation".



L'aide est automatiquement compilée.

WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

5. Choisissez le mode d'installation :

- Installation autonome pour une application indépendante, installée et lancée sur chaque poste. Nous choisissons cette option.
- Installation avec mise à jour automatique, pour une installation sur un poste serveur. Les applications seront installées à partir du serveur. En cas de mise à jour, seule une mise à jour du serveur est nécessaire. Les applications installées se mettront à jour automatiquement. Passez à l'écran suivant.

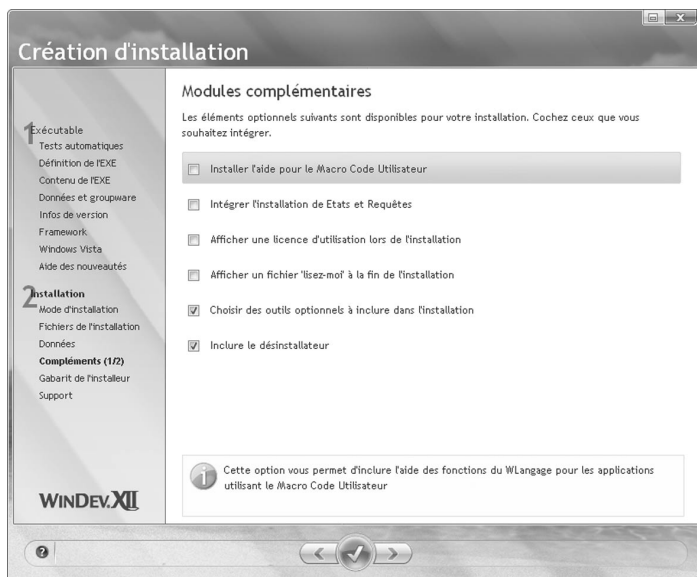
6. Choisissez une installation standard. Passez à l'écran suivant.

7. Nous ne changeons pas le répertoire d'installation par défaut. Passez à l'écran suivant.

8. Dans la liste des fichiers installés, conservez les fichiers proposés. Passez à l'écran suivant.

9. Dans les paramètres de la base de données, décochez toutes les options. Passez à l'écran suivant.

10. Dans les modules complémentaires, conservez les options suivantes :



Nous allons choisir les outils optionnels à installer et inclure le désinstallateur. Grâce au désinstallateur, les utilisateurs pourront désinstaller l'application depuis le gestionnaire de programmes de Windows. Passez à l'écran suivant.

11. Nous allons sélectionner WDOptimiseur. Cet outil permet de réindexer régulièrement les fichiers de données de l'application.

12. Cliquez sur l'option "Support" à gauche de l'assistant. Nous allons générer l'installation dans un répertoire unique. Par défaut, le programme d'installation est créé dans le sous-répertoire "Install" du projet. Il est possible de spécifier un mot de passe pour l'installation. Seul l'utilisateur connaissant le mot de passe pourra lancer l'installation de l'application.

13. Validez. Le programme d'installation est automatiquement créé.

14. Testez immédiatement le programme d'installation créé.

Conclusion

Voilà, nous avons réalisé une application de sa conception sous l'éditeur d'analyse à sa distribution en clientèle. Nous avons découvert de nombreuses fonctionnalités de WinDev. Les chapitres suivants vont permettre d'approfondir certaines de ces fonctionnalités.

LEÇON 2.10. QUESTIONS / RÉPONSES

Ce que vous allez apprendre dans cette leçon ...

- Trucs et astuces

Question Comment créer le menu principal de mon application ?

Effectuez les opérations suivantes :

1. Créez une nouvelle fenêtre ou ouvrez une fenêtre existante.
2. Sélectionnez l'option "Fenêtres .. Menu principal .. Ajouter le menu principal". Une option est créée automatiquement.
3. Faites un clic droit sur cette option.
4. Cliquez sur "Description de l'option" pour modifier l'option sélectionnée.
5. Cliquez sur "Ajouter après" pour ajouter une nouvelle option après.
6. Cliquez sur "Ajouter avant" pour insérer une nouvelle option avant.
7. Cliquez sur "Insérer un sous-menu" pour ajouter un sous-menu dans l'arborescence du menu.

Question Comment lier une fenêtre à une option de mon menu principal ?

La fonction **Ouvre** permet d'associer une fenêtre à une option de menu. Saisissez le code suivant dans le code de clic de votre option de menu :

```
Ouvre (MAFENETRE)
```



Notes

Pour associer un état à une option de menu, utilisez la fonction **ilprimeEtat** :
`iImprimeEtat (MONETAT)`

Question Comment insérer automatiquement le menu "?" ?


Pour insérer automatiquement un menu "?" dans votre application, sélectionnez l'option "Fenêtres ... Menu principal .. Ajouter le menu '?'" et sélectionnez les options désirées dans l'assistant qui se lance.

Question Comment créer un menu contextuel ?


Un menu contextuel peut être ajouté :

- soit au niveau de la fenêtre.
- soit au niveau d'un champ.

Pour une fenêtre :

1. Faites un clic droit sur la fenêtre et sélectionnez l'option "Description".
2. Cliquez sur l'onglet "IHM" et cliquez sur l'icone  à côté de la combo "Menu contextuel".

Pour un champ :

1. Faites un clic droit sur le champ et sélectionnez l'option "Description".
2. Cliquez sur l'onglet "IHM" et cliquez sur l'icone  à côté de la combo "Menu contextuel".

Pour connaître ou modifier le menu contextuel d'un champ ou d'une fenêtre par programmation, utilisez la propriété **MenuContextuel**.

Question Comment fermer une fenêtre automatiquement après une durée prédéfinie ?

Il suffit d'utiliser un bouton et la fonction **DélaiAvantFermeture**. Par exemple :

```
DélaiAvantFermeture("FENETRE", "BOUTON", 200)
```

Pour plus de détails, consultez l'aide en ligne (mot-clé : "Fermer, Fermeture automatique (bouton)"). Notez que vous pouvez également utiliser l'option "Validation automatique" disponible dans l'onglet "Détail" de la description de la fenêtre.

Question Comment dialoguer avec l'utilisateur ?

Il suffit d'utiliser une boîte de dialogue avancée. Ces boîtes de dialogue permettent de gérer :

- l'interrogation directive (fonction **Dialogue**) : l'utilisateur répond à une question grâce à des boutons contenant le texte de l'action à effectuer.
- la saisie immédiate (fonction **Saisie**), en proposant à l'utilisateur de saisir directement dans la boîte de dialogue la valeur voulue

Pour plus de détails sur ces fonctions, consultez l'aide en ligne (mot-clé : "Dialogue" et "Saisie").

Question Comment récupérer des paramètres passés en ligne de commande à un exécutable ?

Il suffit d'utiliser la fonction **LigneCommande** dans le code d'initialisation du projet.

Consultez l'aide en ligne pour plus de détails sur cette fonctionnalité (mot-clé : "Ligne de commande").



Astuce

Vous pouvez simuler le passage de paramètres en ligne de commande à votre projet en mode test. Dans l'éditeur :

1. Cliquez sur l'option "Projet .. Mode test .. Ligne de commande du mode test".
2. Saisissez les paramètres de la ligne de commande.

Question Comment désinstaller une application réalisée avec WinDev ?

Lors de la création du programme d'installation, il est possible de prévoir la désinstallation de l'application.

Si cette option a été choisie, le programme de désinstallation a automatiquement été créé. Votre application a donc été enregistrée auprès de Windows pour pouvoir la désinstaller ultérieurement.

Pour désinstaller une application :

1. Cliquez dans le menu "Démarrer".
2. Sélectionnez le choix "Panneau de configuration".
3. Sélectionnez "Ajout/Suppression de programmes".
4. Sélectionnez l'application et cliquez sur "Désinstaller".

Question Comment créer une installation par CD-ROM ?

Lors de la création de l'installation (option "Atelier .. Créer la procédure d'installation"), vous avez la possibilité de choisir différents supports d'installation. En sélectionnant l'option "CD-ROM", WinDev créera un dossier dans lequel tous les fichiers nécessaires à une installation par CD-ROM seront créés (fichier AUTORUN, fichiers d'installation, ...).

Vous n'avez plus alors qu'à graver le contenu de ce dossier complet sur un CD-ROM vierge pour distribuer votre application !

Question Comment créer un exécutable ?

L'exécutable de votre projet peut être créé en sélectionnant l'option "Atelier .. Générer l'exécutable Windows".

Question Comment installer une application ?

Lorsque l'exécutable a été réalisé (option "Atelier .. Générer l'exécutable"), le répertoire EXE présent dans le répertoire de votre projet contient tous les éléments permettant de faire fonctionner votre application.

Pour préparer une installation de votre application :

1. Sélectionnez l'option "Atelier .. Créer la procédure d'installation". L'assistant de préparation d'une installation démarre.
2. Suivez les instructions à l'écran.

Question Comment associer un icone à mon exécutable ?

L'icone qui est associé à votre exécutable peut être défini lors de la création de l'exécutable. Cet icone doit être au format ICO.



Notes

Un catalogue d'icônes prédéfinis est fourni en standard avec WinDev. Ce catalogue est accessible lors de la sélection de l'icone.

Question Comment associer une image de lancement (splash screen) à mon exécutable ?

L'image de lancement qui est associée à votre exécutable peut être définie lors de la création de l'exécutable. Cette image doit être à un format standard reconnu par WinDev (BMP, WMF, GIF, JPEG, TIFF, ...)

Un catalogue d'images prédéfinies est fourni en standard avec WinDev. Ce catalogue est accessible lors de la sélection de l'image.

WinDev offre la possibilité de personnaliser directement cette image lors de la création de l'exécutable. Vous pouvez ainsi écrire le texte de votre choix (avec la mise en forme voulue) sur cette image.

WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

Question Comment mettre l'icone de mon application en bas à droite dans la barre de tâches ?

Utilisez la fonction **SysIcôneAjoute**. Exemple :

```
Res = SysIcôneAjoute("C:\Icônes\Icône.ICO", "MenuContextuel", ...
                  "Message survol", "Nom Procédure associée")
```

Pour rétablir votre application, utilisez la fonction **SysIcôneSupprime**.

Pour plus de détails sur ces fonctions, consultez l'aide en ligne (mot-clé : "SysIcôneAjoute").

Question Comment installer un raccourci d'une application sur le bureau ?

La fonction **CréeRaccourci** permet de créer un raccourci d'une application par programmation. Cette fonction permet de créer aussi bien un raccourci sur le bureau que dans un groupe de programmes.

Par exemple, pour installer un raccourci sur le bureau :

```
CréeRaccourci(raccourciBureau, "Libellé de mon icône", ...
"C:\Program Files\Compta\Compta.EXE")
```

Consultez l'aide en ligne pour la syntaxe détaillée de cette fonction (mot-clé : "Créer, Un raccourci").

Question

Comment partager le framework WinDev entre les différentes applications installées sur un même poste ?

Lors de la création du programme d'installation de l'application, vous avez la possibilité de dire si votre application partage le Framework WinDev avec les autres applications installées sur le poste. Dans ce cas, le Framework WinDev sera installé dans le répertoire "C:\Program Files\Fichiers communs\PC SOFT\12.0\Framework".

Consultez l'aide en ligne pour plus de détails (mot-clé : "Framework").

Question

Comment détecter les éléments non utilisés par mon application ?

Après quelques mois ou quelques années de développement et de maintenance, le répertoire de votre projet contient souvent plusieurs fichiers qui ne sont plus utilisés, mais que, dans le doute, vous n'osez pas supprimer.

Fichiers et fenêtres de test, images abandonnées, ... Un nettoyage s'impose !

Un outil de WinDev permet d'une part de détecter automatiquement ces éléments inutilisés et d'autre part de les supprimer du projet, sans aucun risque. Les éléments supprimés du projet seront archivés (au format ZIP, ou dans un répertoire de sauvegarde) pour le cas où une de ces ressources aurait un jour une utilité...

- Pour utiliser cet assistant, sélectionnez l'option "Outils .. Nettoyer le répertoire du projet".

Remarque : Il est possible de connaître le code mort et les éléments orphelins grâce à l'option "Projet .. Performances et améliorations .. Code mort, orphelins et liens brisés".

DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 3

Fenêtres et champs

XIII ***Express***

LEÇON 3.1. LES ÉDITEURS

Ce que vous allez apprendre dans cette leçon ...

- Les éditeurs
- Les volets
- Les raccourcis
- Les configurations d'environnement
- Le menu personnalisé



Durée estimée : 10 mn

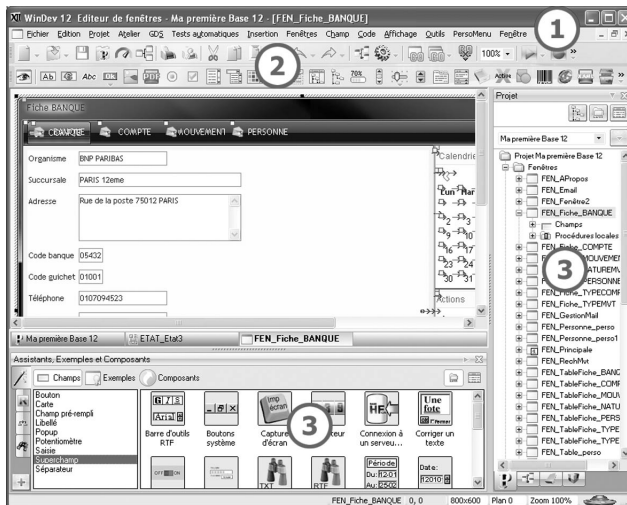
L'environnement

Depuis le début de ce cours, nous avons créé/manipulé des fenêtres. Ces manipulations ont été effectuées sous l'éditeur de fenêtres.

L'éditeur de fenêtres permet de réaliser simplement de superbes fenêtres reliées ou non aux données.

WinDev possède de multiples éditeurs, tous intégrés dans le même environnement : éditeur de code, éditeur de requêtes, éditeur d'états, ...

Tous les éditeurs utilisent le même environnement :



1. Barre de menu
2. Barre d'outils
3. Volets

Voici quelques astuces pour profiter pleinement de l'environnement de WinDev.

Les volets

WinDev propose différents volets affichables à tout moment. Ces volets peuvent être affichés ou non, regroupés dans l'ordre de votre choix. Vous pouvez ainsi définir la configuration qui correspond le mieux à votre environnement et la rappeler à tout moment.

► Pour afficher des volets :



1. Sélectionnez l'option "Affichage .. Barres d'outils".
2. Sélectionnez le volet à afficher ou à rendre invisible (une coche est présente devant le nom des volets actuellement affichés sous l'éditeur).



Astuce

Une barre d'outils spécifique pour la gestion des volets est disponible. Pour l'afficher, sélectionnez l'option "Affichage .. Barre d'outils .. Contrôle des volets".

Chaque volet présent dans l'interface peut être :

- **flottant** : le volet peut être redimensionné et déplacé indépendamment. Les volets flottants sont par exemple utiles si vous utilisez deux écrans.
- **regroupé avec d'autres volets** : il suffit de déplacer un volet vers un autre volet. Lorsque l'icône représentant des onglets apparaît (), il suffit de positionner le volet sur cet icône et de relâcher la souris.
- **ancré à un des bords de l'interface** : il suffit de déplacer un volet vers un des bords de l'interface. Lorsque l'icône représentant un des bords de l'interface apparaît (), il suffit de positionner le volet sur cet icône et de relâcher la souris.
- **invisible** : Si le volet est flottant, il suffit de cliquer sur la croix (x) de fermeture. Si le volet est regroupé, il suffit de faire un clic droit sur l'onglet correspondant et de sélectionner "Fermer l'onglet".

Les configurations d'environnement

La configuration de vos volets vous convient? Vous voulez la mémoriser? Rien de plus simple : vous pouvez sauvegarder jusqu'à 4 configurations grâce aux touches [CTRL] + [ALT] + [1], [CTRL] + [ALT] + [2], ... Et pour afficher la configuration voulue, il suffit d'utiliser la touche [CTRL] suivie du numéro de configuration voulue.



Astuce

La combinaison de touches [CTRL]+[W] permet de faire apparaître ou disparaître les volets regroupés et ancrés.
Il est ainsi possible d'avoir rapidement un espace de travail plus étendu.

Le menu personnalisé

Qui n'a jamais souhaité disposer d'un menu "à soi", entièrement personnalisable?

WinDev le propose désormais en standard.

Pour comprendre comment ça marche, rien ne vaut un exemple pratique.

Nous allons créer un menu personnalisé permettant de :

- lancer la calculatrice Windows
- afficher la description du projet.

- ▶ Première étape : nous allons créer l'option de menu permettant de lancer directement la calculatrice de Windows depuis l'éditeur WinDev :
 1. Sélectionnez l'option "PersoMenu .. Personnaliser le <PersoMenu>". La fenêtre de personnalisation du menu s'affiche.
 2. Cliquez sur le bouton "Ajouter une option".
 3. Saisissez son nom dans le champ "Libellé du menu" ("Calculatrice" par exemple).
 4. Cliquez sur le bouton "Parcourir" et recherchez l'exécutable de la calculatrice (dans "C:\Windows\system32" par exemple).
 5. Saisissez le message à afficher dans la barre de message lors du survol de l'option ("Lance la calculatrice directement" par exemple).
 6. Sélectionnez l'icône à afficher à gauche de l'option dans le menu : il suffit de sélectionner à

nouveau l'exécutable de la calculette dans le champ "Icône".

- ▶ Seconde étape : nous allons rajouter une deuxième option afin de lancer directement le fichier "<Nom du projet>.RTF" créé automatiquement lors de la création d'un nouveau projet. Ce fichier contient la description du projet définie dans l'assistant de création du projet.
 1. Cliquez sur le bouton "Ajouter une option".
 2. Saisissez son nom dans le champ "Libellé du menu" ("Description du projet" par exemple).
 3. Ce fichier se trouve dans le répertoire du projet en cours. Cliquez sur le bouton "Macro" et double-cliquez sur "{\$RepProjet} Chemin du répertoire du projet en cours". La chaîne "{\$RepProjet}" apparaît automatiquement dans le champ "Script de commande".
 4. Saisissez à la suite de cette chaîne le caractère "\".
 5. Cliquez sur le bouton "Macro" et double-cliquez sur "{\$NomProjet} Nom du projet en cours". La chaîne "{\$NomProjet}" apparaît automatiquement dans le champ "Script de commande".
 6. Saisissez à la suite de cette chaîne les caractères ".RTF".
 7. Saisissez le message à afficher dans la barre de message lors du survol de l'option ("Ouvre la description du projet" par exemple).
 8. Validez la fenêtre de configuration du menu personnalisable (bouton "OK").

- ▶ Votre menu personnalisé est fini. Testez-le!

LEÇON 3.2. LES FENÊTRES

Ce que vous allez apprendre dans cette leçon ...

- Comment créer une fenêtre?
- Les caractéristiques d'une fenêtre



Durée estimée : 10 mn

Comment créer une fenêtre?

Depuis le début de ce cours, nous avons créé des fenêtres de différentes façons :

- création de fenêtres vierges basées sur un gabarit (partie 1 de ce cours)
- création de fenêtres de différents types (fiche, table, ...) basée sur les fichiers de données (partie 2 de ce cours).

WinDev propose d'autres possibilités pour créer des fenêtres. Ces possibilités sont regroupées dans l'assistant de création de fenêtres, lancé par l'option "Fichier .. Nouveau .. Fenêtre".

L'assistant de création de fenêtres permet :

- de créer des **fenêtres RAD** (Rapid Application Development) : ces fenêtres sont basées sur les fichiers de données décrits dans l'analyse et contiennent tout le code nécessaire à leur fonctionnement. Un gabarit ou un modèle RAD peut être automatiquement utilisé lors de la création de la fenêtre. Ces fenêtres peuvent être utilisées immédiatement. Plusieurs types de fenêtres sont proposés :
 - fenêtre de type fiche
 - fenêtre avec table
 - fenêtre avec zone répétée, ...
- de créer des **fenêtres RID** (Rapid Interface Development) : ces fenêtres sont basées sur les fichiers de données décrits dans l'analyse. Elles contiennent uniquement les champs et les boutons. La saisie du code correspondant est effectuée par le développeur. Un gabarit ou un modèle RAD peut être automatiquement utilisé lors de la création de la fenêtre. Ces fenêtres peuvent être utilisées immédiatement. Plusieurs types de fenêtres sont proposés :
 - fenêtre de type fiche
 - fenêtre avec table, ...
- de créer des fenêtres standards : ces fenêtres correspondent aux boîtes de dialogue le plus souvent utilisées (OuiNon, Erreur, Info, Mot de passe, ...).



Notes

Boîte de dialogue

Pour dialoguer avec l'utilisateur, WinDev propose de nombreuses fonctions permettant :

- d'informer l'utilisateur (Info)
- de demander une confirmation (Confirmer, OKAnnuler, Dialogue)
- de demander une information spécifique (Saisie)

- de créer des **fenêtres internes**. Les fenêtres internes sont un type spécifique de fenêtre. Nous détaillerons leur fonctionnement dans la suite de cette partie.
- de créer des **fenêtres basées sur un modèle de fenêtres**. Les modèles de fenêtres permettent de définir un ensemble de critères (graphique, champ, code) devant être repris dans chaque fenêtre de l'application. Nous détaillerons la création et l'utilisation d'un modèle de fenêtres dans la suite de cette partie.
- d'**importer une fenêtre existante** (et non WinDev). Cette option permet de "recopier" l'interface d'une fenêtre dans votre application WinDev. Vous sélectionnez la fenêtre qui vous intéresse et WinDev se charge de tout. L'option "Générer des images pour les champs non reconnus" permet de rendre l'interface plus proche de la fenêtre importée.

Bien sûr, vous avez toujours la possibilité de créer une fenêtre vierge, sans aucun champ, utilisant ou non un gabarit.

Après avoir vu les différentes façons de créer une fenêtre, nous allons voir quelles sont les caractéristiques d'une fenêtre.

Description d'une fenêtre : 8 onglets à votre disposition

Dès le début de ce cours d'auto-formation, vous avez créé des fenêtres. La première chose que vous avez affichée, c'est la fenêtre de description afin de saisir le titre de la fenêtre, son nom et sa description.

Mais la fenêtre de description renferme de nombreux paramétrages très pratiques, qui évitent bien souvent des dizaines de lignes de code.

Nous allons détailler quelques-unes de ces fonctionnalités sur un exemple simple.

Exemple pratique pour nos manipulations

Pour apprendre à utiliser ces différents types de champs, nous allons utiliser un projet exemple préparé à cet effet. Pour ouvrir ce projet sous WinDev :

1. Sélectionnez l'option "Fichier .. Ouvrir un projet".
2. Sélectionnez le sous-répertoire "AutoFormation\Corrigés\Fenêtres et champs" du répertoire d'installation de WinDev 12.
3. Sélectionnez le fichier "Fenêtres et Champs.WDP". Le projet se charge.
4. Ouvrez la fenêtre "FEN_Manip".



Notes

Ouvrir une fenêtre sous l'éditeur

Nous avons vu plusieurs méthodes pour ouvrir une fenêtre sous l'éditeur. Voici un récapitulatif des différentes possibilités :

- Option "Fichier .. Ouvrir" et sélection du fichier de la fenêtre à ouvrir.
- Utilisation du raccourci clavier [CTRL] + [E] et sélection de la fenêtre à ouvrir. L'avantage de cette méthode est la visualisation de l'aperçu de la fenêtre.
- Affichage du volet "Projet" et double-clic sur le nom de la fenêtre à ouvrir.

5. Affichez la description de la fenêtre (option "Description" du menu contextuel de la fenêtre). L'onglet affiché par défaut est l'onglet "Général".

Onglet "Général"

Le premier onglet de la fenêtre de description est l'onglet "Général". Cet onglet permet d'indiquer les paramètres généraux de la fenêtre :

- **le nom de la fenêtre** : ce nom sera utilisé en programmation pour manipuler la fenêtre.
- **la description de la fenêtre** : cette description est purement indicative pour le développeur. Elle permet de résumer un peu plus longuement ce à quoi sert la fenêtre. Elle sert également à décrire la fenêtre lors de l'impression du dossier.
- **le titre de la fenêtre** : comme nous l'avons déjà vu, le titre est affiché dans la fenêtre et permet d'indiquer à l'utilisateur le sujet traité.

Onglet "IHM"

Le second onglet de la fenêtre de description est l'onglet "IHM". Cet onglet permet de définir les paramètres concernant l'interface de la fenêtre. Cet onglet regroupe ainsi les caractéristiques concernant :

- la taille et la position de la fenêtre
- le menu contextuel de la fenêtre,
- le curseur de survol et l'action du clic droit sur la fenêtre, ...

Dans cet onglet, de nombreuses options sont intéressantes :

- **Déplacement par le fond** : cette option (sélectionnée par défaut lors de la création de la fenêtre) permet à l'utilisateur de déplacer la fenêtre en cliquant à n'importe quel endroit (et non uniquement par la barre de titre). Vous pouvez d'ailleurs tester cette option directement sur la fenêtre de description.
- **Redimensionnable** : grâce à cette option, l'utilisateur pourra redimensionner la fenêtre. Mais cette option nécessite une gestion précise du redimensionnement des champs (appelé ancrage). Nous consacrerons un chapitre à cette notion dans la leçon concernant l'ergonomie des fenêtres.
- **Mémoriser la taille et la position de la fenêtre** : cette option est très intéressante car la position de la fenêtre déterminée par l'utilisateur sera automatiquement enregistrée et utilisée à la prochaine ouverture de la fenêtre. Ainsi, si l'utilisateur dispose de plusieurs écrans, l'affichage de la fenêtre sur un écran ou un autre sera automatiquement mémorisé.
- **Assombrir la fenêtre lorsqu'elle devient inaccessible** : encore une option très appréciée par les utilisateurs. Si des boîtes de dialogue sont affichées par l'application, l'utilisateur ne se demande plus où il doit cliquer : en effet, la fenêtre en arrière-plan est automatiquement grisée pour que l'utilisateur puisse se concentrer sur le message affiché.

Onglet "Détail"

Dans cet onglet, les paramètres spécifiques au fonctionnement de la fenêtre sont regroupés. On retrouve les caractéristiques concernant :

- Le type de la fenêtre (nous reviendrons en détail sur cette caractéristique dans la leçon sur l'ergonomie),
- Les paramètres avancés de la fenêtre (contexte Hyper File, ...)
- L'animation de la fenêtre en ouverture et / ou en fermeture
- Exécution automatique d'un bouton de la fenêtre

Arrêtons-nous un instant sur ces deux derniers points.

L'animation de fenêtre permet de donner un aspect dynamique à votre application. Vous pouvez paramétrer l'ouverture et / ou la fermeture de votre fenêtre. Pour paramétrer l'animation ? Rien de plus simple : il suffit de cliquer sur les caractéristiques de l'animation. La fenêtre de paramétrage apparaît. Saisissez vos paramètres et testez-les immédiatement (et même sur la fenêtre de paramétrage).

La **fermeture automatique de la fenêtre ou la validation automatique** permet de ne pas laisser votre application bloquée sur une fenêtre. En effet, un utilisateur peut par exemple afficher une fiche Client et la laisser ouverte pendant une absence. Pour éviter le blocage des autres utilisateurs, il suffit de définir une durée et l'action à réaliser au bout de cette durée (clic sur le bouton Valider, Annuler, ...).

Onglet "Image"

Cet onglet permet de définir les différentes images utilisées par la fenêtre :

- l'image de fond de la fenêtre et son mode d'affichage.
Vous pouvez utiliser une image définissant la forme de votre fenêtre (par exemple pour créer une fenêtre reprenant la forme d'une télécommande). Dans ce cas, il suffit de sélectionner l'option "Détourer la fenêtre" et de suivre les indications.
Il est également possible d'utiliser une image qui s'agrandira avec votre fenêtre. Il suffit de définir le mode "9 images" associé.
- L'image de la poignée de redimensionnement de la fenêtre.
- L'icone associé à la fenêtre : cet icone est visible dans le coin en haut à gauche de la fenêtre au niveau de la barre de titre. Il personnalise la fenêtre. Si aucun icone n'est indiqué, c'est l'icone associé au projet qui sera visible.

Onglet "Langue"

Cet onglet permet d'indiquer les différentes langues à gérer dans la fenêtre (Cas des projets "multilingues". Pour plus de détails, consultez la leçon "Multilingue", page 507).

Onglet "Note"

Cet onglet permet de saisir du texte. Ce texte est imprimé lors de l'édition du dossier. Vous pouvez par exemple, saisir des informations concernant le fonctionnement de la fenêtre, les différentes règles de gestion des traitements effectués.

Onglet "Aide"

Cet onglet permet d'indiquer le nom du fichier d'aide contextuelle associé à la fenêtre. Ce fichier d'aide est utilisé lors de l'appui sur le bouton d'aide de la fenêtre pour activer une aide contextuelle au clic sur la fenêtre.

Onglet "Style"

Cet onglet regroupe les paramètres relatifs au style de la fenêtre. On peut sélectionner :

- le gabarit
- les icones affichés (agrandissement, réduction, aide, menu système, ...)
- le type de bordure
- la couleur de fond
- le thème XP
- l'opacité (permet de définir la transparence d'une fenêtre)
- la présence de la barre de message, ...

Comme vous pouvez vous en rendre compte, les onglets les plus importants sont les 3 premiers onglets.

LEÇON 3.3. LES CHAMPS

Ce que vous allez apprendre dans cette leçon ...

- Les différents types de champs
- Les champs les plus classiques
- Les champs spécialisés



Durée estimée : 30 min

Introduction

WinDev met plus de 30 champs à votre disposition pour communiquer avec l'utilisateur final. Ils permettent de saisir ou de visualiser des valeurs.

Les valeurs affichées peuvent provenir d'un calcul effectué par programmation, d'un fichier d'une base de données ou d'une affectation.

Les valeurs saisies peuvent servir à faire des calculs, à être enregistrées dans un fichier de données ou à être affectées à d'autres champs.

Cette leçon va vous permettre de connaître les différents types de champs disponibles, de les tester dans des cas d'utilisation réelle, et pour les plus courants, d'aborder leur programmation.

Pour simplifier cette leçon, nous avons découpé les différents champs en deux catégories :

- les champs classiques : ce sont les champs les plus utilisés.
- les champs spécialisés, qui permettent d'obtenir une interface spécifique, ou de gérer des fonctionnalités spécifiques.

Exemple pratique

Pour apprendre à utiliser ces différents types de champs, nous allons utiliser un projet exemple préparé à cet effet. Pour ouvrir ce projet sous WinDev :

1. Sélectionnez l'option "Fichier .. Ouvrir un projet".
2. Sélectionnez le sous-répertoire "AutoFormation\Corrigés\Fenêtres et champs" du répertoire d'installation de WinDev 12.
3. Sélectionnez le fichier "Fenêtres et Champs.WDP". Le projet se charge.

Les champs classiques

Sous l'appellation "Champs classiques" nous avons regroupé tous les champs les plus communs, utilisés constamment dans les applications créées avec WinDev. Ces champs sont les suivants :

- Libellé
- Champ de saisie
- Bouton
- Liste
- Combo
- Table
- Zone répétée
- Table hiérarchique
- Image
- Sélecteur
- Interrupteur
- Jauge
- Graphe

Nous allons étudier ces champs un par un.

Type de champ : Libellé

Résumé

Les libellés permettent d'afficher du texte statique. Ils ne peuvent pas être saisis en exécution mais leur contenu peut être sélectionné pour effectuer des "copier/coller". Ils peuvent être modifiés par programmation (comme tous les champs).

Les libellés sont utilisés pour afficher une information, par exemple un titre en gros caractères. Le contenu d'un libellé peut être défini sous l'éditeur lors de la description du libellé ainsi que par programmation.

Les libellés permettent également d'afficher un texte en fonction de la langue utilisée dans le projet, sans programmation particulière.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampLibelle.WDW" et testez cette fenêtre.
Cette fenêtre présente :
 - les différents types de libellés pouvant être utilisés.
 - les différentes manipulations pouvant être effectuées sur un libellé.
- ▶ Testez les différentes manipulations des libellés.
Le libellé est affiché en vert.
Le bouton "Couleur" change la couleur du libellé.
Le bouton "Récupère" récupère le libellé affiché.
Les boutons "Change" et "Construit" modifient le libellé affiché.
- ▶ Retournez sous l'éditeur.

Les différents types de libellés

WinDev propose plusieurs options afin de créer le libellé qui vous convient :

- **les libellés au format RTF** : ce type de libellé permet de gérer toutes les caractéristiques du format RTF (Gras, italique, souligné, couleurs, ...). Pour qu'un libellé soit au format RTF, il suffit de cocher l'option "Texte avec mise en forme (R.T.F)" dans l'onglet "Détail" de la description du champ.
- **les libellés animés** : Faire clignoter ou défiler un libellé est un jeu d'enfant. Il suffit de choisir le mode d'animation et de configurer les paramètres correspondants dans l'onglet "Détail" de la description du champ.
- **les libellés avec ombre floue** : une ombre floue apparaît derrière le libellé afin de créer un effet de relief. La configuration de l'ombre floue est réalisée dans l'onglet "Style" de la description du champ.

Manipulation des libellés par programmation

Nous allons étudier le code des boutons permettant de modifier le champ libellé.

Pour construire le libellé "LIB_LibelléManipulé" (bouton "Construit"), il faut adopter la syntaxe suivante (dans cet exemple la chaîne est construite à partir de chaînes et du résultat de la fonction **DateDuJour** qui retourne la date du jour).

```
LIB_LibelléManipulé = "Coucou" + RC + "Nous sommes le " + ...
                    DateVersChaîne (DateDuJour ())
```

Pour récupérer le libellé de "LIB_LibelléManipulé" (bouton "Récupère"), la syntaxe est :

```
Ch est une Chaîne
Ch = "Le libellé est "+ RC + LIB_LibelléManipulé
// ou
// Info("Le libellé est "+ RC + LIB_LibelléManipulé)
```

Il est même possible de changer la couleur du texte du libellé (bouton "Couleur") avec la propriété **Couleur**.

```
LIB_LibelléManipulé..Couleur = RougePastel
```

RougePastel est une constante WLangage correspondant à la couleur Rouge Pastel.



Notes

Les propriétés permettent de modifier par programmation certaines caractéristiques des champs : couleur, libellé, police, ... Nous ne citerons pas toutes les propriétés. Nous découvrirons quelques-unes des principales propriétés au fur et à mesure de cette partie.

Toutes les propriétés sont relatives à un champ donné. Leur syntaxe d'utilisation est :

```
Nom_Champ..Nom_Propriété
```

La liste des propriétés disponibles sur les champs est présente dans l'aide en ligne de WinDev, mot-clé : "Propriétés, Propriétés des champs d'une fenêtre".

Type de champ : Champ de saisie

Résumé

Les champs de saisie sont les premiers champs que vous avez manipulés. Ils permettent de saisir ou d'afficher des données provenant de variables, de calculs ou de fichiers.

Ils permettent par exemple de saisir un mot de passe, une quantité à commander, le nom d'un client, une adresse, ...

Si vous créez une fenêtre, il y a 9 chances sur 10 pour que vous utilisiez ensuite un champ de saisie. Voyons un peu plus en détail leur fonctionnement et leurs possibilités.

Quelle que soit l'information à saisir ou à afficher, vous pouvez choisir le type correspondant :

- texte,
- numérique,
- heure,

- date,
- durée,
- monétaire,
- monétaire+Euro.

À chaque type de champ correspondent plusieurs masques de saisie (ou d'affichage si le champ est en affichage). Nous avons vu l'intérêt de ce masque lors du développement de l'application "Mes comptes", nous n'y reviendrons pas.



Rappel

Les champs de saisie numériques ne sont pas typés. C'est le masque sélectionné pour le champ qui va définir si le champ est un réel, réel double, entier, ...

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampSaisie.WDW" sous l'éditeur. Cette fenêtre présente la gestion des champs de saisie de type texte et des champs numériques.
- ▶ Testez la fenêtre.
- ▶ Saisissez des valeurs dans les champs qui ont le libellé "En saisie". Dans les autres champs, il n'est pas possible de saisir. En effet, ils ont été décrits en affichage. Cliquez sur les différents boutons et observez ce qu'il se passe.

En détail

Pour afficher et récupérer une valeur dans un champ de saisie une simple affectation suffit et ce quel que soit le type du champ de saisie.

Quelques exemples d'initialisation :

- Code d'initialisation d'un champ de saisie texte :

```
SAI_EnAffichage = "Coucou" //Affiche Coucou dans le champ texte
```

- Code d'initialisation du champ de saisie numérique :

```
//Affiche la valeur 20.6 dans le champ numérique  
SAI_Numérique = 20.6
```



Rappel

Il est également possible d'initialiser la valeur d'un champ de saisie en utilisant l'onglet "Contenu" de la description du champ.

Quelques exemples de récupération :

- Code de clic d'un bouton pour récupérer la valeur d'un champ de saisie texte :

```
Ch est une Chaîne  
Ch = SAI_EnSaisie //Récupère le contenu du champ texte
```

- Code de clic d'un bouton pour récupérer la valeur d'un champ de saisie numérique :

```
Valeur est un Réel
//Récupère le contenu du champ numérique
Valeur = SAI_Numérique
```

Gestion de la couleur dans les champs de saisie

Dans la zone de saisie, la couleur du texte et la couleur du fond sont modifiables par programmation avec les propriétés **Couleur** et **CouleurFond**.

```
//Change la couleur du texte
SAI_Manip..Couleur = RougeClair
//Change la couleur du fond
SAI_Manip..CouleurFond = JauneClair
```

Caractéristiques d'un champ de saisie

Comme pour les fenêtres, les champs possèdent une fenêtre de description. Cette fenêtre de description permet de paramétrer de nombreuses options du champ, options d'aspect ou de fonctionnement.

Nous allons détailler les caractéristiques du champ de saisie, sachant que de nombreuses options sont reprises dans la description d'autres champs. Nous ne détaillerons pas toutes les options ici (c'est un cours, pas une documentation exhaustive). Pour plus de détails sur chacune des options des fenêtres de description, il est conseillé de consulter l'aide contextuelle.

Onglet "Général"

Comme nous l'avons déjà vu, l'onglet "Général" du champ de saisie permet de définir :

- le nom du champ
- son libellé
- son type
- son masque de saisie.

Quelques options peuvent se révéler utiles :

- **Texte avec mise en forme (R.T.F)** : Cette option permet d'avoir des mots en Italique, en Gras, en couleur, de tailles et de polices différentes, ... Il suffit de copier / coller dans le champ un contenu au format RTF ("Rich Text Format") pour que celui-ci soit pris en compte.
- **Mot de passe** : Si cette option est cochée, les caractères frappés sont remplacés par des "*" uniquement à l'affichage.



Astuce

Ce type d'option est idéal pour la saisie du mot de passe d'un utilisateur dans une fenêtre de login.

- **Texte multi-lignes** : cette option permet de saisir un texte sur plusieurs lignes. Des RC (Retours Chariots) peuvent être insérés dans le texte.



Astuce

Ce type d'option est idéal pour saisir un commentaire ou une observation. L'utilisation de ce paramètre est fondamentale pour l'utilisation des champs "Mémo" (étudiés plus loin).

Onglet "IHM"

L'onglet IHM permet de définir les caractéristiques d'interface du champ.

Nous allons détailler les points suivants :

- Etat initial
- Gestion de la touche TAB
- La lettre d'appel et clic droit

L'**état initial du champ** correspond à l'état du champ lors de l'exécution de la fenêtre. Cette caractéristique se retrouve pour presque tous les types de champ.

Au lancement d'une fenêtre, un champ de saisie peut être :

- **En saisie** : le champ est actif, il sera en saisie dans la fenêtre. L'utilisateur pourra saisir des informations dans ce champ.
- **Affichage seul** : le champ est inactif, aucune saisie ne sera possible. Ce champ se comportera comme un champ de type libellé. Il sera cependant possible de sélectionner le contenu du champ à l'aide de la souris si l'option "Avec sélection en affichage" est cochée dans l'onglet "Détail".
- **Grisé** : le champ est inactif et il apparaît en grisé. Cet état est très pratique pour indiquer à l'utilisateur que ce champ existe, mais qu'il n'est actuellement pas accessible.
- **Visible** : Comme son nom l'indique, le champ "existe" dans la fenêtre. Ce champ peut être visible ou non. Cet état est souvent utile en programmation lorsqu'un champ doit apparaître à certains moments et disparaître à d'autres!

Cette option est indépendante des 3 options précédentes. Vous pouvez rendre ainsi le champ visible ou non tout en conservant les paramètres d'affichage.

L'état du champ peut être modifié par la suite par programme en utilisant :

- la propriété **Etat** pour les options "Saisie", "Affichage seul" et "Grisé",
- la propriété **Visible** pour l'option "Visible".

La **gestion de la touche TAB** permet de configurer la prise en compte de la touche TAB dans la fenêtre. Le plus souvent, la touche TAB du clavier permet à l'utilisateur de se déplacer entre les différents champs de la fenêtre (option "Champ accessible par TAB"). Il est alors possible de définir l'ordre de navigation dans la fenêtre (nous le verrons dans le chapitre Ergonomie de cette partie).

WinDev permet cependant de ne pas gérer la touche TAB ou d'accepter les TAB en saisie. Cette dernière option peut être intéressante dans des champs multi-lignes permettant de saisir de longs textes.

La **"lettre d'appel"** permet à l'utilisateur de se positionner directement sur le champ en tapant simplement une combinaison de touches du clavier (par exemple [ALT] + [Lettre]).

Dans le libellé d'un champ de saisie, si une lettre est précédée de "&", elle devient automatiquement "lettre d'appel".

Par exemple : "&Nom du client" décrit [ALT]+[N] comme lettre d'appel.

Le traitement du **"clic droit"** (de la souris) permet d'associer une action au clic droit de la souris. Le traitement est par exemple "validation des informations saisies", "appel de l'aide", ... Le traitement est choisi parmi ceux des boutons (texte et graphique) présents dans la fenêtre.



Notes

Parfois il est plus utile d'utiliser les lettres d'appel plutôt que les actions de souris. WinDev permet de réaliser facilement des interfaces (IHM) où l'utilisation de la souris n'est pas nécessaire. Par exemple, pour des applications de points de vente (la place manque sur les comptoirs!) ou dans le domaine industriel : bornes interactives, machine-outil, ...

Onglet "Détail"

L'onglet permet de définir les différentes fonctionnalités s'appliquant au champ de saisie.

Quelques points importants :

- Dans le cas d'une saisie multi-lignes, il est possible de sélectionner le mode de défilement des informations dans le champ (onglet "Détail" de la fenêtre de description).

Si le champ est en "**Défilement Horizontal ET Vertical**", le passage à la ligne suivante s'effectue par la combinaison de touches [CTRL] + [ENTREE] (standard Windows) ou [ENTREE].



Notes

Le saviez-vous? La taille maximale d'une variable texte est de 2 Go. En tapant 3 lettres par seconde, il faudrait ... plus de 22 ans pour remplir le champ!

- Pour les champs de type numérique, il est possible de saisir une **valeur minimale et maximum** (dans l'onglet "Détail" de la fenêtre de description du champ). Lors de la saisie dans ce champ, WinDev contrôlera automatiquement que ces valeurs sont respectées.
- **Saisie obligatoire** : si la saisie de ce champ est obligatoire, il sera possible de se positionner sur un autre champ uniquement si une valeur a été saisie.
- **Saisie assistée** : si le champ est lié à une rubrique clé, la valeur trouvée dans le fichier et correspondant aux premières lettres saisies est automatiquement proposée.
- **Vérifier l'orthographe** : si Open Office et ses dictionnaires sont installés sur le poste de l'utilisateur, la correction orthographique sera automatiquement proposée à l'utilisateur.
- **Mémoriser la valeur** : permet de mémoriser la dernière valeur saisie dans le champ avant la fermeture de la fenêtre. Cette valeur sera automatiquement proposée à la prochaine ouverture de la fenêtre.
- **Historique des saisies** : permet de proposer les dernières valeurs saisies par l'utilisateur pour ce champ.
- **Retourne NULL si vide** : retourne la valeur **NULL** si le champ n'a pas de valeur (Attention! Ni 0, ni chaîne vide!).



Astuce

La valeur **NULL** est très utile pour gérer le passage de paramètres à une requête et rendre les conditions optionnelles pour des recherches multi-critères. Nous aborderons cela plus tard lors de la leçon sur l'éditeur de requêtes.

- **Mode ellipse** : Permet de tronquer le contenu du champ si nécessaire. Si le champ est tronqué "en fin", les caractères "..." remplaceront la fin du texte à l'affichage.

Type de champ : Bouton

Résumé

Les boutons sont utilisés pour lancer des traitements. Le traitement associé au bouton s'exécutera lorsque le bouton sera activé (lorsque l'utilisateur clique sur le bouton).



Les boutons peuvent contenir un texte simple et/ou une image.

Vous avez déjà créé plusieurs boutons.

Revenons sur les types de boutons proposés par WinDev.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampBouton.WDW" sous l'éditeur. Cette fenêtre présente les différents types de boutons. Nous allons étudier tous les types de boutons disponibles.

Les différents types de boutons

Un bouton peut être de 5 types (informations à renseigner dans la fenêtre de description, onglet "IHM") :

1. Normal
2. Validation
3. Interruption
4. Abandon
5. Aide

Bouton NORMAL

Un bouton de type **Normal** est un bouton comme ceux que vous avez déjà créés.

Ce type de bouton n'a pas d'action prédéfinie particulière. Le code de sortie du champ qui était en cours lors de la validation du bouton est exécuté avant le traitement du bouton.

- ▶ Pour le vérifier, cliquez sur le bouton "Normal". Le message "Code de sortie du champ NOM" s'affiche et ensuite le message "Clic sur le bouton Normal" s'affiche.



Astuce

Les boutons de type NORMAL ne doivent pas être utilisés dans des fenêtres contenant des champs.

Les boutons de type NORMAL peuvent être utilisés dans des barres d'outils, des barres d'icônes ou dans des fenêtres sans champs de saisie.

Bouton VALIDATION

Lorsqu'un bouton est en **Validation**, la frappe de la touche [RC] ou [Entrée] exécute le traitement du bouton, le code de sortie du champ qui était en cours est exécuté.

- Pour le vérifier, repositionnez-vous sur le champ "Nom" puis cliquez sur le bouton "Validation".



Astuce

Un bouton de type VALIDATION doit être utilisé pour effectuer la validation de la saisie d'une fenêtre.

Dès qu'une fenêtre permet de faire une saisie dans un champ de saisie, un bouton de type VALIDATION doit être présent. C'est généralement le type utilisé pour les boutons "OK".

Bouton INTERRUPTION

Un bouton en **Interruption** signifie que lorsque le bouton est activé, le code de sortie du champ qui était en cours n'est pas exécuté. Le code du bouton s'exécute, puis le curseur est repositionné sur le champ.

- Pour le vérifier, repositionnez-vous sur le champ "Nom" puis cliquez sur le bouton "Interruption". Seul le code du bouton s'exécute, le code de sortie du champ ne s'exécute pas



Astuce

Un bouton INTERRUPTION sera par exemple utilisé pour appeler une fenêtre (afin d'interrompre la saisie en cours sans valider) puis revenir pour continuer la saisie. **C'est généralement le type utilisé pour les boutons "Aide".**

Bouton ABANDON

Lorsqu'un bouton est en **Abandon**, la frappe de la touche [ESC] ou [Echap] exécute le traitement du bouton, le code de sortie du champ qui était en cours n'est pas exécuté.

- Pour le vérifier, repositionnez-vous sur le champ "Nom" puis cliquez la touche "Echap".



Astuce

Un bouton ABANDON sera utilisé pour "sortir" de la fenêtre en cours de saisie sans effectuer aucune validation.

Dans une fenêtre de saisie, un bouton de type ABANDON est nécessaire pour abandonner la saisie en cours sans la valider.

C'est généralement le type utilisé pour les boutons "Annuler".



Attention!

La "croix" est identique à un bouton de type Abandon. Lors du clic sur la "croix", la fenêtre est fermée en exécutant le code du bouton Abandon de la fenêtre. Le code de fermeture de la fenêtre est ensuite appelé. L'équivalent de cliquer sur la "croix" est d'appuyer sur la combinaison de touches [Alt]+[F4].

Bouton AIDE

Lorsqu'un bouton est de type **Aide**, la validation du bouton affiche automatiquement l'aide associée au champ en cours (activée normalement par [F1]) sans exécuter les codes de sortie et d'entrée du champ en cours.

- Pour le vérifier, cliquez sur le bouton "Aide".

Récapitulatif	
Si votre fenêtre ne comporte pas de champs de saisie	
Type de bouton à utiliser	Tous les boutons peuvent être de type NORMAL
Si votre fenêtre a au moins un champ de saisie	
Type de bouton à utiliser	Un bouton de type VALIDATION
	Un bouton de type ABANDON
	Un bouton de type AIDE (si les champs ont des fenêtres d'aide associées)
	Tous les autres boutons de type INTERRUPTION

Caractéristiques d'un bouton

Action associée à un bouton

Un bouton peut être associé à une action prédéfinie. WinDev propose de nombreuses actions prédéfinies : ouvrir une fenêtre du projet, fermer une fenêtre, ouvrir l'aperçu avant impression d'un état, lancer un site Internet, ... Pour toutes ces actions, aucune ligne de code n'est à saisir : WinDev s'occupe de tout.

Les actions peuvent être associées aux boutons dans l'onglet "Général" de la fenêtre de description.

Lettre d'appel d'un bouton

Un bouton peut être associé à une lettre d'appel. Cette lettre d'appel peut correspondre à une touche de fonction, une touche de déplacement ou encore une combinaison de touches.

La lettre d'appel associée au bouton est indiquée dans les informations concernant l'IHM du champ. Le bouton doit être de type "Interruption" (ou "Normal" dans une fenêtre ne permettant pas la saisie). Cela permet par exemple d'exécuter un traitement, d'ouvrir une fenêtre, ... sous l'action d'une touche.

- ▶ Dans notre fenêtre, appuyez sur [F2], une fenêtre de type "Info" s'affiche.



Astuce

Dans certaines applications, pour éviter l'utilisation de la souris, les traitements sont déclenchés par les touches de fonctions.

Pour cela, il suffit de créer les boutons hors écran et de leur associer une lettre d'appel.

Pour créer un bouton hors écran, il suffit de placer ce bouton dans la fenêtre d'accueil de la fenêtre sous l'éditeur.

Bouton minuterie

Un bouton de votre fenêtre peut être défini comme bouton minuterie : au bout d'une certaine durée, le bouton s'exécutera automatiquement. Ce type de bouton est très utile pour ne pas bloquer des enregistrements (par exemple, la fiche d'un client, pendant la pause déjeuner) : il suffit de définir le bouton annuler comme bouton minuterie.

Image d'un bouton

Il est possible d'associer des images aux boutons. Deux types d'images peuvent être utilisés :

- l'image qui illustre le bouton. Dans ce cas, le bouton est un bouton graphique.



- l'image qui définit la forme et le fond du bouton



L'image du bouton peut être :

- une image que vous avez créée
- choisie parmi celles proposées dans le catalogue de WinDev (dans les "clipart" par exemple).

Il est aussi possible d'utiliser pour le bouton une image avec un fond transparent. Dans ce cas, vous pouvez :

- soit créer une image dont la couleur de fond est "Magenta clair".
- soit définir que la couleur du point (pixel) en haut à gauche de l'image est la couleur transparente.



Astuce

Nous vous conseillons de :

- ne pas donner de libellé à un bouton graphique dont l'image comporte un libellé, sinon le bouton comportera deux libellés.
- ne pas mélanger dans une même fenêtre, les boutons graphiques et les boutons texte.

Le libellé d'un bouton texte ou d'un bouton graphique peut être modifié par programmation avec la propriété **Libellé**. Si le libellé contient le caractère "&", la lettre qui suit sera soulignée et deviendra la lettre d'appel du bouton :

```
BTN_Texte.Libellé = "Nouveau &libellé"
```

De la même façon, l'image d'un bouton graphique peut être modifiée par programmation :

```
BTN_Image.Libellé = "MonImage.BMP"
```

La propriété **Libellé** reconnaît automatiquement si c'est le libellé ou l'image qui doit être modifié.

Type de champ : Image**Résumé**

Les champs images permettent d'afficher des images, statiques ou animées. Il est ainsi possible d'afficher les images d'un catalogue de produits, la photo d'une personne, ...

Ces images peuvent être au format BMP, JPEG, GIF, PCD, PSD, TGA, TIF, PNG, ICO (icônes Windows), EMF, WMF, CUR, OS2 (BMP sous OS/2), PCX.

Tous les modes sont supportés du mode "256 couleurs" au mode "couleurs vraies" (16 bits, 24 bits ou 32 bits).

Les champs images peuvent également servir de zones de clicage.

Exemple pratique










- ▶ Ouvrez la fenêtre "FEN_ChampImage.WDW" sous l'éditeur. Cette fenêtre présente différents types d'images :
 - image statique.
 - gif animé et image animée.
 - image clicable.
- ▶ Testez cette fenêtre.

Les différents types d'images**Images statiques**

Les images statiques sont les images les plus courantes. Pour utiliser une image statique dans une application WinDev, il suffit de créer un champ image et d'associer à ce champ un fichier image. Cette image peut être sélectionnée dans un des répertoires de votre disque, ou provenir du catalogue d'images.

WinDev permet de paramétrer directement le mode d'affichage de l'image (onglet "Général" de la fenêtre de description) :

Mode d'affichage :	<input type="text" value="100%"/>	Transparence :	<input type="text" value="Aucune"/>
Position :	<input type="text" value="Haut gauche"/>	Symétrie :	<input type="text" value="Aucune"/>

Mode d'affichage	Exemple	Description
100 %		L'image s'affiche avec sa taille initiale. La taille du champ peut être modifiée mais cela n'a aucune incidence sur l'image.
Centré		L'image est centrée dans le champ image. La taille du champ peut être modifiée. Si l'image est plus petite (en hauteur ou en largeur) que le champ, des bandes apparaissent sur les côtés.
Etiré		L'image est déformée pour occuper toute la surface du champ. La taille de l'image s'adapte à celle du champ.
Répété		L'image dans sa taille initiale est répétée autant de fois que possible pour occuper toute la surface du champ image.
Homothétique		L'image est agrandie proportionnellement pour être affichée entièrement dans le champ image. L'image est cadrée en haut à gauche.
Homothétique étendu		L'image est agrandie proportionnellement pour que le plus petit coté de l'image soit affiché entièrement dans le champ image
Homothétique centré		L'image est agrandie ou réduite (selon le cas) en conservant les proportions hauteur/largeur pour occuper la plus grande surface possible du champ. L'image n'est pas déformée.
Homothétique centré étendu		L'image est agrandie proportionnellement pour que le plus petit coté de l'image soit affiché entièrement dans le champ image, et centrée dans le champ image.
Homothétique sans agrandissement		L'image peut être réduite si nécessaire mais ne dépassera pas sa taille originale si le champ d'accueil est trop grand.

L'option "Affichage Haute Qualité" permet en mode homothétique d'améliorer la qualité de l'image lorsque celle-ci est réduite par rapport à sa taille initiale.

Le catalogue d'images

Dès qu'une image peut être affichée dans un champ, une fenêtre, WinDev propose d'utiliser le catalogue d'image grâce au bouton "CATALOGUE". Ce catalogue contient plusieurs centaines d'images, de cliparts, ...

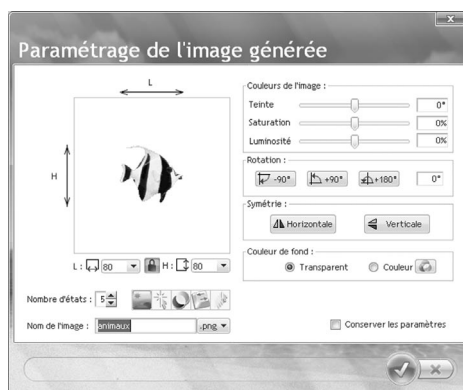
Pour sélectionner un clipart, vous pouvez effectuer une recherche selon un mot-clé.

Dans la fenêtre de recherche :

- Saisissez un mot-clé (par exemple "Animaux")
- Indiquez si nécessaire le thème et la catégorie de recherche.
- Cliquez sur le bouton de recherche (la loupe). Les images se rapprochant le plus du mot-clé s'affichent dans la fenêtre.
- Choisissez l'image que vous préférez en cliquant dessus (par exemple un poisson) et validez.



- Indiquez les caractéristiques de l'image à générer : taille, luminosité, extension, ...



Le nom de l'image spécifiée s'affiche alors dans la fenêtre de description de l'image et le fichier de l'image (au format voulu) est copié dans le répertoire du projet.

WinDev gère des images avec un fond transparent. Dans ce cas, il faut :

- soit créer une image avec une couleur de fond "Magenta Clair" (couleur RVB : 255, 0, 255). Le magenta est automatiquement considéré comme la couleur de transparence.
- soit créer une image dont la couleur du pixel (point) en haut à gauche est la couleur de référence pour la transparence. Vous êtes maître de la couleur "transparente".



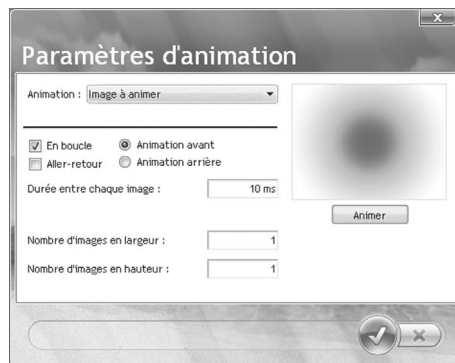
Attention !

La couleur "transparente" est analysée dans la totalité de l'image. Il peut donc y avoir des zones transparentes en plein centre de l'image. Cela peut donner des effets d'affichage, vérifiez les images !

GIF animé et image animée

Pour animer une image, WinDev propose 2 solutions :

- utiliser une image au format GIF animé : l'animation est contenue dans le fichier image.
 - utiliser une image contenant les différents dessins de l'animation.
- Dans la fenêtre "FEN_ChampImage.WDW", affichez la description du champ contenant l'image de la terre. L'onglet "Général" contient uniquement le nom de l'image à afficher. Ce fichier est au format GIF. L'animation est contenue dans le fichier.
- La fenêtre "FEN_ChampImage.WDW" utilise également une image animée par WinDev. Le champ "Une image non animée" contient l'image dans sa totalité : cette image est composée de 6 dessins en largeur, et de 4 dessins en hauteur. Nous allons voir en détail le paramétrage utilisé pour créer l'image animée.
- Affichez la description du champ image contenant l'image de l'imprimante. Dans l'onglet "Détail" du champ, l'option "Animé à la création" est cochée. Cette option permet d'indiquer que le champ contient une image à animer dès l'affichage du champ. Cliquez sur la flèche située à côté de l'option "Animation : Image à animer". La fenêtre de configuration de l'animation s'affiche.



Cette fenêtre contient tous les paramètres de l'animation. On retrouve dans cette fenêtre le nombre de dessins utilisés en largeur et en hauteur. Pour tester ces paramètres sur l'image en cours, cochez la case "Utiliser l'image du champ". Vous pouvez modifier les paramètres de l'animation et tester immédiatement leur effet.

- Fermez la description du champ image.



Exemple

Pour plus de détails sur l'utilisation de tous les paramètres proposés, consultez l'exemple "WDAAnimated" (exemple didactique), livré avec WinDev (accessible depuis le volet "Assistants, Exemples et Composants").



Astuce

Les paramètres d'animation peuvent être décrits directement dans le champ, mais peuvent aussi être modifiés par programme à l'aide des fonctions du WLanguage : **AnimationChangeParamètre**, **AnimationChangeType** et **AnimationLitParamètre**. Consultez l'aide en ligne pour une utilisation approfondie de ces fonctions (mots-clés : "Animation, Animer une image").

Image cliquable

WinDev permet de transformer vos images en images cliquables en ... un clic de souris. En effet, il suffit de :

1. Afficher la description de l'image.
 2. Dans l'onglet "Détail", sélectionner les options :
 - "Cette image est une zone de clicable" : cette option permet de définir que l'image peut être cliquée
 - "Prioritaire pour les clics souris" : cette option permet de gérer le clic sur l'image quelle que soit l'altitude du champ (premier plan ou non).
 3. Un nouveau code est alors associé à l'image : Code de clic sur l'image. Dans ce code, vous pouvez gérer l'effet du clic sur le champ.
- Dans la fenêtre "FEN_ChampImage.WDW", les zones de clicable sont :
- l'image contenant la tête de tigre.
 - différents champs images placés sur la tête de tigre. Ces champs image n'ont pas d'image associée et permettent de définir les zones de clic.

Type de champ : Sélecteur

Résumé

Les sélecteurs d'options sont également appelés "case d'options". Ils permettent de sélectionner une option et une seule, parmi celles proposées.

Le libellé des options proposées doit être saisi lors de la description du sélecteur sous l'éditeur. Ils sont modifiables par programmation.

Civilité

 Monsieur Madame Mademoiselle

Astuce !

Comment différencier le sélecteur et l'interrupteur ?

Le sélecteur est également appelé "Bouton radio". un moyen mnémotechnique simple : pensez aux anciennes radios : le bouton permettait de sélectionner une seule fréquence!

Le sélecteur permet de sélectionner une seule option.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampSelecteur.WDW" sous l'éditeur.
- ▶ Testez la fenêtre et cliquez sur les différents boutons, modifiez le sélecteur. Constatez ce qu'il se passe. Pour chaque bouton, vous pouvez visualiser le code exécuté.
- ▶ Retournez sous l'éditeur

À chaque option d'un sélecteur est associé un entier (qui peut prendre une valeur de 1 à N, N étant le nombre total d'options du sélecteur).

La première option a le numéro 1, la deuxième le numéro 2, ... Les options sont numérotées de haut en bas et de gauche à droite.

Dans notre fenêtre :

- "SEL_Civilité = 1" signifie que l'option "Monsieur" est sélectionnée,
- "SEL_Civilité = 2" signifie que l'option "Madame" est sélectionnée
- et "SEL_Civilité = 3" signifie que l'option "Mademoiselle" est sélectionnée.

Pour initialiser une case d'option, il suffit d'affecter une valeur entière au champ :

```
SEL_Civilité = 2 //Sélectionne "Madame"
```

Pour connaître l'option sélectionnée, il suffit de récupérer la valeur entière du champ :

```
Choix = SEL_Civilité //Récupère la valeur sélectionnée
```



Notes

Par défaut, l'option 1 du sélecteur est toujours sélectionnée. Il n'y a jamais aucun choix sélectionné.

Type de champ : Interrupteur**Résumé**

Les interrupteurs sont également appelés "cases à cocher". Ils permettent de choisir si la valeur de chaque option proposée est "vraie" ("Oui") ou "fausse" ("Non").

Vous aimez

 Lire Ecrire Danser Chanter

Le libellé des options proposées doit être saisi lors de la description de l'interrupteur sous l'éditeur. Il peut être modifié par programmation.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampSelecteur.WDW" sous l'éditeur.
- ▶ Testez la fenêtre et cliquez sur les différents boutons, modifiez l'interrupteur. Constatez ce qu'il se passe. Pour chaque bouton, vous pouvez visualiser le code exécuté.
- ▶ Retournez sous l'éditeur

Chaque option d'un interrupteur est "repérée" par un indice (qui peut prendre une valeur de 1 à N, N étant le nombre total d'options de l'interrupteur).

La première option a le numéro 1, la deuxième le numéro 2, ... Les options sont numérotées de haut en bas et de gauche à droite :

- "INT_Loisir[2] = Vrai" signifie : "Ecrire" est cochée.
- "INT_Loisir[3] = Faux" signifie : "Danser" n'est pas cochée.

Lorsqu'il y a plusieurs options, l'interrupteur se comporte comme un tableau d'options.

Pour accéder à une option, il suffit de passer l'indice de l'option en utilisant la syntaxe avec les crochets [].

Pour initialiser une case à cocher, il faut initialiser le choix correspondant avec la valeur "Vrai" ou "Faux" selon le cas.

```
INT_Loisir[2] = Vrai //pour cocher l'option 2 "Ecrire"
```

Pour connaître la valeur d'une option, il suffit de récupérer la valeur du champ ("Vrai" ou "Faux").

```
SI INT_Loisir[2] = Vrai ALORS
  //option cochée
SINON
  //option non cochée
FIN
```



Notes

Par défaut, aucune option n'est cochée dans un interrupteur.



Astuce

Les programmeurs habitués à manipuler des expressions booléennes peuvent omettre le test avec la valeur "Vrai" ou "Faux" :

```
SI INT_Loisir[2] ALORS
    //option cochée
SINON
    //option non cochée
FIN
```

Il est possible de modifier le libellé d'un interrupteur ou d'un sélecteur avec la propriété **Libellé**. Cependant, il n'est pas possible d'ajouter des options par programmation. Par exemple :

```
//Change le libellé
INT_Loisir..Libellé = "Vous aimez plutôt"
//Change la seconde option
INT_Loisir[2]..Libellé = "Ecrire avec un stylo"
```

Type de champ : Liste

Résumé

Les listes sont toujours déroulées. C'est pourquoi on les nomme également "listes déroulées". Elles permettent de visualiser plusieurs éléments et d'en sélectionner un ou plusieurs. Une liste peut contenir du texte et / ou des graphismes.

Lors de la création d'un champ "Liste", un assistant démarre et propose de créer :

- une liste dont le contenu est lié à un fichier ou une requête.
- une liste dont vous définissez le contenu dans l'éditeur ou par programmation.



Attention !

Règles à appliquer pour connaître le type de liste à créer :

- Utilisez une liste liée à un fichier ou à une requête lorsque vous souhaitez proposer à l'utilisateur une liste de valeurs qui sont contenues dans un fichier. Si la liste est liée à un fichier, c'est tout le contenu du fichier qui est proposé directement. Si la liste est liée à une requête, la liste contiendra une sélection d'enregistrements du fichier.
- Utilisez une liste dont le contenu est défini dans l'éditeur si les valeurs sont "STATIQUES". Elles ne changent pas durant le déroulement du programme.
- Utilisez une liste remplie par programmation lorsque le contenu de la liste est issu de calculs ou si la sélection des éléments à intégrer à la liste ne peut être réalisée par une requête.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampListe.WDW" sous l'éditeur. Cette fenêtre présente les différentes possibilités des listes et des combos. Nous allons étudier dans ce paragraphe les listes et leurs manipulations.
- ▶ Lancez le test de cette fenêtre.
 - La liste du haut est mono sélection.
 - Les deux listes du bas sont multi-sélections.



Notes

Liste mono sélection

Une liste mono sélection est équivalente à :

- une combo
- un sélecteur

Un seul choix est possible. Selon l'interface de votre application, choisissez le champ qui permet le plus facilement à l'utilisateur de sélectionner son choix.

- ▶ Cliquez sur les différents boutons et observez ce qu'il se passe. En exécution, dans une liste multi-sélections (selon le standard Windows) :
 - pour sélectionner plusieurs éléments un par un, appuyez sur la touche [CTRL] tout en sélectionnant les éléments avec la souris.
 - pour sélectionner plusieurs éléments consécutifs en même temps, cliquez sur le premier élément, appuyez sur la touche [Shift], maintenez cette touche enfoncée et cliquez sur le dernier élément de la sélection.
- ▶ Revenez sous l'éditeur.

En détail : Mode de remplissage d'une liste

Nous avons vu que l'assistant de création d'une liste proposait de créer deux types de liste :

- liste reliée à un fichier ou une requête (appelée liste fichier)
- liste remplie par programmation (appelée liste mémoire)

À chaque type de liste correspond un mode de remplissage spécifique.

Remplissage d'une liste mémoire

Si votre champ liste n'est pas lié à un fichier ou une requête, vous disposez de deux méthodes pour remplir la liste :

- sous l'éditeur,
- par programmation.

Sous l'éditeur, rien de plus simple : le contenu d'une liste peut être défini directement dans la fenêtre de description du champ (onglet "Général", champ "Contenu Initial"). Vous pouvez bien entendu saisir plusieurs valeurs. Pour aller à la ligne suivante, appuyez sur la touche [Entrée].

Pour remplir une liste **par programmation**, il faut utiliser la fonction **ListeAjoute**. Par exemple :

```
//Ajoute Lundi dans la liste LISTE_Jour
ListeAjoute(LISTE_Jour, "Lundi")
```

Il est par exemple possible de remplir une liste avec des informations contenues dans un tableau WLanguage :

```
//Remplit la liste avec le contenu d'un tableau
//Jour est un tableau de 7 chaînes
POUR i=1 A 7
    ListeAjoute(LISTE_Jour, Jour[i])
FIN
```

Remplissage d'une liste à partir d'un fichier ou d'une requête

Lors de la création d'un champ liste remplie à partir d'un fichier ou d'une requête, vous devez indiquer plusieurs informations :

- le fichier de l'analyse ou la requête qui servira de source de données pour le remplissage.



Notes

Si la requête n'existe pas lors de la création du champ, il est possible de créer la requête en même temps que le champ. La requête sera alors intégrée à la fenêtre qui contient le champ.

Attention : si vous utilisez une requête intégrée, cette requête sera utilisée uniquement pour ce champ. Elle ne pourra pas être réutilisée dans votre projet.

Ce fichier ou cette requête sera lu(e) automatiquement sans avoir à écrire une seule ligne de code.

- la rubrique à afficher dans la liste.
- la clé de parcours utilisée : cette rubrique sert de tri pour le remplissage de la liste et donc indique l'ordre dans lequel seront visualisées les valeurs de la liste.
- la valeur récupérée : lors de la sélection d'un élément de la liste, la valeur récupérée pourra être :
 - soit l'indice de la ligne sélectionnée,
 - soit une valeur du fichier.

Généralement, on choisit comme valeur récupérée une rubrique du fichier. Il s'agit le plus souvent de la clé unique, ce qui permet de faire par exemple des recherches dans les fichiers.

Caractéristiques de la liste

Lorsque la liste est créée, ses caractéristiques sont modifiables directement dans la fenêtre de description du champ (option "Description" du menu contextuel).

Vous retrouverez toutes les caractéristiques indiquées dans l'assistant, et d'autres options très intéressantes.

Par exemple, l'écran "Détail" permet de :

- définir si la liste est multisélection.
- définir si la liste est triée.

En détail : Programmation

La programmation des champs Liste est très simple avec WinDev : il suffit d'utiliser les fonctions WLangage commençant par les lettres LISTE. Ces fonctions peuvent être utilisées sur les listes mémoire et/ou fichier.



Notes

Vous ne savez pas si la fonction peut être utilisée sur une liste mémoire ou fichier ? Consultez l'aide en ligne! Utilisez la touche F1 sur le nom de la fonction.

Modification d'une liste mémoire

Pour modifier un élément dans une liste, deux méthodes sont disponibles :

- la fonction **ListeModifie**
- l'affectation directe en utilisant l'indice de l'élément à modifier.

```
ListeModifie(LISTE_Jour, "LUNDI") //Modifie l'élément en cours
ListeModifie(LISTE_Jour, "LUNDI", 1) //Modifie l'élément 1
LISTE_Jour[1] = "LUNDI" //Modifie l'élément 1
```

Remarque : Pour accéder à un élément d'une liste, il faut préciser son indice. Le premier élément d'une liste a l'indice "1".

Récupération de l'élément sélectionné

Dans une liste mono-sélection, pour récupérer l'élément sélectionné, il suffit de connaître l'indice de l'élément sélectionné. La fonction **ListeSelect** retourne l'indice de l'élément sélectionné. Si aucun élément n'est sélectionné, la fonction **ListeSelect** retourne la valeur "-1".

```
Indice est un entier
Jour est une chaîne
Indice = ListeSelect(LISTE_Jour)
Jour = LISTE_Jour[Indice] //Récupère l'élément sélectionné
//Ou bien
Jour = LISTE_Jour[LISTE_Jour] //Récupère l'élément sélectionné
```

Dans une liste multi-sélections, pour récupérer toutes les options sélectionnées, il faut utiliser la fonction **ListeSelect**. Tant que la fonction **ListeSelect** ne renvoie pas la valeur "-1", cette fonction retourne l'indice de l'option sélectionnée.

```
Indice est un entier = 0
Rang est un entier = 1 //rang de la sélection
//Choix est un tableau de chaîne contenant les sélections
BOUCLE
    Indice = ListeSelect(LISTE_Jour, Rang)
    SI Indice =- 1 ALORS SORTIR
    Choix[Rang] = LISTE_Jour[Indice]
    Rang = Rang + 1
FIN
```

Pour connaître le nombre d'éléments sélectionnés, il faut utiliser la fonction **ListeSelectOccurrence**.

Pour connaître le nombre d'éléments dans la liste, il faut utiliser la propriété **Occurrence** ou la fonction **ListeOccurrence** :

```
SI LISTE_Jour.Occurrence = 0 ALORS //liste vide
```

Suppression

Pour supprimer une valeur, il faut utiliser la fonction **ListeSupprime**.

Pour supprimer toutes les valeurs de la liste, il faut utiliser la fonction **ListeSupprimeTout**.

```
//Supprime l'élément sélectionné
ListeSupprime(LISTE_Jour)
```

Recherche

Pour chercher un élément dans une liste, il faut utiliser la fonction **ListeCherche**.

```
SI ListeCherche(LISTE_Jour, Valeur) = -1 ALORS
//n'existe pas
SINON
//existe déjà
FIN
```

Type de champ : Liste Image

Résumé

Une liste image est une liste permettant d'afficher à la fois du texte et des images. Un élément d'une liste image est composé de :

- un libellé
- une miniature (ou image)



Une liste image peut être affichée selon deux modes :

- mode liste image : chaque élément de la liste est composé d'un libellé et d'une image.
- mode classique : seul le libellé est affiché dans la liste image.

Comme pour une liste, lors de la création d'un champ "Liste image", vous pouvez créer :

- une liste image dont le contenu est lié à un fichier ou une requête.
- une liste image dont vous définissez le contenu par programmation.

On appliquera les mêmes règles que pour les listes pour choisir la méthode de remplissage.

Ce paragraphe présente uniquement les spécificités des listes image. Pour tout ce qui concerne le fonctionnement des listes image, consultez le paragraphe concernant les listes.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampListeImage.WDW" sous l'éditeur. Cette fenêtre présente les différentes possibilités des listes image.
- ▶ Lancez le test de cette fenêtre.
- ▶ Cliquez sur les différents boutons et observez ce qu'il se passe.
- ▶ Revenez sous l'éditeur et étudiez les traitements des boutons.

En détail : mode de remplissage de la liste image

Comme une liste, une liste image peut être une liste fichier (liée à un fichier de données ou à une requête) ou une liste mémoire.

Remplissage d'une liste image mémoire

Pour remplir une liste image mémoire, elle doit être remplie par programmation grâce à la fonction **ListeAjoute**. Par exemple :

```
ListeAjoute(LSI_ListeImage1,"image abandon.ico","abandon.ico")
//Ajoute l'image "abandon.ico" et le libellé "Image abandon.ico"
//dans la liste image LSI_ListeImage1
```

Remplir une liste image à partir d'un fichier ou d'une requête

Lors de la création d'un champ liste image remplie à partir d'un fichier ou d'une requête, vous devez indiquer le fichier de l'analyse ou la requête qui servira de source de données pour le remplissage. Ce fichier (ou cette requête) doit contenir :

- une rubrique qui correspondra au libellé de l'élément affiché dans la liste image.
- une rubrique qui correspondra à l'image de l'élément affiché dans la liste image.



Notes

Si la requête n'existe pas lors de la création du champ, il est possible de créer la requête en même temps que le champ. La requête sera alors intégrée à la fenêtre qui contient le champ.

Attention : si vous utilisez une requête intégrée, cette requête sera utilisée uniquement pour ce champ. Elle ne pourra pas être réutilisée dans votre projet.

Les autres éléments à indiquer dans l'assistant de création de la liste image sont identiques à ceux d'une liste simple.

Caractéristiques de la liste image

Lorsque la liste image est créée, ses caractéristiques sont modifiables directement dans la fenêtre de description du champ (option "Description" du menu contextuel).

Vous retrouverez tous les caractéristiques indiquées dans l'assistant, et d'autres options très intéressantes.

L'écran "Général" permet de paramétrer la taille d'une cellule de la liste image.

L'écran "Détail" permet par exemple de :

- définir si la liste image est multisélection.
- définir si la liste image est triée.
- donner l'orientation du champ.

En détail : Programmation

La programmation des champs Liste image est très simple avec WinDev : comme pour les listes, il suffit d'utiliser les fonctions WLangage commençant par les lettres LISTE. Ces fonctions peuvent être utilisées sur les listes image mémoire et/ou fichier.



Notes

Vous ne savez pas si la fonction peut être utilisée sur une liste image? Consultez l'aide en ligne! Utilisez la touche F1 sur le nom de la fonction.

De plus, certaines propriétés spécifiques aux listes image peuvent également être utilisées pour manipuler les images ou le mode d'affichage de la liste.

Modifier une liste image mémoire par programmation

Pour modifier un élément dans une liste image, vous pouvez utiliser :

- la fonction **ListeModifie**.

```
//Modifie l'élément en cours
ListeModifie(LSI_ListeImg1,"image abandon.ico","abandon.ico")
//Modifie l'élément 1
ListeModifie(LSI_ListeImg1,"image abandon.ico","abandon.ico", 1)
```

- les propriétés **Valeur** et **Miniature** sur l'élément à modifier.

```
// Modifie l'élément 2
LSI_ListeImg1[2]..Valeur = "Image Aide.ico"
LSI_ListeImg1[2]..Miniature = "Aide.ico"
```

Récupérer la valeur d'une liste image

Dans une liste image, il est possible de récupérer :

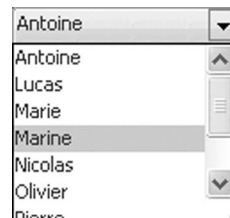
- le libellé de l'élément grâce à la propriété **Valeur**.
- l'image associée à l'élément, grâce à la propriété **Miniature**.

Type de champ : Combo

Résumé

Une combo est également appelée "combo box" ou encore "liste déroulante". Comme pour une liste, lors de la création d'un champ "Combo", vous pouvez créer :

- une combo "Fichier" dont le contenu est lié à un fichier ou une requête.
- une combo "Mémoire" dont vous définissez le contenu dans l'éditeur ou par programmation.



On appliquera les mêmes règles que pour les listes pour choisir la méthode de remplissage.

La partie déroulée de la combo peut correspondre à :

- une liste
- une liste image
- une table
- une fenêtre popup

Caractéristiques d'une combo

La combo se déroule lorsque le champ est sélectionné. Il existe 2 types de combos :

- les combos dans lesquelles un élément peut être sélectionné dans la liste. Ces combos sont appelées "**combo sans saisie**".
L'utilisateur déroule la liste pour sélectionner un élément.
- les combos dans lesquelles un élément peut être sélectionné dans la liste ou directement saisi dans le champ de la combo. Ces combos sont appelées "**combo avec saisie**".
L'utilisateur peut au choix saisir une valeur ou bien dérouler la liste pour sélectionner un élément.

Le type de la combo se définit dans la fenêtre de description de la combo.



Attention !

Au contraire des listes, les combos ne sont pas multi-sélections : un seul élément peut être sélectionné dans la combo.

Les combos peuvent contenir du texte et du graphisme.

Ce paragraphe présente uniquement les spécificités des combos. Pour tout ce qui concerne le fonctionnement des combos, consultez le paragraphe concernant les listes.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampListe.WDW" sous l'éditeur. Cette fenêtre présente les différentes possibilités des listes et des combos. Nous allons étudier dans ce paragraphe les combos et leurs manipulations.
- ▶ Lancez le test de cette fenêtre. La combo du haut est sans saisie et triée, celle du milieu est avec saisie et triée.

- ▶ Cliquez sur les différents boutons et observez ce qu'il se passe.
- ▶ Revenez sous l'éditeur et étudiez les traitements des boutons.

Caractéristiques de la combo

Lorsque la combo est créée, ses caractéristiques sont modifiables directement dans la fenêtre de description du champ (option "Description" du menu contextuel).

Vous retrouverez tous les caractéristiques indiquées dans l'assistant, et d'autres options très intéressantes.

L'écran "Général" permet de paramétrer si la combo est avec saisie ou non.

L'écran "Détail" permet par exemple de :

- définir les caractéristiques de la combo avec saisie
- définir si la combo est triée.
- proposer l'historique des sélections, dans les premières valeurs de la liste déroulante.
- permettre à l'utilisateur de modifier la largeur et la hauteur de la combo une fois déroulée!

En détail : programmation

La programmation des combos est identique à celle des listes. Les fonctions WLangage à utiliser commencent par les lettres LISTE. Des fonctions spécifiques aux combos (commençant par COMBO) sont également utilisables.

Le seul point particulier concerne le type de combo : "**avec saisie**" ou "**sans saisie**".



Attention !

Contrairement au WEB, une combo n'a jamais d'élément sélectionné par défaut, donc elle est vide. Il est conseillé de ne jamais laisser vide une combo, sauf si la combo est en saisie.

Cas d'une combo avec saisie

Pour affecter ou récupérer le champ d'une combo avec saisie, la syntaxe est la même que pour affecter ou récupérer un champ de saisie :

```
//Affecte la sélection de la combo  
COMBO_Prénom = "Julie"
```

```
//Récupère la sélection de la combo  
Ch est une Chaîne  
Ch = COMBO_Prénom
```

Il n'est pas possible de récupérer l'indice de l'élément sélectionné dans une combo avec saisie. On ne peut récupérer que la valeur de l'élément en cours de sélection dans la combo.

Cas d'une combo sans saisie

Pour récupérer l'élément sélectionné dans la combo, la méthode est similaire à une liste. Il faut utiliser la fonction **ListeSelect** pour récupérer l'indice de l'élément sélectionné. Ensuite, il faut utiliser

cet indice pour récupérer la valeur.

```
//Récupère l'élément sélectionné
Indice est un entier
Ch est une chaîne
Indice = ListeSelect (COMBO_Combol)
Ch = COMBO_Combol [Indice]

//Récupère le 3ème élément de la liste
Ch est une Chaîne
Ch = COMBO_Combol [3]
```

Pour sélectionner un élément, il faut utiliser la fonction **ListeSelectPlus**.

Type de champ : Table

Résumé

Il ne faut pas confondre "Table" et "champ Table".

On parle de **Table** pour une base de données de type SQL.

On parle de **champ Table** pour visualiser en tableau des données (ou "browse" ou encore "feuille de données"). Dans ce cas, ces données peuvent être saisies ou provenir d'un fichier (ou d'une table). Ouf!

Le champ table WinDev permet de visualiser ou de modifier des données :

- en mémoire : on parle alors de **table mémoire** ou table remplie par programmation.
- provenant de fichiers : on parle alors de **table reliée à un fichier** ou à une requête.

L'avantage des tables est de pouvoir visualiser plusieurs données dans une seule fenêtre.

La gestion des tables est différente selon s'il s'agit d'une table mémoire ou d'une table reliée à un fichier ou une requête. Le choix du type de table est réalisé dès la création de la table, dans l'assistant.

Nous allons détailler les deux types de champs table.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampTable.WDW" sous l'éditeur. Cette fenêtre présente différents traitements sur les tables mémoires

En détail : mode de remplissage de la table

Comme une liste, une table peut être une table fichier (liée à un fichier de données ou à une requête) ou une table mémoire.

Remplissage d'une table mémoire

Une table mémoire peut être remplie par programmation grâce à la fonction **TableAjouteLigne**. Cette fonction admet en paramètre le nom de la table et le contenu de la ligne. Le contenu de la

ligne est donné colonne par colonne : chaque valeur de colonne est séparée par "une virgule".

```
//Ajout d'une ligne dans sa totalité
TableAjouteLigne(TABLE_Menu, "Lundi", "Œuf meurette", ...
    "Blanquette", "Glace vanille")
```

Table reliée à un fichier ou une requête

Lors de la création d'un champ table remplie à partir d'un fichier ou d'une requête, vous devez indiquer plusieurs informations :

- le fichier de l'analyse ou la requête qui servira de source de données pour le remplissage.



Notes

Si la requête n'existe pas lors de la création du champ, il est possible de créer la requête en même temps que le champ. La requête sera alors intégrée à la fenêtre qui contient le champ.

Attention : si vous utilisez une requête intégrée, cette requête sera utilisée uniquement pour ce champ. Elle ne pourra pas être ré-utilisée dans votre projet.

Ce fichier ou cette requête sera lu automatiquement sans avoir à écrire aucune ligne de code.

- les rubriques affichées dans une ligne de la table.
- la clé de parcours utilisée : cette rubrique sert de tri pour le remplissage de la table et donc indique l'ordre dans lequel seront visualisées les valeurs de la table. Par exemple, vous pourrez visualiser dans une table les produits et avec leur libellé.

Cas particulier : Table reliée à une requête paramétrée

Si une table est liée à une requête paramétrée, l'assistant de création de table propose par défaut de "Générer le code d'initialisation des paramètres de la requête". Cette option génère automatiquement le code d'initialisation de la table, avec des paramètres par défaut. Si vous avez spécifié des données de test, ces données sont automatiquement reprises (nous l'avons vu dans la partie 2).



Notes

Rafraîchir une table basée sur une requête

Pour rafraîchir l'affichage d'une table basée sur une requête (paramétrée ou non), utilisez la fonction **TableAffiche** avec la constante **taReExecuteRequête**.

Caractéristiques de la table

Lorsque la table est créée, ses caractéristiques sont modifiables directement dans la fenêtre de description du champ (option "Description" du menu contextuel).

Vous retrouverez toutes les caractéristiques indiquées dans l'assistant, et d'autres options très intéressantes.



Notes

La fenêtre de description d'une table est composée de deux zones :

- la zone supérieure présentant le nom de la table, des colonnes et leur type
- la zone inférieure composée des différents onglets de description.

Si le nom de la table est sélectionné, la partie inférieure présente les caractéristiques de la table.

Si une colonne est sélectionnée, la partie inférieure présente les caractéristiques des colonnes.

Quelques informations intéressantes :

- Il est possible d'effectuer des totaux automatiques dans les tables (onglet "Général" de la description de la table et onglet "Détail" des colonnes). Nous verrons cette fonctionnalité en détail dans la partie "Totaux dans les tables", page 219.
- Les tables peuvent être mono ou multi-sélection (onglet "Détail" de la description de la table).
- L'onglet "Style" permet de définir les couleurs des lignes paires et impaires, la couleur de la ligne sélectionnée (ou l'image de fond utilisée), ...

Que la table soit une table fichier ou une table mémoire, les colonnes peuvent être :

- déplaçables par l'utilisateur (grâce à la souris)
- avec recherche : une loupe sera affichée, permettant de rechercher dans la colonne une valeur.
- triables. Des flèches apparaîtront dans le titre des colonnes, signalant à l'utilisateur que la colonne peut être triée, ainsi que le sens de tri.



Notes

Il faut noter que la loupe ou la flèche de tri sont visibles sur toutes les colonnes de la table si celle-ci est remplie par programmation (table mémoire ou table basée sur une requête); par contre, la loupe et la flèche ne sont visibles que sur les colonnes liées à une clé (index) pour les tables reliées à un fichier.

En détail : Programmation d'une table mémoire

Dans une table mémoire, chaque ligne de la table est identifiée par un indice.

Pour manipuler une ligne comme pour manipuler une colonne, il faut préciser l'indice de la ligne concernée. Si l'indice n'est pas précisé, c'est la ligne sélectionnée qui est concernée.

- Une table mémoire peut être manipulée ligne par ligne ou colonne par colonne.
- Une table mémoire peut être en saisie ou en affichage.
- Une table mémoire peut être en mono-sélection ou en multi-sélections.

Fonctions permettant de manipuler une table mémoire

La programmation des champs Table est très simple avec WinDev : il suffit d'utiliser les fonctions WLangage commençant par les lettres TABLE. Ces fonctions peuvent être utilisées sur les tables mémoire et/ou fichier.



Notes

Vous ne savez pas si la fonction peut être utilisée sur une table mémoire ou fichier? Consultez l'aide en ligne! Utilisez la touche F1 sur le nom de la fonction.

Voici quelques-unes des fonctions les plus utiles pour manipuler une table mémoire :

- **TableModifieLigne** modifie une ligne de la table (soit la ligne en cours, soit la ligne dont le numéro d'indice est précisé).
- **TableSelect** retourne l'indice de la ligne en cours (sélectionnée) ou l'indice d'une des lignes sélectionnées dans le cas d'une table multi-sélections.
- **TableSelectPlus** permet de sélectionner par programmation une ligne de table en fonction de son indice et/ou de sélectionner plusieurs lignes.
- **TableSupprime** permet de supprimer une ligne de table.
- **TableSupprimeTout** vide toute la table mémoire.

États d'une table mémoire

Une table peut être :

- en affichage : aucune colonne n'est en saisie.
- en saisie : au moins une colonne est en saisie.

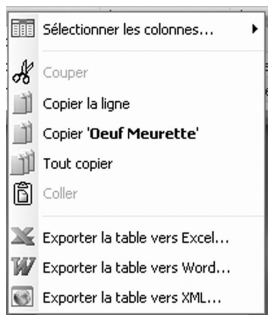
Ces options sont définissables au niveau de chaque colonne (onglet "Détail" de chaque colonne dans la fenêtre de description de la table).

L'état d'une table ou de chaque colonne de la table est défini sous l'éditeur, il peut être modifié par programmation avec la propriété **Etat**.

Astuces et FAA sur les tables

- ▶ Testez la fenêtre "FEN_ChampTable.wdw".

Et si vous effectuez un clic droit sur la table, le menu contextuel par défaut s'affiche :



Les options "Graphique" et "Calculs automatiques" sont disponibles uniquement si le clic droit a été effectué sur une colonne de type numérique

Voici quelques astuces pour optimiser le look de vos tables :

- **Pour utiliser un titre de colonne multi-lignes**, il suffit de cocher l'option "Titre de colonne multi-lignes" dans l'onglet "Détail" de la fenêtre de description du champ Table.
- **Pour utiliser une couleur de fond dégradée**, affichez l'onglet "Style" de la fenêtre de description du champ. Sélectionnez l'élément "Cadre extérieur", puis sélectionnez le cadre de type "Dégradé".

Pour plus de détails sur le fonctionnement d'une table fichier ou mémoire, consultez l'aide en ligne (mot-clé : "Table").

Type de champ : Zone répétée

Résumé

Les zones répétées permettent de répéter "n" fois un ensemble de champs. Lors de cette répétition, les champs de la zone répétée peuvent afficher des informations différentes. Il est ainsi possible d'afficher, à chaque répétition, des enregistrements d'une base de données.

Les zones répétées peuvent être de deux types :

- **Zone répétée mémoire** : la zone répétée est remplie par programmation.
- **Zone répétée fichier** : la zone répétée est remplie à partir d'un fichier de données ou d'une requête, automatiquement, sans une seule ligne de code.



Pour chaque champ de la zone répétée, plusieurs caractéristiques peuvent être modifiées lors de la répétition : valeur, libellé, couleur du texte, état, ...

La zone répétée peut être déplacée comme un champ. Lors de son déplacement, tous les champs associés à la zone répétée se déplacent en même temps.

Lors de la création d'une zone répétée, il est possible de définir le nombre de colonnes qui seront affichées. Cette option permet d'obtenir une répétition horizontale dans la limite du nombre de colonnes indiqué. Lorsque le nombre de colonnes est atteint, une nouvelle répétition verticale est ajoutée.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampZoneRépétée.wdw". Cette fenêtre utilise une zone répétée listant des collaborateurs. Les données de cette zone répétée proviennent d'un fichier de données Hyper File.

La gestion des zones répétées est différente s'il s'agit :

- d'une zone répétée mémoire,
- d'une zone répétée reliée à un fichier ou une requête.

Nous allons détailler les deux types de zones répétées existant.

En détail

Zone répétée mémoire

Une zone répétée peut contenir plusieurs champs (libellé, champ de saisie, image, ...). Chacun des champs de la zone répétée peut être associé à un attribut.

Cet attribut permet de modifier une propriété du champ associé (sa valeur, son libellé, sa couleur, sa visibilité, ...). Il est également possible de créer un attribut dans la zone répétée sans que celui-ci ne soit associé à un champ : on parle alors d'attribut libre (cet attribut peut contenir la valeur d'un compteur par exemple).

Le comportement d'une zone répétée mémoire est très proche d'un champ Table.

Dans une zone répétée mémoire, chaque ligne de la zone répétée est identifiée par un indice.

Pour ajouter une ligne dans une zone répétée, il faut utiliser la fonction **ZoneRépétéeAjouteLigne**.

Cette fonction attend en paramètre :

- le nom de la zone répétée,
- les valeurs des attributs (dans l'ordre) séparées par des virgules :

```
//Ajout d'une ligne dans la zone répétée
//La zone répétée contient 4 attributs, associés respectivement
//aux champs : LIBMAQUETTE, IMGMAQUETTE, QUANTITE, PRIXUNI
ZoneRépétéeAjouteLigne(ZR_Maquette, ...
    "Maquette n°1", "IMAGE_MAQUETTE1.JPG", 10, 45.65)
```

La fonction **ZoneRépétéeModifieLigne** modifie une ligne de la zone répétée.

Une ligne de zone répétée est supprimée avec la fonction **ZoneRépétéeSupprime**.

Zone répétée reliée à un fichier ou une requête

Lors de la création d'un champ zone répétée fichier vous devez indiquer le fichier de l'analyse ou la requête qui servira de source de données pour le remplissage.

Ce fichier ou cette requête sera lu automatiquement sans avoir à écrire aucune ligne de code.

La clé de parcours sélectionnée sert de tri pour le remplissage de la zone répétée et indique donc l'ordre dans lequel seront visualisées les valeurs.

Par exemple, vous pourrez visualiser dans une zone répétée :

- la photographie d'un produit
- la description du produit
- le prix du produit
- ...

Les caractéristiques d'une zone répétée sont disponibles dans la fenêtre de description de la zone répétée (menu contextuel, option "Description").

Pour plus de détails sur le fonctionnement d'une zone répétée fichier ou mémoire, consultez l'aide en ligne (mot-clé : "Zone répétée").

Type de champ : Arbre

Résumé

Aussi appelé champ "Liste arborescente" ou champ "Treeview", un champ Arbre est comparable à une liste dont le contenu est classé hiérarchiquement. Plus simplement, les données sont classées comme lorsque vous utilisez l'explorateur de fichiers de Windows.

En détail

- ▶ Ouvrez la fenêtre "FEN_ChampArbre.wdw".
- ▶ Lancez le test de cette fenêtre.
- ▶ Cliquez sur les différents boutons et observez ce qu'il se passe.
- ▶ Revenez sous l'éditeur et étudiez les traitements des boutons.

Le champ Arbre est géré par programmation. Les fonctions du WLangage qui permettent de programmer un arbre commencent par **ArbreXXX**.

Ces fonctions sont comparables aux fonctions de gestion des tables remplies par programmation. Par exemple, **ArbreAjoute**, **ArbreSupprimeTout**, **ArbreSupprime**, **ArbreSelect**, ...

Il est également possible de manipuler le champ Arbre ligne par ligne pour modifier les propriétés d'une ligne. Il est par exemple possible d'utiliser :

- la propriété **Couleur** pour modifier la couleur d'une ligne,
- la propriété **CouleurFond** pour modifier la couleur de fond d'une ligne,
- la propriété **Police** pour modifier les caractéristiques de la police d'une ligne,
- ...

Nous ne détaillerons pas les fonctions de manipulation des arbres. Pour plus de détails, consultez l'aide en ligne (mot-clé : "Arbre, Manipuler un arbre par programmation").

Type de champ : Table Hiérarchique**Résumé**

Une table hiérarchique est le rapprochement entre un champ Table et un champ Arbre.

Ce champ se base sur la structure d'un champ table (colonne, ligne, cellule), et possède une colonne de type Arbre, permettant d'introduire la notion d'arborescence, de hiérarchie. Cette notion est inexistante dans une table.

	Nom	Téléphone	Email
- B			
-	BORSANI Sarah	05-41-40-36-01	SarahBorsani@wanadoo.fr
-	BRODIER Céline	03-86-76-03-10	BrodierC@yahoo.fr
-	BUN Muy-Ly	04-06-94-64-15	BunMuyLy@laposte.net
- C			
-	CARLIN Paul	02-43-07-33-61	Carlin.Paul@laposte.net
-	CLARK Fabien	04-11-64-05-28	
- F			
-	FERA Lydia	01-02-56-51-64	Fera@wanadoo.fr
-	FILLOT Myriam	04-45-52-55-58	
- M			

Une table hiérarchique ne peut pas être basée sur un fichier de données.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampTableHiérarchique.WDW" sous l'éditeur. Cette fenêtre présente différents traitements sur les tables hiérarchiques.
- ▶ Testez cette fenêtre.

Type de champ : Jauge**Résumé**

Les jauges permettent de représenter la progression d'un traitement. La valeur de la jauge peut être modifiée par programme. La jauge peut être horizontale ou verticale. Les couleurs et le décor sont entièrement paramétrables.

Exemple pratique

- ▶ Ouvrez et testez la fenêtre "FEN_ChampJauge.wdw".

Quelques astuces de programmation

- ▶ Pour affecter le champ jauge avec une valeur, il suffit de faire une affectation, comme pour un champ numérique.

```
JAUGE_Copie = 75
```

- Pour définir les bornes minimales et maximales d'une jauge, utilisez les propriétés **BorneMin** et **BorneMax** :

```
JAUGE_Copie..BorneMin = 0
JAUGE_Copie..BorneMax = 100
```

Voici le code du bouton "Exécution" qui permet de faire évoluer les jauges.

```
I est un entier
POUR I = 1 A 100
    JAUGE_Horizontale1 = I
    JAUGE_Verticale = I
    JAUGE_Image = I
    Multitache(1)      // pour rafraîchir l'affichage
FIN
JAUGE_Horizontale1 = 0
JAUGE_Verticale = 0
JAUGE_Image = 0
```

Type de champ : Champ Graphe

Résumé

Le champ Graphe permet d'inclure simplement un graphique dans une fenêtre. La source de données de ce champ peut être :

- définie par programmation
- un fichier ou une requête
- une colonne de table
- un champ Liste
- un tableau WLangage

Il est possible de réaliser plusieurs types de graphes en 2 ou 3 dimensions :

- Secteurs,
- Histogramme,
- Boursier
- Courbe,
- Nuage de points.

Pour chacun des types de graphes, des options d'affichage permettent d'avoir des rendus différents.

Pour plus de détails, consultez l'aide en ligne (mot-clé : "Champ Graphe").

Exemple pratique

- Ouvrez la fenêtre "FEN_ChampGraphe". Cette fenêtre utilise un champ Graphe. Les données de ce graphe sont définies par programmation grâce à la fonction **grAjouteDonnée**. La fenêtre de description du champ Graphe permet de définir les principaux paramètres du champ.

Les champs spécialisés

Sous l'appellation "Champs spécialisés" nous avons regroupé tous les champs d'une utilisation rare ou spécifique. Ces champs sont les suivants :

- Carrousel
- Cube
- Tiroir
- Forme
- Code-barres
- Ascenseur
- Potentiomètre
- Spin
- WebCam
- Conférence
- HTML
- OLE
- Boite à outils
- Barre d'outils
- ActiveX
- Xaml

Nous allons étudier rapidement ces différents types de champs. Le projet "Fenêtres et champs" contient un exemple pour chacun de ces types de champs. Nous vous conseillons de tester la fenêtre associée au champ, d'étudier le code présent dans cette fenêtre et de consulter l'aide en ligne pour obtenir plus de détails sur l'utilisation de ces champs.

Type de champ : Carrousel

Résumé

Le champ Carrousel permet de créer des menus et des listes originales.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampCarrousel". Cette fenêtre utilise un champ Carrousel. Les différents boutons présentent quelques manipulations pouvant être effectuées sur un champ Carrousel.

Type de champ : Cube

Résumé

Le champ Cube est un champ de décor, mobile, qui effectue automatiquement des rotations sur tous les axes. Ce champ peut par exemple être présent en petite taille dans un coin d'une fenêtre.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampCube".
Attention : le champ Cube utilise le Framework DotNet 3.0. Le lancement de ce framework peut être très lent.
Remarque : Cette fenêtre ne peut pas être utilisée avec une version inférieure à Windows XP.

Type de champ : Tiroir**Résumé**

Le champ tiroir peut être déplié afin d'afficher ou non les champs qu'il contient. Ce type de champ est utilisé abondamment par Windows XP.

Exemple pratique

- ▶ Ouvrez la fenêtre "FEN_ChampTiroir".

Type de champ : Forme**Résumé**

Les formes permettent de dessiner des éléments aux formes géométriques (cercle, ellipse, rectangle, carré, ligne ou flèche). Un assistant de création de champs forme permet de définir toutes les caractéristiques d'un champ forme :

- couleurs.
- formes, dimensions, ...

Exemple pratique

- ▶ Pour obtenir un aperçu de ces champs, ouvrez et testez la fenêtre "FEN_ChampForme.wdw".

Type de champ : Champ Code-Barres**Résumé**

Le champ Code-Barres permet d'afficher simplement des codes-barres dans vos fenêtres. Il est possible par exemple de lire le code-barres d'un produit à l'aide d'un appareil spécialisé, d'afficher le code-barres à l'écran et d'enregistrer sa valeur dans un fichier de données.



Notes

Les codes-barres peuvent être imprimés soit depuis l'éditeur d'états, soit par programmation (fonction `ilprimeCodeBarre`).

Pour plus de détails, consultez l'aide en ligne.

Type de champ : Ascenseur

Résumé

Dans certains traitements particuliers, il peut être intéressant de créer des champs de type "Ascenseur".

Le principe de gestion des ascenseurs est le suivant :

- la valeur de la cage de l'ascenseur (le petit carré qui bouge !) est récupérée dans une variable.
- une valeur peut être affectée à cette cage.

Pour déplacer l'ascenseur, l'utilisateur peut cliquer sur un des boutons fléchés ou déplacer directement la cage elle-même. L'utilisateur peut même cliquer entre la cage et les flèches pour avoir une avance rapide.

En détail

- ▶ Ouvrez et testez la fenêtre "FEN_ChampAscenseur.wdw".

Les ascenseurs permettent de déplacer l'image dans la fenêtre.

Par défaut, l'ascenseur varie entre les valeurs 1 et 100. Chaque clic sur un des boutons fléchés déplace de 1 la cage et chaque clic entre la cage et le bouton déplace de 10.

Ces valeurs peuvent être modifiées dans la fenêtre de description de l'ascenseur ou en programmation grâce aux propriétés **BorneMin**, **BorneMax** et **AvanceRapide**.

Type de champ : Potentiomètre

Résumé

Un champ Potentiomètre correspond à un objet graphique circulaire ou linéaire permettant de faire varier une valeur. La valeur en cours du potentiomètre peut être récupérée par programmation dans une variable et être ainsi exploitée dans des calculs.

En détail

- ▶ Pour affecter le champ potentiomètre avec une valeur, il suffit de faire une affectation, comme pour un champ numérique.

```
POT_Son = 75
```

- ▶ Pour définir les bornes minimales et maximales d'un potentiomètre, utilisez les propriétés **BorneMin** et **BorneMax** :

```
POT_Son..BorneMin = 0
```

```
POT_Son..BorneMax = 100
```

Type de champ : Spin**Résumé**

Un champ Spin permet de gérer l'incrémentation et la décrémentation d'une valeur. Un Spin est constitué de 2 boutons collés.

Contrairement à un bouton, si l'on clique sur le champ Spin et que l'on garde le bouton de la souris enfoncé, l'événement de clic est répété automatiquement.

En détail

► Ouvrez et testez la fenêtre "FEN_ChampSpin.wdw".

Les spins permettent de :

- incrémenter / décrémentation une valeur.
- déplacer l'image dans la fenêtre.

Type de champ : Web caméra**Résumé**

Un champ Web caméra permet de visualiser une source vidéo provenant d'une caméra extérieure reliée à l'ordinateur.



EXEMPLE

Exemple

L'exemple "Webcam" (exemple complet) livré avec WinDev montre l'utilisation d'un champ Web caméra dans une fenêtre. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

En détail

Les fonctions permettant de gérer le champ Web caméra commencent par **VidéoXXX** :

```
// Sauvegarde de la vidéo retransmise dans le champ Web Caméra
// "CAM_WebCamera" sous forme d'un fichier AVI de 60 secondes
Res est un booléen = VidéoCapture(CAM_WebCamera, ...
    "C:\Temp\MaVideo.AVI", ...
    viCaptureVidéo, 60)

SI Res = Vrai ALORS
    Info("Vidéo correctement sauvegardée")
SINON
    Erreur("Impossible d'enregistrer la vidéo")
FIN
```



Notes

Seule la séquence vidéo actuellement retransmise peut être visualisée dans un champ Web caméra.

Pour visualiser :

- une image capturée avec la fonction **VidéoCapture**, utilisez un champ Image.
 - une séquence vidéo capturée avec la fonction **VidéoCapture**, utilisez un champ ActiveX permettant de lire une vidéo.
- Pour afficher en direct la vidéo en provenance de la Web caméra, utilisez la fonction **VidéoAffiche**.
- Pour comprendre le fonctionnement du champ Web caméra, vous allez simplement charger une fenêtre exemple et l'exécuter.
- Ouvrez et testez la fenêtre "FEN_ChampWebCaméra.WDW".
Attention : Une Web Caméra doit être accessible depuis le poste en cours.

Pour plus de détails sur les fonctionnalités des champs Web caméra et la syntaxe des fonctions évoquées ici, consultez l'aide en ligne (mot-clé : "WEB, Champ Web Caméra").

Type de champ : Conférence

Résumé

Un champ Conférence permet de visualiser et/ou d'écouter de la vidéo et/ou du son en temps réel. Il est ainsi possible de faire communiquer deux utilisateurs entre eux. L'image et la voix des utilisateurs sont retransmises en temps réel sur chaque poste connecté.



EXEMPLE

Exemple

L'exemple "VisioConférence" (exemple complet) livré avec WinDev montre l'utilisation d'un champ Conférence. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

En détail

Les fonctions permettant de gérer le champ Conférence commencent par **FluxXXX**.

Pour plus de détails sur les fonctionnalités des champs Conférence, consultez l'aide en ligne (mot-clé : "WEB, Champ Conférence").

Type de champ : HTML

Résumé

Un champ HTML permet d'intégrer une page HTML dans une fenêtre WinDev.

En détail

- Pour comprendre le fonctionnement du champ HTML, ouvrez et testez la fenêtre "FEN_ChampHTML.wdw".

Type de champ : OLE, ActiveX

Champ ActiveX

Résumé

Le champ ActiveX permet de gérer simplement les contrôles ActiveX. Les contrôles ActiveX sont des composants réutilisables utilisant la technologie ActiveX et pouvant être intégrés dans une fenêtre.

En détail

Un ActiveX est associé à des propriétés et des méthodes. Ces propriétés et méthodes peuvent être utilisées en WLangage pour manipuler directement le champ ActiveX. Il suffit d'utiliser la syntaxe Automation du WLangage.

Par exemple :

```
// AX_LecteurAVI est un champ ActiveX
AX_LecteurAVI>>Zoom = 12 // Affectation d'une propriété
AX_LecteurAVI>>Pause(10) // Appel de méthode
```

Pour connaître les propriétés et méthodes associées à un ActiveX, vous pouvez :

- soit consulter directement la documentation du contrôle ActiveX.
- soit utiliser l'utilitaire WDXVIEW livré avec WinDev.



EXEMPLE

Exemple

Les exemples "Agent", "Pilotage d'Excel", "Pilotage de Word", "Recherche sur Internet" (exemples didactiques et complets), livrés avec WinDev utilisent un champ ActiveX. Ces exemples sont accessibles depuis le volet "Assistants, Exemples et Composants" de WinDev.

Champ OLE

Résumé

Le Champ OLE permet de manipuler un objet OLE.

Rappel : OLE est un protocole d'échange de données et de commandes dont le principe consiste à imbriquer et lier des objets. Dès que l'objet incorporé est ouvert, la connexion client/serveur est ouverte.

Un **objet OLE** est un fichier qui contient :

- la référence du Serveur OLE qui a créé l'objet.
- l'objet lui-même (image, document feuille ...).

En détail

Plus techniquement, un objet OLE correspond à des données encapsulées contenant des données diverses telles que des textes, des images, des sons, ... L'encapsulation de données dans des objets OLE permet d'incorporer ces objets sans avoir besoin de connaître leur structure.

WinDev est un client OLE. Les applications développées avec WinDev demandent des données,

affichent et stockent les objets OLE.

Les serveurs OLE mettent à disposition des objets aux clients qui en font la demande. Ce sont par exemple Paint Brush, Excel, WinWord, ...

En tant que client, WinDev sait gérer des objets OLE. Il permet de :

- dessiner un objet dans un champ OLE. Un objet peut être par exemple une feuille Excel, une image de Paint Brush ...
- éditer un objet OLE dans un champ OLE. L'objet OLE peut être :
 - choisi sous l'éditeur ou par programmation.
 - défini en fonction d'un serveur ou d'un fichier.

Remarques :

- En programmation, un champ OLE est manipulé par des fonctions spécifiques. Une application WinDev **ne peut pas être** serveur OLE.
- OLE version 1 et OLE version 2 sont gérés (les versions supérieures ne sont pas gérées).
- Un objet OLE ne peut pas être imprimé avec WinDev. La mise à jour de l'objet s'effectue à la demande.
- Un champ OLE ne peut contenir qu'un seul objet OLE.

Nous n'allons pas détailler le fonctionnement de ces deux types de champ.

Pour plus de détails, consultez l'aide en ligne (mots-clés : "OLE", "Champ OLE", "ActiveX (champ)").

Type de champ : Boîte à outils

Résumé

Un champ Boîte à outils est constitué de plusieurs volets. Chaque volet peut contenir tous les types de champs (y compris une autre boîte à outils).

En détail

- ▶ Ouvrez et testez la fenêtre "FEN_ChampBAO.wdw".

Type de champ : Barre d'outils

Résumé

Le champ Barre d'outils est un type de champ avancé, lié à une fenêtre. Une barre d'outils contient plusieurs champs (bouton, champ de saisie, combo, ...) permettant à l'utilisateur de lancer rapidement certaines fonctionnalités de son application. Une barre d'outils peut être :

- **Ancrée** : la barre d'outils est "collée" à un des bords de la fenêtre.
- **Flottante** : la barre d'outils est une fenêtre indépendante.

L'utilisateur peut changer la position de la barre d'outils en utilisant la souris.

En détail

Pour associer des champs existants à la barre d'outils :

Méthode 1 : Réalisez un "Drag and Drop" du champ vers la barre d'outils.

Méthode 2 : Dans la description de la barre d'outils (onglet "Général"), cliquez sur le bouton "Ajouter" et sélectionnez le champ à ajouter à la barre d'outils. Appliquez les modifications, le champ sélectionné est automatiquement positionné dans la barre d'outils.

Pour associer des nouveaux champs à la barre d'outils :

Méthode 1 : Réalisez un "Drag and Drop" du champ vers le champ "Barre d'outils".

Méthode 2 : Dans la description de la barre d'outils (onglet "Général"), cliquez sur le bouton "Créer" et sélectionnez le type de champ à ajouter à la barre d'outils. Le champ sélectionné est automatiquement créé dans la barre d'outils. Pour le modifier, utilisez le bouton "Editer" : la description du nouveau champ apparaît.

- ▶ Ouvrez et testez la fenêtre "FEN_ChampBarreOutils". Cette fenêtre utilise deux barres d'outils : une barre d'outils ancrée et une barre d'outils flottante.

Remarque : L'espace entre les champs de la barre d'outils peut être configuré dans l'écran "Détail" de la description de la barre d'outils.

Type de champ : Xaml

Résumé

Le champ Xaml permet de gérer simplement le langage Xaml.

Pour utiliser ce champ, nous vous conseillons fortement de vous reporter à la documentation de Microsoft.

LEÇON 3.4. ERGONOMIE DE L'APPLI- CATION

Ce que vous allez apprendre dans cette leçon ...

- Améliorer l'ergonomie des champs et des fenêtres
- Améliorer l'ergonomie des applications



Durée estimée : 20 mn

Améliorer l'ergonomie des champs et des fenêtres

WinDev propose de nombreux outils pour améliorer l'ergonomie de vos fenêtres et de vos champs. Nous vous présentons ici quelques solutions simples pour améliorer efficacement l'utilisabilité de vos fenêtres.

Style / gabarit

Chaque projet WinDev est associé à une feuille de styles. La feuille de styles permet de définir les caractéristiques de chaque type de champ du projet.

Pour simplifier l'utilisation des styles, WinDev est livré avec de nombreux gabarits. Les gabarits utilisent une feuille de styles spécifique, contenant la description de l'interface de tous les types de champs pouvant être utilisés.



Notes

Lors de la modification du style d'un champ (onglet "Style"), WinDev propose de surcharger les propriétés du style en cours avec les options modifiées : c'est le mécanisme de surcharge des styles. Pour plus de détails, consultez l'aide en ligne.

Lors de la création du projet, il est possible de définir la charte graphique du projet. Cette charte graphique correspond au gabarit qui sera utilisé. A tout moment, il est possible de modifier la charte graphique utilisée :

- soit directement dans les options du projet (option "Projet .. Description du projet", onglet "Style").



Le changement est alors effectué pour toutes les fenêtres du projet.

- soit directement depuis l'éditeur de fenêtres (option "Fenêtres .. Appliquer un gabarit"). Le changement est alors effectué pour le champ en cours.



Notes

WinDev permet de créer vos propres gabarits. Vous pouvez ainsi définir entièrement la charte graphique de votre application. Pour plus de détails, consultez l'aide en ligne.

Redimensionnement des fenêtres

Les fenêtres des applications WinDev peuvent être redimensionnées. Dans ce cas, l'utilisateur pourra utiliser la poignée de redimensionnement pour modifier la taille de la fenêtre. La poignée de redimensionnement est placée en bas de la fenêtre.

Pour permettre le redimensionnement d'une fenêtre, il suffit de sélectionner l'option "Redimensionnable" dans l'onglet "IHM" de la description de la fenêtre.

Lors du redimensionnement de la fenêtre, les champs de la fenêtre doivent suivre le redimensionnement de la fenêtre. Cette fonctionnalité s'appelle l'ancrage des champs. C'est ce que nous allons étudier maintenant.

Il est conseillé de rendre les fenêtres redimensionnables uniquement si les fenêtres contiennent des champs qui peuvent s'agrandir : table, liste, image, boîte à outils, zone répétée, séparateur, onglet, arbre, liste image, HTML, ...

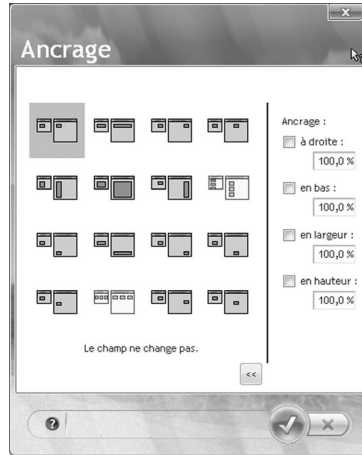
Ancrage

L'ancrage permet de lier la taille et/ou la position des champs à la taille de la fenêtre. Si l'utilisateur de l'application a la possibilité de modifier la taille (largeur et hauteur) de la fenêtre, vous pourrez lier certains champs de la fenêtre à ce redimensionnement.

Le champ pourra :

- se déplacer
- s'agrandir
- rester tel quel

L'ancrage peut être mis en place en faisant un clic droit sur un champ et en sélectionnant "Ancrage" dans le menu contextuel :



Notes

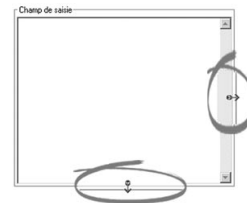
Il ne faut pas mettre systématiquement l'ancrage sur tous les champs d'une fenêtre. Il faut choisir judicieusement les champs et les fenêtres sur lesquels l'ancrage est utile.

Par exemple, dans une fenêtre contenant un champ Table, il peut être intéressant de lier le redimensionnement de la table lorsque la fenêtre s'agrandit.

Autre cas, une image contenant une photo pourra aussi être redimensionnée si la fenêtre est agrandie.

Par contre, on évitera de redimensionner des champs de saisie, les boutons, les sélecteurs ou les interrupteurs.

L'ancrage est symbolisé par des flèches rouges sur les côtés du champ :



Remarque : Si votre fenêtre est redimensionnable et si vos champs ne sont pas ancrés, une AAD apparaît automatiquement pour vous signaler le problème et vous aider à ancrer les champs.

Tabulation / ordre de saisie

L'ordre de saisie des champs est l'ordre dans lequel l'utilisateur pourra saisir les valeurs dans les différents champs de la fenêtre. Le passage d'un champ à un autre sera effectué lors de l'appui sur la touche [TAB] pendant l'exécution.

L'ordre de saisie par défaut correspond à l'ordre de création des champs. Il peut être modifié :

- soit en donnant un ordre de saisie automatique : le premier champ en saisie sera le champ situé le plus en haut à gauche, le deuxième sera celui qui est immédiatement situé à droite ou immédiatement en dessous ...
- soit en spécifiant un ordre de saisie par sélection.

- ▶ Pour définir un ordre de navigation automatique, sélectionnez l'option "Fenêtres .. Ordre de navigation .. Définir en automatique".
- ▶ Pour éditer l'ordre de saisie des champs, sélectionnez l'option "Fenêtres .. Ordre de navigation .. Editer" :



- ▶ Pour définir un ordre de saisie par sélection :
 1. Sélectionnez les champs dans l'ordre de saisie que vous souhaitez tout en maintenant la touche [CTRL] enfoncée.
 2. Sélectionnez l'option "Fenêtres .. Ordre de navigation .. Définir par la sélection".

Pour connaître l'ordre de saisie des champs de votre fenêtre, pas besoin de tester votre fenêtre : il suffit d'utiliser la touche [F5].

Règles / Correcteur d'interface / Alignement

Un aspect important de l'interface d'une application réside dans l'aspect de l'interface. Un point important est l'harmonie des différents champs, leur alignement dans la fenêtre.

Pour vous aider à réaliser des interfaces respectant les normes de programmation, WinDev met à votre disposition plusieurs outils : grille, règles, options d'alignement, positionnement automatique, ... Voici quelques pistes pour choisir l'outil qui vous convient le mieux.

Vous avez sans doute remarqué, lors de la création de nos fenêtres dans les leçons précédentes, des traits en pointillés apparaissant lors du déplacement de vos champs dans la fenêtre. C'est le **positionnement automatique**. Ces "règles" vous aident à positionner en temps réel vos champs. Vous pouvez tout de suite voir si le champ est aligné avec le champ précédent.

Pour paramétrer le positionnement automatique, il suffit de sélectionner l'option "Affichage .. Options .. Modifier les options" et de sélectionner l'onglet "Magnétisme".




Notes

Pour déplacer vos champs sans utiliser le magnétisme ou les règles, maintenez la touche [SHIFT] enfoncée pendant le déplacement.


Mais il se peut que vous vouliez aligner certains champs de votre fenêtre après les avoir créés (après leur déplacement par exemple). Vous pouvez utiliser les options d'alignement. Ces options sont toutes regroupées dans une barre d'outils. Pour l'afficher, sélectionnez l'option "Affichage .. Barre d'outils .. Alignement".



Pour utiliser les fonctionnalités de cette barre d'outils, il suffit de sélectionner plusieurs champs (à l'aide du lasso de la souris ou bien avec la touche CTRL) et de choisir un des alignements pré-définis. Le champ pris pour base pour effectuer l'alignement est le premier champ sélectionné.

Si aucun des alignements pré-défini ne vous convient, vous pouvez toujours effectuer un alignement entièrement personnalisé : l'icone  de la barre d'outils permet de paramétrer toutes les options.

Si vous souhaitez positionner vos champs au millimètre près, vous pouvez également utiliser les règles. Les règles sont affichables par les touches [CTRL] + [R].

Et enfin, si vos interfaces doivent respecter toutes les options définies pour les interfaces Windows (taille des champs, ...), vous pouvez utiliser le correcteur d'interface (icone  de la barre d'alignement). Ce correcteur étudie votre interface et vous propose toutes les modifications nécessaires. Son utilisation est avancée et est très rare.

Plan / Onglet


Votre fenêtre contient un nombre de champs important ? Pourquoi ne pas utiliser les plans ou les onglets pour regrouper les informations.

Les onglets

Les onglets permettent de regrouper les informations par thèmes. L'utilisateur peut accéder directement à un thème en cliquant sur "le volet" voulu.

Les onglets ressemblent à quelque chose de connu ? En effet, c'est le système utilisé dans toutes les fenêtres de description disponibles dans WinDev.

Le principe de fonctionnement des onglets est très simple. Il faut :

1. Créer le champ onglet (icone .

2. Créer autant de volets que nécessaire et donner un libellé à chaque volet (onglet "Général" de la description du champ).
3. Associer les champs sur les onglets voulus. Un champ peut être commun à tous les onglets ou associé à un onglet spécifique.

► Pour associer le champ à un onglet, deux solutions :

1. En déplaçant le champ :

- Cliquez sur l'onglet souhaité pour rendre un des volets actifs.
- Cliquez sur le champ à associer et déplacez-le vers le volet actif.
Le champ est alors directement associé à l'onglet et au volet d'onglet.

2. En utilisant le menu contextuel :

- Déplacez si nécessaire le champ à associer sur votre onglet.
- Cliquez avec le bouton droit de la souris sur le champ à associer et choisissez l'option "Associer à un onglet..." du menu contextuel.
- Sélectionnez l'onglet et le volet d'onglet souhaité et validez.
Le champ est automatiquement associé à l'onglet et au volet d'onglet spécifié.

Un champ ne peut être associé qu'à un seul volet d'onglet. Le champ sera actif et visible lorsque le volet d'onglet sera activé.

Votre onglet est prêt à fonctionner, aucune programmation spécifique n'est nécessaire.

Les plans

Les plans permettent également de regrouper les champs, tout en évitant la multiplication des fenêtres. Mais avec les plans, ce n'est plus l'utilisateur qui décide d'afficher les champs, c'est le développeur.

Les plans sont souvent utilisés dans les fenêtres de type "Assistant". Des boutons "Suivant" et "Précédent" permettent de passer d'un plan à un autre.



EXEMPLE

Exemple

L'exemple "Assistant" (exemple didactique) livré avec WinDev montre l'utilisation des plans dans une fenêtre. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

Les plans sont très faciles à manipuler sous l'éditeur de fenêtres de WinDev. En effet, lorsque vous souhaitez décrire un nouvel écran sans changer de fenêtre, utilisez la touche [PageSuivante] pour passer au plan suivant et la touche [PagePrécédente] pour passer au plan précédent. Par défaut, le premier plan d'une fenêtre est nommé "Plan 0". Chaque plan suivant est numéroté : "Plan 1", "Plan 2", ...



Notes

Pour simplifier la manipulation des plans sous l'éditeur de fenêtres, le numéro du plan en cours est affiché sous l'éditeur de fenêtres (dans la fenêtre d'accueil et dans la barre de message de l'éditeur).

Pour changer de plan par programmation, il suffit d'utiliser la propriété ..Plan. Par exemple :

```
// Code de clic du bouton PRECEDENT
MaFenêtre..Plan = MaFenêtre..Plan - 1
```

```
// Code de clic du bouton SUIVANT
MaFenêtre..Plan = MaFenêtre..Plan + 1
```



Notes

MaFenêtre est un mot-clé qui retourne le nom de la fenêtre qui contient le traitement où est exécuté **MaFenêtre**. Il peut être utilisé dans les traitements des champs de la fenêtre, les traitements de la fenêtre et également dans les procédures locales de la fenêtre.

Séparateur

Le séparateur est un champ permettant de découper une fenêtre en plusieurs zones. Ces zones sont redimensionnables par l'utilisateur final. Le découpage peut être horizontal, vertical ou une combinaison des deux.



Notes

La mise en place de séparateurs dans une fenêtre nécessite la mise en place de la gestion des ancrages : la dimension des champs doit s'adapter dans les différentes zones de la fenêtre.

Grâce aux séparateurs, l'utilisateur peut ainsi personnaliser son environnement de travail selon ses besoins.

Animation de fenêtres

Pour doter vos applications d'un côté ludique, ou étonner favorablement un utilisateur au lancement, les fenêtres peuvent être animées à leur ouverture et / ou à leur fermeture: une fenêtre peut s'ouvrir en tournant, en venant du fond de l'écran, ... (un peu comme des effets de montage vidéo). L'effet peut être fixé ou aléatoire.

Cette fonctionnalité peut être définie dans les options du projet, dans la description de la fenêtre ou par programmation.

Si une fenêtre charge de nombreux éléments, une "animation" (qui s'effectue indépendamment des traitements en cours) permet à l'utilisateur de patienter sans s'en rendre compte !

Cette fonctionnalité est à utiliser avec parcimonie, pour ne pas agacer l'utilisateur. Vous pouvez par exemple la réserver à la première fenêtre de l'application ou demander de n'afficher qu'une fenêtre sur 20 avec un effet visuel.

Il est possible d'animer :

- soit toutes les fenêtres de l'application (option "Projet .. Description du projet", onglet "Avancé", bouton "Animations de fenêtre"),
- soit uniquement la première fenêtre de l'application (option "Projet .. Description du projet", onglet "Avancé", bouton "Animations de fenêtre"),
- soit une ou plusieurs fenêtres données (onglet "Détail" de la description de chaque fenêtre).

Grisage automatique des fenêtres

Votre application ouvre plusieurs fenêtres simultanément et vos utilisateurs ne savent plus quelle fenêtre est celle à utiliser ? La ou les fenêtres inactives peuvent se griser automatiquement. Ainsi, vous devinez instinctivement la fenêtre à utiliser. Cette fonctionnalité s'appelle GFI pour Grisage des Fenêtres Inaccessibles.

Cette fonctionnalité est activable dans la description du projet (option "Projet .. Description du projet", onglet "Avancé").

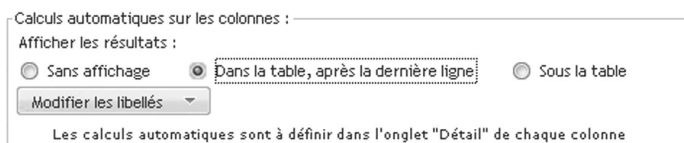
Totaux dans les tables

Vous utilisez des tables mémoire ou fichier dans vos applications ? Vous voulez afficher un total mais vous vous demandez comment faire ? Faut-il parcourir le fichier lié à la table ? Faire une requête ?

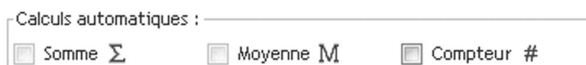
Il y a une solution beaucoup plus simple : utiliser les calculs automatiques des colonnes : somme, comptage ou moyenne.

Il suffit de :

1. Sélectionner la position d'affichage du calcul (onglet "Général" de la description de la table) :



2. Sélectionner la colonne sur laquelle le calcul portera.
3. Dans l'onglet "Détail" de cette colonne, cocher le calcul souhaité.



Notes

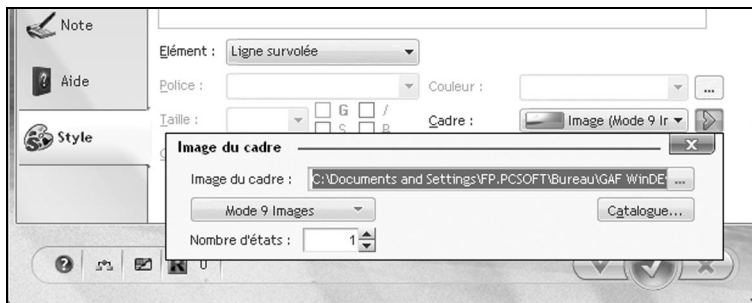
L'utilisateur a également la possibilité d'ajouter un calcul automatique sur la table grâce au menu contextuel automatique de la table.

Images des champs paramétrables dans le style

WinDev offre la possibilité de personnaliser l'interface de vos champs sans pour autant refaire entièrement la feuille de style ou le gabarit de l'application. Comment ? C'est très simple, il suffit de préciser les images à utiliser dans le style du champ.

Par exemple, dans l'onglet "Style" d'une table, il est possible de paramétrer l'image de fond de la

ligne sélectionnée :



Transparence des champs, transparence des fenêtres

Pour obtenir des effets sur vos champs et sur vos fenêtres, vous pouvez jouer sur la transparence. Il est par exemple possible d'utiliser la transparence pour afficher une fenêtre d'alerte ou d'avertissement "par dessus" une fenêtre sans perturber l'utilisation en cours.

La définition de la transparence se fait :

- sous l'éditeur pour les fenêtres (option "Opacité" de l'onglet "Style" de la description de la fenêtre) et pour les champs (option "Opacité" de l'onglet "Général" de la description du champ)
- par programmation pour les fenêtres et les champs grâce à la propriété **Opacité**.

Sauvegarde de la position dans les colonnes dans les tables

Une autre option qui sera vivement appréciée de vos utilisateurs, sans développement particulier : la mémorisation automatique de la configuration des colonnes de table. En effet, quel confort de retrouver à chaque lancement de l'application, la dernière configuration effectuée pour les colonnes : position, largeur des colonnes, ...

Aucune programmation n'est nécessaire : il suffit de cocher l'option "Mémoriser la config. des colonnes" dans l'onglet "Détail" de la fenêtre de description de la table.

Ergonomie de l'application

Nous venons de voir comment améliorer l'ergonomie de vos champs et de vos fenêtres. Mais il faut également tenir compte de l'ergonomie générale de l'application. En effet, des éléments comme le menu, le mode d'ouverture des fenêtres de l'application sont des éléments importants intervenant dans l'appréciation de l'utilisateur final dans l'application. Nous avons déjà vu la création d'un menu, nous allons voir en détail, les modes de gestion du multi-fenêtrage.

WinDev propose plusieurs modes d'utilisation du multi-fenêtrage :

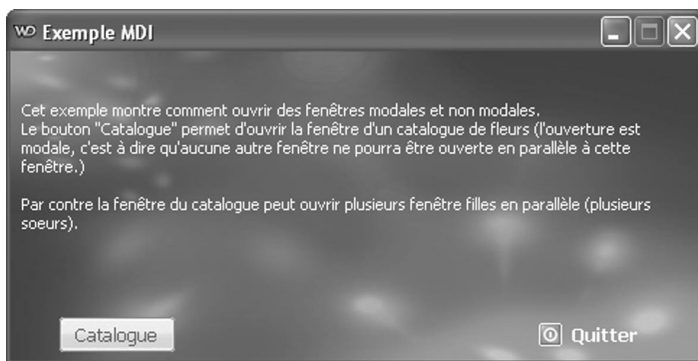
- utilisation d'une interface MDI
- multi-fenêtrage à partir de fenêtres libres.

Nous allons voir immédiatement un exemple de chacun de ces modes de gestion.

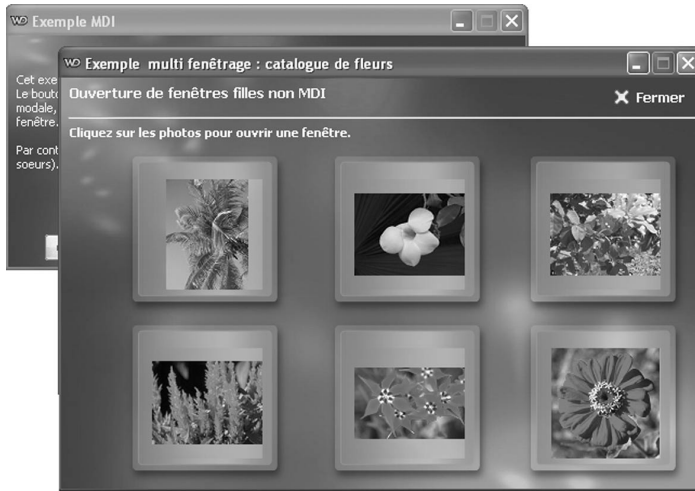
Exemple de programme gérant le multi fenêtrage de fenêtres libres (non MDI)

Pour illustrer les différents modes d'ouverture, nous allons travailler avec le projet "MULTIFEN".

- ▶ Testez cet exemple en sélectionnant l'option de menu "? .. Guide d'auto-formation .. Faire du Multifenêtrage".
Cet exemple est volontairement simple : il ne fait qu'ouvrir et fermer des fenêtres. Il ne gère pas de saisie ni de fenêtre MDI. Il a pour but d'illustrer les différents modes d'ouverture des fenêtres libres.
- ▶ La première fenêtre contient un bouton "Catalogue". Ce bouton ouvre une fenêtre en ouverture modale.



- ▶ Cliquez sur ce bouton.
Remarquez que vous ne pouvez pas cliquer en dehors de la fenêtre fille (le catalogue) et que la fenêtre mère d'introduction ne peut pas être en saisie tant que la fenêtre fille est ouverte.



- ▶ Cliquez sur une des images de fleur, la fiche de la fleur (fenêtre fille du catalogue) s'affiche. Remarquez que la fenêtre du catalogue (fenêtre mère) reste en saisie.



- ▶ Cliquez sur la fenêtre fiche pour qu'elle passe en saisie. Vous pouvez ouvrir en parallèle toutes les fiches en cliquant dans le catalogue sur chaque fleur.
- ▶ Depuis une fiche de fleur, vous pouvez également ouvrir une autre fiche (fenêtre sœur) en sélectionnant la fleur dans la combo et en cliquant sur le bouton "Voir".
- ▶ Si vous cliquez sur le bouton "Fermer" d'une fiche d'une fleur, seule la fiche se ferme. Par contre si vous cliquez sur le bouton "Fermer" du catalogue, toutes les fiches se ferment en même temps que la fenêtre du catalogue. En effet, la fermeture de la fenêtre mère ferme toutes ses fenêtres filles.

Détail des traitements

- ▶ Ouvrez le projet "MULTIFEN.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\Faire du Multifenêtrage" de WinDev.
- ▶ Ouvrez la fenêtre "FEN_Multi1.WDW". Les traitements associés à chaque champ image sont les suivants :

```
-- Initialisation de IMG_CLICAGE1
MoiMême = "Plantel.JPG"

-- Clic sur IMG_CLICAGE1
sNomImage est une chaîne
sNomImage = MoiMême
SI FenEtat("IMG_CLICAGE1") = Inexistant ALORS
    OuvreFille("IMG_CLICAGE1" + "=FEN_Multi2,0,0", sNomImage)
FIN
```

La fenêtre fiche de la fleur est affichée par la fonction **OuvreFille**. De cette façon, plusieurs fiches pourront être ouvertes en parallèle.

Pour que la fenêtre d'une fleur ne soit pas ouverte plusieurs fois, la fenêtre n'est ouverte que si elle est inexistante. La fonction **FenEtat** permet de vérifier si une fenêtre est déjà ouverte ou non.

La fenêtre fiche de la fleur qui a pour nom "FEN_Multi2" peut être ouverte plusieurs fois. Il faut alors pouvoir distinguer chaque fenêtre ouverte. Pour cela il suffit d'utiliser un alias.

Qu'est ce qu'un alias ?

Un alias permet de différencier la même fenêtre ouverte plusieurs fois.

L'alias est donné lors de l'ouverture de la fenêtre avec la fonction **OuvreFille** ou **OuvreSoeur**.

Dans la fenêtre fiche d'une fleur, le traitement du bouton permettant d'ouvrir une autre fiche est le suivant :

```
//ouvre la fenêtre des fleurs sélectionnées dans la combo
//après avoir vérifié qu'elle n'est pas déjà ouverte
//récupère la fleur sélectionnée
nIndice = ListeSelect(COMBO_Fleurs)
sNomImage = Majuscule(COMBO_Fleurs[COMBO_Fleurs]) + ".JPG"
                //fleur sélectionnée
SI FenEtat(sNomImage) = Inexistant ALORS
    OuvreSoeur(sNomImage+
        "=FEN_Multi2,"+(50*nIndice)+", "+(50*nIndice), sNomImage)
FIN
```

La fenêtre fiche de la fleur est affichée par la fonction **OuvreSoeur**. De cette façon, plusieurs fiches pourront être ouvertes en parallèle.

Exemple de programme gérant des fenêtres MDI

Pour illustrer la gestion des fenêtres MDI, nous allons travailler avec le projet "EXMDI".

- ▶ Ouvrez le projet "EXMDI.WDP" (présent dans le sous-répertoire "Autoformation\Corrigés\Faire du MDI" du répertoire d'installation de WinDev).

La première fenêtre est une mère MDI (elle a pour nom "FEN_MDICatalogue"). Elle contient une zone barre d'icônes.



La zone barre d'icônes contient :

- 7 boutons
- 1 champ de saisie
- 6 zones de clicage

Chaque zone de clicage permet d'ouvrir une fenêtre fille.

Par exemple, pour la première zone de clicage, on trouvera dans le code de clic :

```
sNomImage est une chaîne
sNomImage=MoiMême
SI FenEtat("IMG_CLICAGE1") = Inexistant ALORS
    MDIOuvre (FEN_MDIFleur, "IMG_CLICAGE1", "Plante1", sNomImage)
FIN
```


Les filles sont des filles MDI, elles s'affichent dans la fenêtre mère.



Les icônes permettent (de gauche à droite) :

- D'ouvrir toutes les filles (les filles sont alors empilées)
- De fermer toutes les filles ouvertes :

MDIMenuFenêtre (mdiFermeTout)

- D'afficher les filles ouvertes en mosaïque :

MDIMenuFenêtre (mdiMosaïque)

- D'afficher les filles ouvertes en cascade :

MDIMenuFenêtre (mdiCascade)

- De réduire en icône toutes les filles ouvertes :

MDIMenuFenêtre (mdiIconiseTout)

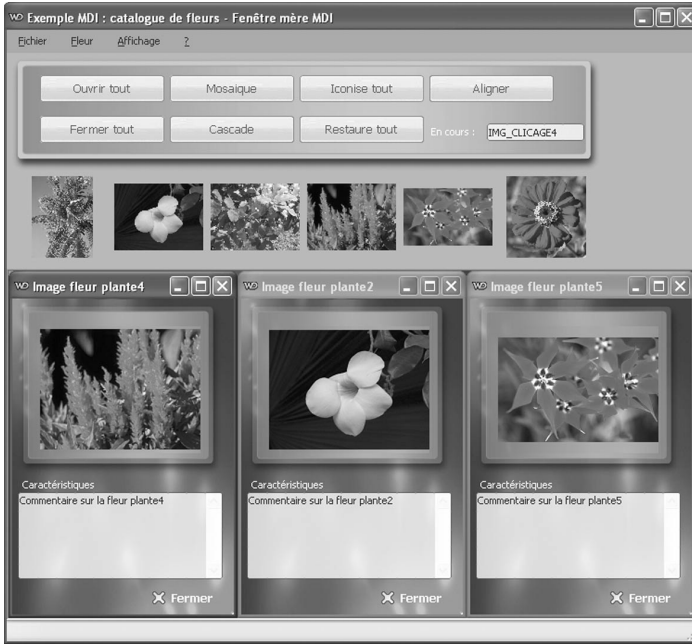
- De restaurer toutes les filles iconisées :

MDIMenuFenêtre (mdiRestaureTout)

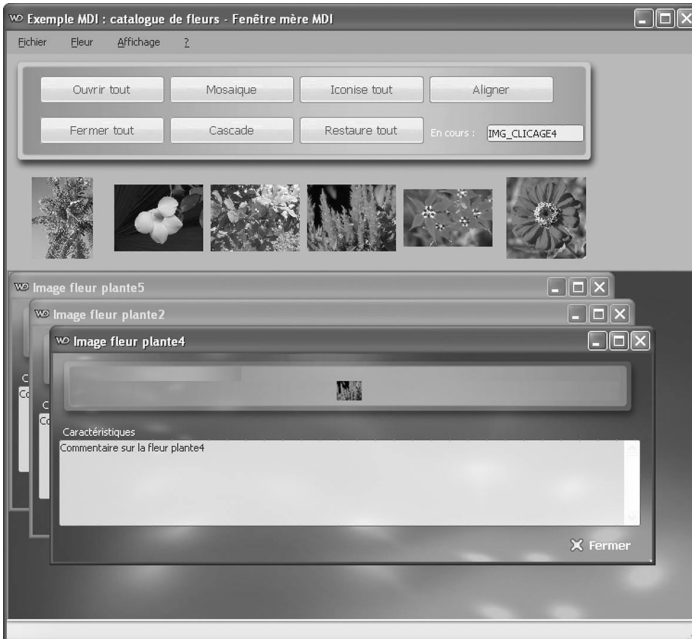
- D'aligner les icônes dans la fenêtre mère :

MDIMenuFenêtre (mdiArrange)

Affichage en mosaïque :



Affichage en cascade :



Le champ "en cours" indique quelle est la fille active. La fille active est la fille qui se trouve en premier plan, celle dont l'altitude est la plus élevée par rapport aux autres fenêtres.

La fonction **MDIActive** retourne le nom de la fenêtre fille en avant-plan.

Dans notre exemple, une procédure appelée par "timer" affiche le nom de la fenêtre fille active.

L'utilisation des timers est expliquée dans la leçon "Timer", page 388.

La fonction **MDIMère** permet de connaître le nom de la mère MDI.

LEÇON 3.5. RÉUTILISABILITÉ

Ce que vous allez apprendre dans cette leçon ...

- Réutiliser les éléments d'un projet



Durée estimée : 10 mn

Qu'est-ce que la réutilisabilité ?

Vous avez créé un sélecteur de répertoire que vous aimeriez réutiliser dans tous vos projets ? Vous souhaitez utiliser toujours les mêmes boutons "Fermer" placés au même endroit dans vos applications ?

WinDev vous offre différents moyens de gérer la réutilisabilité des fonctionnalités que vous créez.

Réutiliser un ensemble de champs

Vous voulez réutiliser un ensemble de champs ? WinDev met à votre disposition les outils suivants :

- Superchamps
- Modèle de champs
- Fenêtres internes


Nous allons voir ces trois outils en détail, avec un tableau récapitulatif pour chaque outil pour vous aider à choisir l'outil le mieux adapté à vos souhaits.

Le superchamp

Le superchamp est un type de champ avancé. Un superchamp regroupe un ensemble de champs dans un but spécifique (par exemple sélecteur de dates, sélecteur de fichiers, ...). Le superchamp contient :

- les champs nécessaires
- le code nécessaire à son fonctionnement.

Tableau récapitulatif

Création	<p>Création d'un superchamp :</p> <ol style="list-style-type: none"> 1. Créer le superchamp (icone . 2. Mettre les champs dans le superchamp. <p>Refactoring : Sélectionner les champs et faire "Champ .. Refactoring .. Créer un superchamp avec la sélection".</p>
Réutilisabilité	Via le dictionnaire
Mise à jour	Via le dictionnaire (grâce au système d'abonnement)
Travail en groupe	Partage du dictionnaire par réseau. Gestionnaire de sources non disponible.

Fenêtre interne

Le champ Fenêtre interne permet d'inclure une fenêtre (et son code) dans une autre fenêtre.

A l'exécution, la fenêtre à fusionner sera dynamiquement fusionnée à la fenêtre de réception.

Une fenêtre interne est une fenêtre spécifique (pas de barre de titre, pas de menu, ...). Dans cette fenêtre, vous pouvez mettre tout type de champs. Une fenêtre interne est un fichier d'extension "WDW".

Le champ "Fenêtre interne" permet de partager dynamiquement une même partie d'interface au sein d'une ou de plusieurs applications.

Tableau récapitulatif

Création	<p>Création d'une fenêtre interne :</p> <ol style="list-style-type: none"> 1. Créer la fenêtre interne (option "Fichier .. Nouveau .. Fenêtre .. Fenêtre interne") 2. Définir les champs et le code. 3. Enregistrer. <p>Refactoring : Sélectionner les champs et faire "Champ .. Refactoring .. Créer une fenêtre interne avec la sélection".</p>
Réutilisabilité	Via le champ "fenêtre interne".
Mise à jour	Via le dictionnaire (grâce au système d'abonnement) ou via le GDS.
Travail en groupe	Partage par le réseau ou par le GDS
Avantages	Possibilité de modifier dynamiquement (par programmation) la fenêtre interne utilisée dans le champ Fenêtre interne.
Inconvénients	<ul style="list-style-type: none"> - Pas améliorable dans l'interface d'accueil : pas de surcharge, pas de déplacement de champs. - Zone rectangulaire.

Exemple d'utilisation : fenêtre dont l'interface ne va pas changer quel que soit le projet : Fenêtre d'options.

Modèle de champs

Un modèle de champs est un ensemble de champs, réutilisables dans plusieurs fenêtres. Un modèle de champs est une fenêtre spécifique contenant différents champs. Dans cette fenêtre, vous pouvez mettre tout type de champs. Un modèle de champs est un fichier d'extension "WDT".

Tableau récapitulatif

Création	<p>Création d'un modèle de champs :</p> <ol style="list-style-type: none"> 1. Créer le modèle de champ (option "Fichier .. Nouveau .. Modèle de champs") 2. Définir les champs et le code. 3. Enregistrer. <p>Refactoring : Sélectionner les champs et faire "Champ .. Refactoring .. Créer un modèle de champs avec la sélection".</p>
Réutilisabilité	Via le champ "Modèle de champs".
Mise à jour	Via le dictionnaire (grâce au système d'abonnement) ou via le GDS.
Travail en groupe	Partage par le réseau ou par le GDS
Avantages	Les modèles de champs peuvent être surchargés : du code peut être ajouté, les champs peuvent être déplacés dans la fenêtre utilisant le modèle de champs. Les champs peuvent être modifiés.

Dans la majorité des cas, il est conseillé d'utiliser un modèle de champs.

LEÇON 3.6. QUESTIONS / RÉPONSES

Ce que vous allez apprendre dans cette leçon ...

- Astuces sur la manipulation de champs



Durée estimée : 10 mn

Question Comment dupliquer un champ dans une fenêtre par programmation ?

Le "clonage humain" n'est pas encore en vogue, mais avec WinDev, il est déjà possible de cloner les champs. La fonction **ChampClone** permet de dupliquer par programmation un champ dans une fenêtre ou un état.

Le champ est dupliqué avec les mêmes caractéristiques mais avec un nom différent.

Question Comment supprimer un champ dans une fenêtre par programmation ?

La fonction **ChampSupprime** permet de supprimer par programmation un champ dans une fenêtre ou un état.

Question Comment gérer les plans d'une fenêtre ?

Les plans d'une fenêtre permettent de répartir des champs dans différentes "couches" afin d'éviter les écrans trop chargés ou la multiplication des fenêtres dans un projet.

Pour associer un champ à un plan :

1. Faites un clic droit sur le champ.
2. Sélectionnez l'option "Associer à un plan".
3. Choisissez le numéro du plan auquel le champ doit être associé.

Pour passer d'un plan à l'autre dans l'éditeur, utilisez les touches [Page Suivante] et [Page Précédente] du clavier. Le numéro du plan en cours est affiché :

- dans la barre de message de l'éditeur (en bas à gauche).
- dans la fenêtre d'accueil de la fenêtre en cours (en haut à droite).

**Astuce**

Pour éviter de dupliquer un même champ dans une fenêtre, vous pouvez associer le champ à "aucun plan". Le champ est alors visible dans tous les plans.

Seuls les champs du plan en cours et les champs n'appartenant à aucun plan sont visibles en édition et en exécution.

Vous pouvez également utiliser la propriété **Plan** pour :

- connaître et changer le plan actif d'une fenêtre.
- connaître et changer le plan associé à un champ.

Le volet "Mode Plan" (option "Affichage .. Barres d'outils .. Mode plan") permet d'éditer l'ensemble des libellés des champs de tous les plans d'une fenêtre pour les visualiser et éventuellement les modifier simultanément.

Question Comment gérer le clic sur un volet d'un onglet ?

Un onglet est une sorte de bouton constitué de plusieurs zones clicables nommées "Volets". Pour gérer et déterminer sur quel volet un clic est réalisé, utilisez la syntaxe suivante dans le code de clic de l'onglet :

```

SELON NomChampOnglet
  CAS 1 // premier volet
  //...Traitement à effectuer...
  CAS 2 // deuxième volet
  //...Traitement à effectuer...
  AUTRES CAS
  //...Autres traitements à effectuer...
FIN

```

Question Comment afficher la progression d'un traitement ?

Pour afficher la progression d'un traitement, on utilise le plus souvent une jauge. Pour cela, utilisez un champ Jauge dans une fenêtre (option "Insertion .. Champ .. Jauge").

Dans le code d'initialisation du champ Jauge :

1. Initialisez la valeur minimale de la jauge :

```
NomJauge..BorneMin = ValeurMini
```

2. Initialisez la valeur maximale de la jauge :

```
NomJauge..BorneMax = ValeurMaxi
```

Dans le code du traitement voulu, incrémentez la jauge à chaque étape du traitement :

```

NomJauge ++
// ou NomJauge = NomJauge + 1

```

Question Comment afficher un site internet dans une fenêtre ?

Avec WinDev, vous pouvez facilement insérer une liaison Internet dans vos applications. Pour cela :

1. Insérez un champ HTML dans une fenêtre (option "Insertion .. Champ .. Champ HTML").
2. Initialisez le champ HTML avec l'adresse du site auquel vous souhaitez vous connecter :

```
NomChampHTML = "http://www.pcsoft.fr"
```

Bien évidemment, votre poste (ainsi que le poste de l'utilisateur final) doit avoir une connexion à Internet active.

Question Comment modifier la couleur d'un champ ?

La couleur d'un champ se définit dans le style du champ (option "Champ.. Choisir un style"). Cependant, il est possible de modifier la couleur de ce champ par programmation. La syntaxe est la suivante :

```
// Colorier le texte en rouge
NomChamp..Couleur = RougePastel

// Colorier la couleur de fond du libellé en vert
NomChamp..CouleurFond = VertClair

// Remettre la couleur d'origine (celle du style)
NomChamp..Couleur = iCouleurDefaut
```



Notes

Cette syntaxe s'applique à tous les types de champs.

La fonction **RVB** permet de définir une couleur à partir des différentes valeurs des composantes Rouge, Vert et Bleu.

```
<CodeCouleur> = RVB (<rouge>, <vert>, <bleu>)
```

Il est également possible de modifier les couleurs des lignes, des colonnes ou d'une cellule d'un champ Table. La syntaxe est la suivante :

```
// Modification de la couleur d'une colonne
NomColonne..Couleur = <CodeCouleur>

// Modification de la couleur d'une ligne
NomTable[IndexeLigne]..Couleur = <CodeCouleur>

// Modification de la couleur d'une cellule
NomColonne[IndexeLigne]..Couleur = <CodeCouleur>
// ou
// NomTable[IndexeLigne, IndexeColonne]..Couleur = <CodeCouleur>
```

Question Comment rendre un bouton invisible ?

Un bouton peut être rendu invisible par programmation avec la syntaxe suivante :

```
NomBouton.Visible = Faux
```

Donnez la valeur "Vrai" pour rendre le champ à nouveau visible.

Cette syntaxe peut également être appliquée sur tous les types de champs et sur les groupes de champs.

Question Comment créer le menu principal de mon application ?

Effectuez les opérations suivantes :

1. Créez une nouvelle fenêtre ou ouvrez une fenêtre existante.
2. Sélectionnez l'option "Fenêtres .. Menu principal .. Ajouter le menu principal".
Une option est créée automatiquement.
3. Faites un clic droit sur cette option.
4. Cliquez sur "Description de l'option" pour modifier l'option sélectionnée.
5. Cliquez sur "Ajouter après" pour ajouter une nouvelle option après.
6. Cliquez sur "Ajouter avant" pour insérer une nouvelle option avant.
7. Cliquez sur "Insérer un sous-menu" pour ajouter un sous-menu dans l'arborescence du menu.

Question Comment créer un menu contextuel ?

Un menu contextuel peut être ajouté :

- soit au niveau de la fenêtre.
- soit au niveau d'un champ.

Pour une fenêtre :

1. Faites un clic droit sur la fenêtre et sélectionnez l'option "Description".
2. Cliquez sur l'onglet "IHM" et cliquez sur l'icone à côté de la combo "Menu contextuel".

Pour un champ :

1. Faites un clic droit sur le champ et sélectionnez l'option "Description".
2. Cliquez sur l'onglet "IHM" et cliquez sur l'icone à côté de la combo "Menu contextuel".

Pour connaître ou modifier le menu contextuel d'un champ ou d'une fenêtre par programmation, utilisez la propriété **MenuContextuel**.

Question Comment modifier le curseur de survol pour un champ ou une fenêtre ?

Un curseur de survol peut être défini :

- soit au niveau de la fenêtre.
- soit au niveau d'un champ.

Pour définir le curseur de survol

1. Faites un clic droit sur l'élément (fenêtre ou champ) et sélectionnez l'option "Description".
2. Cliquez sur l'onglet "IHM" et sélectionnez le curseur de survol voulu dans la combo "Curseur".

Vous pouvez choisir parmi des curseurs prédéfinis ou sélectionner un curseur que vous avez créé.
Pour modifier le curseur de survol par programmation, utilisez la propriété **CurseurSouris**.

Question Comment passer des paramètres à une fenêtre ?

La méthode pour passer des paramètres à une fenêtre est similaire au passage des paramètres à une procédure.

Dans le code de déclaration des globales de la fenêtre, saisissez la syntaxe du code WLangage suivante :

```
PROCEDURE NomFenêtre(pNomParam1, pNomParam2, ...)
```

Lors de l'ouverture de la fenêtre avec la fonction **Ouvre**, passez les paramètres après le nom de la fenêtre, par exemple :

```
Ouvre(Nomfenêtre, ValeurParam1, ValeurParam2, ...)
```

Si vous initialisez un paramètre lors de la déclaration dans la fenêtre, ce paramètre devient optionnel :

```
// pNomParam2 est un paramètre optionnel
PROCEDURE NomFenêtre(pNomParam1, pNomParam2 = "Test")
```



Notes

Il est préférable de passer des paramètres à une fenêtre plutôt que de déclarer des variables globales dans le projet.

Question Comment regrouper des champs pour modifier leurs propriétés par programmation ?

Effectuez la manipulation suivante :

1. Sélectionnez plusieurs champs avec la souris.
2. Dans le menu de l'éditeur, sélectionnez l'option "Champ .. Groupes .. Associer la sélection".
3. Vous pouvez :
 - Associer la sélection à un groupe existant.
 - Associer la sélection à un nouveau groupe (bouton "Nouveau"). Il suffit d'indiquer le nom du groupe et de valider.

Les groupes de champs sont utilisables aussi bien dans les fenêtres que dans les états.

Les champs seront associés à ce groupe. Vous pourrez ensuite modifier les propriétés des champs de ce groupe par la syntaxe :

```
NomGroupe.<NomPropriété> = Valeur
```



Attention !

Seules les propriétés communes à tous les champs sont modifiables.

Question Comment transformer un champ interrupteur en un champ sélecteur ?

L'option de menu "Champ .. Permuter .. Sélecteur/Interrupteur" inverse les deux types de champs.



Notes

Cette opération peut également être effectuée entre une combo et une combo avec table ou entre un champ Potentiomètre et un Potentiomètre rotatif.

Question Comment mettre des boutons à la même taille ?

- ▶ Sélectionnez en premier le bouton servant de référence pour la taille (largeur et hauteur), puis les autres boutons à redimensionner.
- ▶ Sélectionnez les options d'alignement "Même largeur, Même hauteur" dans les outils d'alignement (option "Champ .. Alignement").

Question Comment gérer la persistance des champs ?

- ▶ Pour gérer la persistance des champs, sélectionnez l'option "Mémoriser la valeur" dans l'onglet "Détail" de la fenêtre de description du champ.

Remarque : l'utilisateur peut également mémoriser la valeur d'un champ grâce au menu contextuel du champ.

Question Comment ajouter une image en fond dans une fenêtre ?

Sur la fenêtre :

1. Faites un clic droit et sélectionnez l'option "Description".
2. Sélectionnez l'onglet "Image". Vous pouvez choisir une image et son mode d'affichage.

Question Comment harmoniser le look de mes fenêtres ?

Pour harmoniser le look des fenêtres d'une application, il suffit de créer un modèle de fenêtres (option "Fichier .. Enregistrer comme un modèle") et d'utiliser ce modèle pour toutes les fenêtres de votre application.

Pour plus de détails, consultez l'aide ligne (mot-clé : "Modèle, Modèle de fenêtres").

Question Comment gérer la transparence d'une fenêtre ?

Sur la fenêtre :

1. Faites un clic droit et sélectionnez l'option "Description".
2. Sélectionnez l'onglet "Style".
3. Définissez l'opacité de la fenêtre grâce au potentiomètre.

Il est également possible de définir l'opacité d'une fenêtre ou d'un champ grâce à la propriété WLangage **Opacité**.



DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 4

**Bases de données
et analyses**

XIII

Express

LEÇON 4.1. INTRODUCTION

Ce que vous allez apprendre dans cette leçon ...

- Vocabulaire utilisé.
- Les différents modes d'accès aux bases de données.



Durée estimée : 20mn

Présentation

Lors de la conception d'une application, vous pouvez être amené à manipuler des données. Pour les stocker, vous devez constituer ce que l'on nomme "une base de données".

Dans WinDev, lors de la création d'un projet manipulant des données, vous devez tout d'abord créer une "analyse".

Une "analyse" contient la description des fichiers (ou tables) contenant les données de l'application.

C'est seulement lors de l'exécution de l'application, que ces descriptions sont utilisées pour créer la base de données et/ou les fichiers de données. C'est dans cette base ou dans ces fichiers que seront stockées les données.



Notes

Plusieurs outils de maintenance des bases de données Hyper File sont livrés en standard avec WinDev. Ils sont accessibles depuis le Centre de Contrôle Hyper File.

WinDev sait gérer différents formats de base de données (pour ne pas dire tous). Les plus courantes sont :

- Hyper File, système de base de données intégrée à WinDev et livrée en standard. La base de données Hyper File est disponible en mode Classic ou Client/Serveur.
- AS/400, Access, Sybase, Informix ...
- Oracle, SQL Server, MySQL, xBase, ...
- Toute base accessible en langage SQL sous Windows.
- Texte (fichiers ASCII).

Pour accéder aux données, il existe différentes techniques (appelées "modes d'accès") :

- Accès Natif
- Accès OLE DB
- Accès ODBC direct
- Accès ODBC via OLE DB

Les différents modes d'accès aux bases de données

Accès Natif

Un accès natif manipule directement et exclusivement un format de base de données. Ce type d'accès optimisé est développé spécialement pour chaque format de base de données.

Dans WinDev, il existe un accès natif pour les bases de type :

- Hyper File Classic ou Client/Serveur (en standard)
- xBase (en standard)
- Access (en standard)
- XML (en standard)
- Oracle (optionnel)
- AS/400 (optionnel)
- SQL Server (optionnel)

- Sybase (optionnel)
- Informix (optionnel)
- DB2 (optionnel)
- Progress (optionnel)
- MySQL (optionnel et gratuit)

D'autres accès natifs seront bientôt disponibles, contactez notre service commercial !
Les fonctions WLanguage **SQL*** et **HLit*** sont utilisables avec ce type d'accès. Le code est ainsi portable et indépendant de la base de données.

Accès ODBC direct

Un accès via ODBC direct utilise un standard d'accès multi-bases. Vous devez installer la couche ODBC 32 bits sur votre machine. Cette couche est en général déjà installée sur les versions récentes de Windows. Vous pouvez le vérifier dans le panneau de configuration de Windows en choisissant l'option "Administrateur ODBC".

Attention : toutes les bases de données ne sont pas obligatoirement accessibles via cette méthode. Si vous souhaitez utiliser ce type d'accès, vérifiez qu'il existe un pilote ODBC.

Seules les fonctions WLanguage **SQL*** sont utilisables avec ce type d'accès.

Accès OLE DB

Un accès via OLE DB est un accès qui utilise un standard d'accès multi-bases. Ce type d'accès est basé sur le MDAC (Microsoft Data Access Component) de Microsoft.



Attention !

Si vous utilisez un accès OLE DB, vous devez obligatoirement installer le MDAC sur les postes utilisateurs (version 2.6 minimum).

Toutes les bases de données ne sont pas obligatoirement accessibles via cette méthode. Si vous souhaitez utiliser ce type d'accès, vérifiez qu'il existe un pilote OLE DB.

Les fonctions WLanguage **SQL*** et **HLit*** sont utilisables avec ce type d'accès.

Accès ODBC via OLE DB

En résumé, il s'agit d'un "mélange" de OLE DB et de ODBC. Cette technique est la plus "lourde" et la moins efficace en terme de performances. Il est déconseillé de l'utiliser sur des bases de petite taille.

Les fonctions WLanguage **SQL*** et **HLit*** sont utilisables avec ce type d'accès.

Quel accès utiliser ?

- Pour obtenir plus d'informations sur les différents modes d'accès aux bases de données et savoir quel type d'accès utiliser, sélectionnez l'option "? .. Aide à l'utilisation des bases de données" dans le menu de WinDev.

LEÇON 4.2. PARCOURS DE FICHIERS ET DE REQUÊTES

Ce que vous allez apprendre dans cette leçon ...

- Parcours de type HLitPremier
- Parcours de type HLitRecherche
- Parcours de type POUR TOUT
- Parcours de type POUR TOUT AVEC



Durée estimée : 20mn

Parcours de type **HLitPremier**

Le parcours de type **HLitPremier** permet de parcourir un fichier de données selon une rubrique clé (ou index) de parcours. Dans ce cas, la fonction **HLitPremier** est combinée à la fonction **HLitSuivant** qui permet, comme son nom l'indique de lire l'enregistrement suivant.

Par exemple, le code suivant permet de parcourir le fichier Client selon la rubrique clé "Nom", du premier enregistrement au dernier.

```
HLitPremier(Client, Nom)
TANTQUE PAS HEnDehors(Client)
    // Traitement de l'enregistrement
    HLitSuivant(Client, Nom)
FIN
```

Dans ce code, la boucle de parcours teste la valeur de la fonction **HEnDehors**. La fonction **HEnDehors** permet de savoir si la fin du fichier est atteinte ou non.

Lors du parcours d'une requête, la fonction **HLitPremier** ré-exécute automatiquement la requête.



Remarque

Il est possible de réaliser de la même façon un parcours du fichier du dernier enregistrement au premier. Il suffit d'utiliser les fonctions **HLitDernier** et **HLitPrécédent**.

Parcours de type **HLitRecherche**

Le parcours de type **HLitRecherche** permet de se positionner dans le fichier de données pour réaliser un parcours des enregistrements correspondant à une condition. Cette condition porte sur la clé de parcours (par exemple, toutes les commandes d'un client). La fonction **HLitSuivant** permet de lire l'enregistrement suivant correspondant à la condition.

Par exemple, le code suivant permet de rechercher les clients dont le nom commence par "Dupont" :

```
HLitRecherche(Client, Nom, "Dupont")
TANTQUE HTrouve(Client)
    // Traitement de l'enregistrement
    HLitSuivant(Client, Nom)
FIN
```

Dans ce code, la boucle de parcours teste la valeur de la fonction **HTrouve**. La fonction **HTrouve** permet de savoir si un enregistrement correspondant à la condition a été

trouvé.



Remarque

Par défaut, la fonction **HLitRecherche** effectue une recherche générique : tous les enregistrements commençant par la valeur indiquée sont recherchés (dans notre exemple, Dupont et Dupontel seront trouvés).

Pour faire une recherche à l'identique :

- utilisez la fonction **HLitRecherchePremier**.
- utilisez la constante *hIdentique* avec la fonction **HLitRecherche**.

Parcours de type POUR TOUT

Le parcours de type POUR TOUT permet d'effectuer un parcours complet d'un fichier, d'une requête ou d'une vue Hyper File.

Le parcours peut être effectué :

- sur la meilleure clé du fichier, détectée automatiquement par le moteur Hyper File.
- sur une clé spécifique.

Par exemple, le code suivant permet de parcourir le fichier Client selon la rubrique "Nom", du premier enregistrement au dernier.

```
POUR TOUT Client SUR Nom
    // Traitement des enregistrements
FIN
```

Ce code est équivalent au code présenté pour le parcours de type **HLitPremier** :

```
HLitPremier(Client, Nom)
TANTQUE PAS HEnDehors(Client)
    // Traitement de l'enregistrement
    HLitSuivant(Client, Nom)
FIN
```

Il est possible d'effectuer le parcours du dernier enregistrement au premier, en précisant le sens de parcours :

```
POUR TOUT Client SUR Nom DepuisFin
    // Traitement des enregistrements
FIN
```


Parcours de type POUR TOUT AVEC

Le parcours de type POUR TOUT AVEC permet de parcourir un fichier Hyper File selon une condition.

Le parcours peut être effectué :

- sur la meilleure clé du fichier, détectée automatiquement par le moteur Hyper File.
- sur une clé spécifique.

Par exemple, le code suivant permet de parcourir le fichier Client selon le numéro de client et en donnant une condition sur le nom du client.

```
// Parcours avec filtre
POUR TOUT Client AVEC "Nom = 'Dupont'"
    // Ajout des clients dans la liste
    ListeAjoute(ListeClient, Client.NumClient)
FIN
```

Ce code est équivalent au code présenté pour le parcours de type **HLitRecherche** :

```
HLitRecherche(Client, Nom, "Dupont")
TANTQUE HTrouve(Client)
    // Traitement de l'enregistrement
    HLitSuivant(Client, Nom)
FIN
```

Quel type de parcours choisir ?

Tous les types de parcours présentés se valent. Le choix de l'un ou l'autre va dépendre des clés présentes dans le fichier, de la taille du fichier, du type de recherche effectuée (recherche de type "Contient" ou non).

LEÇON 4.3. GESTION DES ERREURS

Ce que vous allez apprendre dans cette leçon ...

- La gestion automatique des erreurs
- La gestion programmée des erreurs



Durée estimée : 20mn

Présentation

Lors de la gestion d'une base de données, plusieurs types d'erreurs peuvent apparaître lors de l'ajout ou de la modification d'un enregistrement :

- Erreur de mot de passe sur un fichier
- Erreur de saisie obligatoire
- Erreur de doublons
- Erreur d'intégrité, ...

WinDev propose plusieurs modes de gestion de ces erreurs :

- **le mode automatique** : pour chaque erreur rencontrée lors de la gestion des enregistrements de la base de données, une fenêtre spécifique est affichée à l'utilisateur. Cette fenêtre permet à l'utilisateur de modifier directement ses données.
- **le mode programmé manuellement** : pour chaque erreur rencontrée lors de la gestion des enregistrements de la base de données, le programmeur traite directement l'erreur dans son programme.
- **le mode programmé avancé** : pour chaque erreur rencontrée lors de la gestion des enregistrements de la base de données, une procédure ou une fenêtre personnalisée de gestion des erreurs est appelée.

Nous allons détailler ces trois modes de gestion des erreurs à l'aide d'un exemple simple. Cet exemple est constitué de deux fichiers (Produit et TVA) nous permettant de tester simplement les erreurs de doublons et les erreurs d'intégrité.

- ▶ Ouvrez le projet "ErreursHF.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\ErreursHF" de WinDev.

Gestion des erreurs en mode automatique

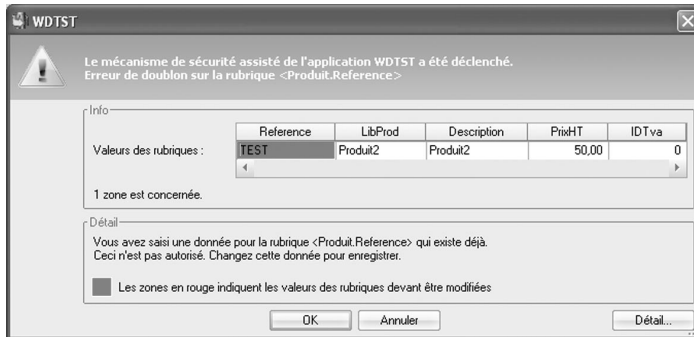
Par défaut, le mode automatique est activé. Dans ce mode, WinDev propose des fenêtres standard pour chaque type d'erreur. Ces fenêtres indiquent à l'utilisateur le problème.

Aucun code spécifique n'est nécessaire dans l'application.

Testons immédiatement le type d'erreur pouvant être affiché dans notre exemple simple.

- ▶ Exécutez le projet "ErreursHF" grâce au bouton "GO".
- ▶ Cliquez sur le bouton "Gestion des erreurs de doublons et d'intégrité en mode automatique".
- ▶ Nous allons tout d'abord tester les erreurs de doublons. Pour cela, saisissez les informations suivantes :
 - Référence : TEST
 - Libellé : Produit1
 - Description : Produit1
 - Prix : 50
 - Taux de TVA : 5,5%

- Cliquez sur le bouton "OK". L'écran suivant s'affiche :



Cette fenêtre explique clairement le problème : Vous avez saisi une donnée pour la rubrique "Référence" qui existe déjà. Une erreur de doublon apparaît. Cette fenêtre permet à l'utilisateur de corriger directement la valeur.

Le bouton "Valider" permet de valider la saisie dans la fenêtre d'erreur.

Le bouton "Annuler" permet de revenir à l'écran ayant provoqué l'erreur.

Cliquez sur le bouton "Annuler".

- Testons maintenant les erreurs d'intégrité. Pour cela, saisissez les informations suivantes :
 - Référence : TEST2
 - Libellé : Produit2
 - Description : Produit2
 - Prix : 50
 - Taux de TVA : laissez le champ vide.
- Cliquez sur le bouton "OK". L'écran suivant s'affiche :



Cette fenêtre indique qu'une erreur d'intégrité est rencontrée (dans notre cas, la saisie du taux de TVA n'a pas été réalisée).

Le bouton "Modifier" permet de revenir à l'écran en cours de saisie.

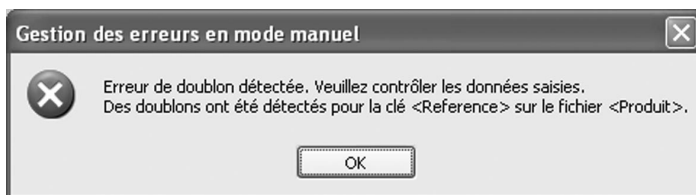
Dans le mode automatique, dès qu'une erreur de gestion de base de données est rencontrée, les fenêtres d'assistance s'affichent. Il peut être intéressant dans certains cas de gérer plus ou moins manuellement ces erreurs. Dans ce cas, le programmeur peut intervenir de manière plus précise sur les erreurs rencontrées.

- Cliquez deux fois sur le bouton "Annuler". Le menu de l'exemple est affiché.

Gestion des erreurs en mode programmé manuellement

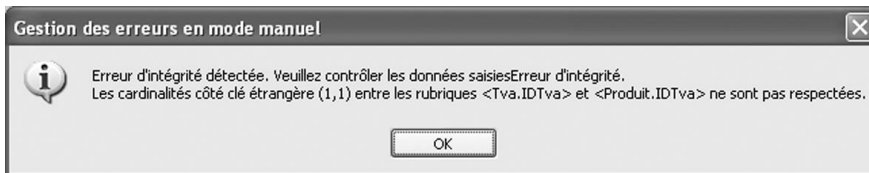
Nous allons exécuter tout de suite l'exemple en mode programmation manuelle puis examiner le code correspondant.

- ▶ Cliquez sur le bouton "Gestion des erreurs de doublons et d'intégrité en mode programmé manuellement".
- ▶ Nous allons tout d'abord tester les erreurs de doublons. Pour cela, saisissez les informations suivantes :
 - Référence : TEST
 - Libellé : Produit1
 - Description : Produit1
 - Prix : 50
 - Taux de TVA : 5,5%
- ▶ Cliquez sur le bouton "OK". L'écran suivant s'affiche :



Cette fenêtre affiche la raison du problème.

- ▶ Testons maintenant les erreurs d'intégrité. Pour cela, saisissez les informations suivantes :
 - Référence : TEST2
 - Libellé : Produit2
 - Description : Produit2
 - Prix : 50
 - Taux de TVA : laissez le champ vide.
- ▶ Cliquez sur le bouton "OK". L'écran suivant s'affiche :



- ▶ Cliquez sur le bouton "OK". Fermez l'application.
- ▶ Regardons maintenant le code qui a permis d'afficher ces messages :
 1. Sous l'éditeur, ouvrez la fenêtre "FEN_Erreurs_Manuel" (double-cliquez sur le nom de la fenêtre dans le volet "Projet").
 2. Affichez les traitements de la fenêtre (clic droit sur la fenêtre, option "Code"). Le code d'initialisation de la fenêtre contient la ligne de code suivante :

```
HSurErreur ("*", hErrTout, " ")
```

La fonction **HSurErreur** permet de modifier le mode de gestion des erreurs de la base de don-

nées. Les paramètres de la fonction permettent de spécifier l'étendue de la modification :

- "*" pour indiquer que tous les fichiers de l'analyse sont concernés.
- la constante **hErrTout** pour indiquer que tous les types d'erreurs sont concernés. Il est possible d'utiliser des constantes spécifiques pour chaque type d'erreur à prendre en compte.
- "" (chaîne vide) pour indiquer qu'aucune fenêtre spécifique ou procédure spécifique ne doit être lancé : le traitement de l'erreur sera géré par le programmeur.

3. Fermez cette fenêtre de code.

4. Affichez le code du bouton "OK" (Sélectionnez le bouton et utilisez la touche [F2] par exemple).

5. Le code utilisé est le suivant :

```
EcranVersFichier()
HAjoute(Produit)
SI ErreurDéteectée ALORS
    SI HErreurDoublon() ALORS
        Erreur("Erreur de doublon détectée."+...
            "Veuillez contrôler les données saisies." + RC + ...
            HErreurInfo())
        RepriseSaisie(SAI_Referance)
    FIN
    SI HErreurIntégrité() ALORS
        Info("Erreur d'intégrité détectée."+...
            "Veuillez contrôler les données saisies"+ RC+ ...
            HErreurInfo())
        RepriseSaisie(SAI_Referance)
    FIN
FIN
```

Quelle est la signification de ce code ?

- Les fonctions **EcranVersFichier** et **HAjoute** permettent d'ajouter les données saisies à l'écran dans le fichier **Produit**.
- La variable **ErreurDéteectée** permet de savoir si une erreur est survenue lors de l'exécution de la dernière instruction (dans notre cas, la fonction **HAjoute**). Si **ErreurDéteectée** est à **Vrai**, une erreur est survenue.

Astuce : Il est également possible de tester directement le résultat renvoyé par la fonction **HAjoute**, en utilisant le code :

```
EcranVersFichier()
SI HAjoute(Produit) = Faux ALORS
```

Si une erreur s'est produite lors de l'ajout de l'enregistrement, l'enregistrement n'est pas ajouté et il est nécessaire de traiter l'erreur afin de pouvoir continuer l'exécution de l'application. C'est ce que nous allons faire dans les lignes de code suivantes.

- La fonction **HErreurDoublon** permet de tester si une erreur de doublons est survenue : **HErreurDoublon** renvoie alors **Vrai**. Que faire dans ce cas ? Dans notre code, nous affichons un message d'erreur, en précisant le message exact retourné par le moteur Hyper File (grâce à la fonction **HErreurInfo**). Ensuite, la fonction **RepriseSaisie** permet de revenir sur l'écran de saisie.
- La fonction **HErreurIntégrité** permet de tester si une erreur d'intégrité est survenue. **HErreurIntégrité** renvoie alors **Vrai**. La suite du traitement est identique à la gestion des doublons.

Ce code nous a permis de gérer correctement les erreurs d'intégrité et / ou de doublons rencontrés

lors de l'ajout de l'enregistrement.

Mais le mieux serait d'intervenir également en amont, afin de détecter l'erreur effectuée par l'utilisateur et ainsi lui permettre de corriger sa saisie avant de la valider.

Ainsi, pour empêcher une erreur de doublons, il suffit de rechercher si un enregistrement identique existe déjà dans la base. Une seule instruction suffit : **HRecherche** :

```
// Contrôle pour les doublons
// On recherche si le produit existe déjà
SI HRecherche (Produit, Reference, SAI_Reference) = Vrai ALORS
    Erreur ("Cette référence existe déjà" + RC + ...
           "Merci de modifier cette valeur")
    RepriseSaisie (SAI_Reference)
FIN
```

Pour empêcher l'erreur d'intégrité, il suffit de vérifier que tous les éléments nécessaires ont été saisis. Dans notre exemple, l'erreur d'intégrité apparaît car le taux de TVA n'est pas saisi. Un simple contrôle de la valeur permet d'éviter l'apparition de l'erreur d'intégrité :

```
// Contrôle pour l'intégrité
// Vérification qu'une valeur a été saisie dans la combo
SI TauxTVA.ValeurAffichée="" ALORS
    Erreur ("Veuillez saisir un taux de TVA")
    RepriseSaisie (TauxTVA)
FIN
```

- ▶ Pour tester cette solution, lancez l'exécution de l'application :
 1. Cliquez sur le bouton "Gestion des erreurs de doublons et d'intégrité en mode programmé manuellement".
 2. Effectuez les tests de doublons et d'intégrité en validant avec les boutons "OK avec contrôles".
 3. Fermez l'application.

Gestion des erreurs en mode programmé avancé

La gestion des erreurs en mode programmé avancé consiste à personnaliser complètement le traitement de l'erreur par programmation. WinDev offre la possibilité d'utiliser :

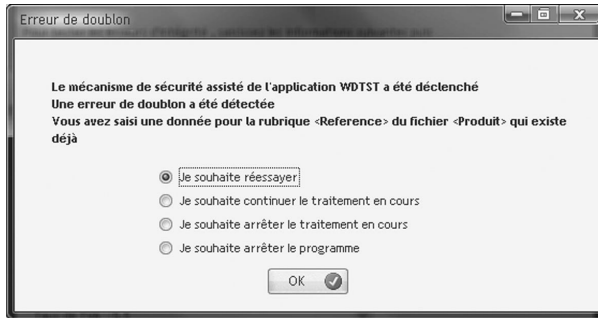
- soit une fenêtre de gestion des erreurs complètement personnalisée.
- soit une procédure spécifique.

Dans notre exemple, nous avons choisi d'utiliser une fenêtre spécifique.

- ▶ Exécutez le projet "ErreursHF" grâce au bouton "GO".
- ▶ Cliquez sur le bouton "Gestion des erreurs de doublons et d'intégrité en mode programmé avancé".
- ▶ Nous allons tout d'abord tester les erreurs de doublons. Pour cela, saisissez les informations suivantes :
 - Référence : TEST
 - Libellé : Produit1
 - Description : Produit1

- Prix : 50
- Taux de TVA : 5,5%

▶ Cliquez sur le bouton "OK". L'écran suivant s'affiche :

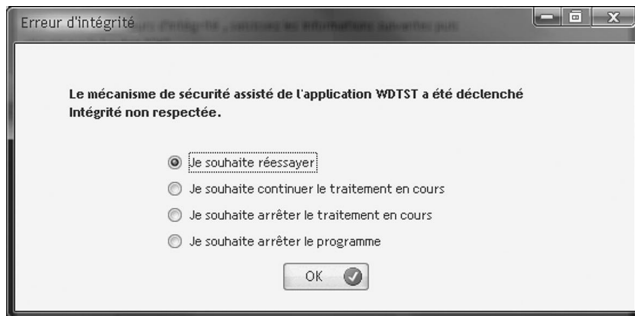


Cette fenêtre affiche la raison du problème avec une interface personnalisée. Sélectionnez l'option "Je souhaite continuer le traitement en cours" et validez.

▶ Testons maintenant les erreurs d'intégrité. Pour cela, saisissez les informations suivantes :

- Référence : TEST2
- Libellé : Produit2
- Description : Produit2
- Prix : 50
- Taux de TVA : laissez le champ vide.

▶ Cliquez sur le bouton "OK". L'écran suivant s'affiche :



Sélectionnez l'option "Je souhaite continuer le traitement en cours" et validez.

▶ Regardons maintenant le code qui a permis d'afficher ces messages :

1. Sous l'éditeur, ouvrez la fenêtre "FEN_Erreurs_Programme" (double-cliquez sur le nom de la fenêtre dans le volet "Projet").
2. Affichez les traitements de la fenêtre (clic droit sur la fenêtre, option "Code"). Le code d'initialisation de la fenêtre contient la ligne de code suivante :

```
HSurErreur ("*", hErrDoublon, FEN_Doublon2)
HSurErreur ("*", hErrIntégrité, FEN_Integrite2)
```

C'est encore la fonction **HSurErreur** qui permet de paramétrer le mode de gestion des erreurs de la base de données. Les paramètres de la fonction permettent de spécifier l'étendue de la modi-

fication :

- "*" pour indiquer que tous les fichiers de l'analyse sont concernés.
- la constante **hErrDoublon** pour indiquer que seules les erreurs de doublons sont concernées. Il est nécessaire de spécifier ensuite le nom de la fenêtre à utiliser dans ce cas (FEN_Doublon2).
- la constante **hErrIntégrité** pour indiquer que seules les erreurs de d'intégrité sont concernées. Il est nécessaire de spécifier ensuite le nom de la fenêtre à utiliser dans ce cas (FEN_Integrite2).

3. Fermez cette fenêtre de code.

Nous ne détaillerons pas ici le code des fenêtres utilisées. Nous vous conseillons de consulter l'aide en ligne pour plus de détails ainsi que l'exemple "WD Detection Erreurs" fourni en standard avec WinDev.

LEÇON 4.4. LA JOURNALISATION

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce que la journalisation ?
- Utilisation de la journalisation.



Durée estimée : 15mn

Qu'est-ce que la journalisation ?

Cette leçon concerne uniquement la journalisation de base de données Hyper File. Pour les bases de données externes (Oracle, ...) consultez la documentation de ces bases de données.

La journalisation pour quoi faire ?

Le journal est un fichier particulier qui vous permettra de répondre aux besoins suivants :

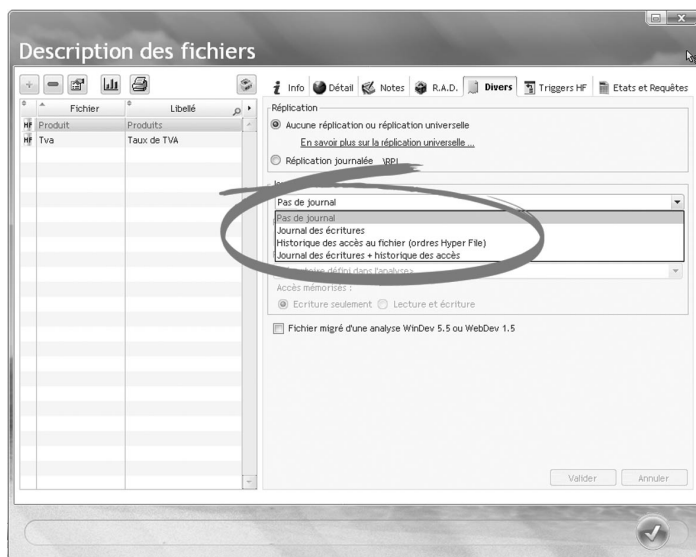
- remettre la base de données dans l'état où elle était tel jour à telle heure suite à un problème physique, des erreurs de traitements ou de saisie, ...
- avoir un suivi sur les opérations effectuées sur un fichier "sensible" (qui a modifié ou consulté le fichier ? par exemple).
- gérer la réplication (voir "La réplication de données", page 271)

Le journal est une sécurité pour votre base de données. Il vous permettra de remettre rapidement votre base de données dans le dernier état connu avant le problème rencontré, depuis la dernière sauvegarde, sans avoir à tout ressaisir.

Comment mettre en place la journalisation

La mise en place de la journalisation est réellement simple avec WinDev. Pour chaque fichier voulu, il suffit d'indiquer le type de journalisation à appliquer (onglet "Divers" de la fenêtre de description du fichier).

Par exemple ici sur le fichier "PRODUIT" :



Il suffit de choisir le type de journalisation souhaitée :

- Si votre objectif est de pouvoir rapidement restaurer votre base de données suite à un problème physique ou une erreur de traitement il suffit de choisir "Journal des écritures".

- Si vous souhaitez juste conserver une trace des accès à la base vous choisissez "Historique des accès au fichier".



Attention !

La journalisation de type "Historique des accès au fichier" ne permet pas de restaurer votre base de données en cas de problème. Cette option sert à effectuer des contrôles sur les accès aux fichiers "sensibles" ou tout simplement pour des besoins statistiques.

Ce type de journalisation est néanmoins déconseillé.

- Si vous souhaitez combiner les deux options, choisissez "Journal des écritures + Historique des accès au fichier".

Physiquement, comment se matérialise la journalisation ?

Tous les accès à tous les fichiers journalisés d'une analyse donnée sont archivés dans un fichier nommé "Table des opérations" (1 par analyse).

Ce fichier unique garantit la séquentialité des opérations.

Toutes les modifications sur un fichier de données sont stockées dans le fichier de journal correspondant (1 par fichier journalisé). Par exemple pour un fichier de données "PRODUIT", le fichier journal sera "PRODUITJNL".

Ces fichiers "journaux" sont situés par défaut dans le répertoire défini lors de la description des fichiers de l'analyse :

- Si vous n'avez rien spécifié, ils se trouveront dans le sous-répertoire "JNL" du répertoire des fichiers de données.
- Si vous avez spécifié un dossier, ils se trouveront dans le répertoire que vous aurez indiqué.

Le nom des répertoires est stocké dans le fichier lui-même.

Il est possible de modifier les répertoires des journaux après la mise en place de la journalisation :

- avec WDJournal, livré avec WinDev (option "Information sur un fichier")
- par programmation. Vous pouvez modifier cet emplacement en utilisant les fonctions **HChangeRepJNL**, **HChangeRepRPL**, ... (consultez l'aide en ligne pour plus de détails, mot-clé : "Journalisation").



Attention !

La modification des répertoires des journaux est une opération "critique".

Vous devez absolument suivre le mode opératoire ci-dessous :

1. Interdire les accès à la base (tous les fichiers).
2. Déplacer les fichiers journaux et/ou tables des opérations dans les nouveaux répertoires.
3. Modifier le nom des répertoires dans TOUS les fichiers de données concernés (option "Divers", dans l'écran de description des fichiers).
4. Ré-autoriser les accès à la base de données.

Comment réussir la mise en place de la journalisation ?

La mise en place de journaux pour une base de données est liée à la gestion des sauvegardes de cette même base (sauvegardes des fichiers de données).

En effet, un journal ne peut être "joué" que sur un état donné de la base. L'état qui correspond à la création ou re-cr ation du journal.

Vous pouvez r aliser ces diff erentes op erations sans programmation en utilisant WJournal, livr  avec WinDev.



Notes

WJournal est redistribuable avec vos applications.

Nous vous conseillons  galement de suivre les modes op eratoires suivants :

- ▶ Mise en place de la journalisation :
 1. Sauvegarde des fichiers de donn es initiaux (s'ils existent).
 2. Mise en place de la journalisation.

- ▶ Sauvegarde d'une application avec des fichiers journal s :
 1. Sauvegarde des fichiers de donn es.
 2. Sauvegarde  ventuelle des fichiers journaux.
 3. R initialisation des fichiers journaux.

- ▶ Restauration suite   un probl me :
 1. Restaurer la derni re sauvegarde.
 2. Rejouer le journal en cours jusqu'  l'op ration souhait e.
 3. Refaire une sauvegarde et r initialiser les journaux.

Pour plus de d tails sur la journalisation, consultez l'aide en ligne (mot-cl  : "Journalisation").



Notes

Journalisation et Hyper File Client / Serveur

La journalisation est disponible en Hyper File Client / Serveur. Pour plus de d tails sur les sp cificit s de la journalisation en mode Client / Serveur, consultez l'aide en ligne.

LEÇON 4.5. LES TRANSACTIONS

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce qu'une transaction ?
- Comment utiliser les transactions ?



Durée estimée : 15 mn

Qu'est-ce qu'une transaction ?

Une transaction est un ensemble d'opérations sur fichiers indissociables :

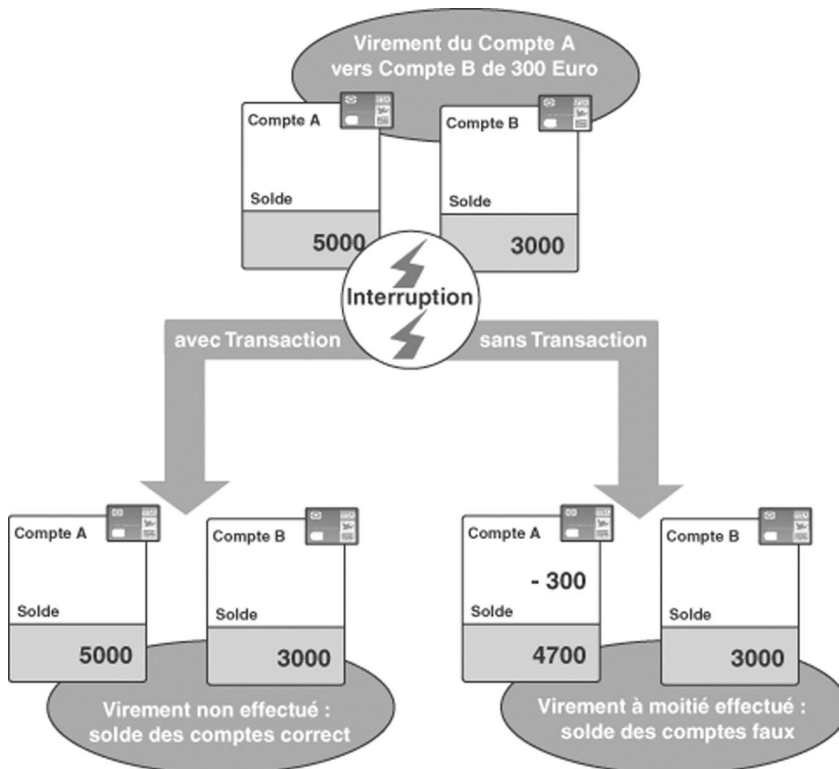
- soit toutes les opérations de la transaction sont réalisées.
- soit aucune opération n'est réalisée.

La gestion des transactions est le meilleur moyen d'assurer la cohérence d'un ensemble d'écritures indissociables sur des fichiers Hyper File.

Les transactions sont également un moyen sûr et fiable de sécuriser vos traitements sur des fichiers Hyper File.

Un exemple simple de la nécessité des transactions pour certains types de traitements

Lors d'une opération bancaire de virement, un compte est débité pour en créditer un autre. Pendant les opérations qui sont effectuées, il peut survenir à tout moment une panne sur le réseau (panne électrique, défaillance des systèmes informatiques). C'est pourquoi il est alors indispensable de gérer ce traitement par une transaction. Si une panne survient, toutes les opérations à effectuer sont alors annulées. On ne se retrouve donc pas avec un compte débité et un compte non crédité !



Les transactions sur Hyper File

Chaque opération d'écriture effectuée lors d'une transaction est mémorisée dans un fichier spécial. A tout moment, il est possible d'annuler la transaction : toutes les opérations effectuées depuis le début de la transaction seront annulées.

Les transactions seront annulées dans les cas suivants :

- erreur du programme.
- fin de programme.
- panne de courant ou arrêt brusque de l'application.

Lorsque l'application sera relancée, la cohérence de la base de données pourra être rétablie grâce :

- à l'outil WinDev "WDTRANS".
- au premier appel de la fonction **HTransactionDébut**.
- au premier appel de la fonction **HTransactionAnnule**.

Lorsque les opérations d'écriture incluses dans la transaction sont terminées, le programme peut valider les opérations de la transaction.

Utilisation des transactions

Mettre en place la gestion des transactions

1. Si vos fichiers sont protégés par mot de passe, ouvrez tous les fichiers utilisés pendant la transaction (fonction **HOuvre**) avant le début de la transaction ou spécifiez les mots de passe avec **HPasse**.

Si vos fichiers ne sont pas protégés par mot de passe, les fichiers manipulés après la fonction **HTransactionDébut** seront automatiquement mis en transaction.

2. Commencez la transaction avec la fonction **HTransactionDébut**. Cette fonction permet éventuellement de définir le nom du journal des transactions.

3. Effectuez vos opérations. Toutes les opérations d'écriture sur les fichiers en transaction sont automatiquement enregistrées dans le fichier de transaction.



Attention !

Les traitements réalisés sont relativement plus lents (puisque chaque opération est également enregistrée dans un fichier spécifique).

4. Annulez si nécessaire les opérations réalisées pendant la transaction (fonction **HTransactionAnnule**).

5. Spécifiez la fin de la transaction avec la fonction **HTransactionFin** : la transaction est validée.

Pour plus de détails sur les fonctions de gestion des transactions, consultez l'aide en ligne (mot-clé : "Transactions Hyper File").

Manipuler les enregistrements lors d'une transaction : les règles à suivre

Les enregistrements modifiés pendant la transaction peuvent être lus avant ou après le début de la transaction : ils seront toujours pris en compte dans le journal des transactions.

La gestion des transactions n'exclut pas la gestion des blocages des enregistrements en transaction.

En effet, les enregistrements manipulés pendant la transaction sont automatiquement bloqués en écriture.

Dans une application en réseau, si l'utilisateur tente de modifier un enregistrement en transaction, un message apparaîtra lui demandant de retenter l'opération.

Il est donc nécessaire que la transaction soit la plus courte possible, pour éviter tout blocage des utilisateurs. Pensez donc à limiter le nombre d'instructions de parcours, modification ... sur le fichier entre les appels aux fonctions **HTransactionDébut** et **HTransactionFin**.

Que faire si une transaction a été interrompue ?

Si une panne (coupure de courant, re-démarrage du poste, ...) survient pendant une transaction, les fichiers de données risquent d'être incohérents : la transaction n'a été ni validée, ni abandonnée. Le fichier de transaction est toujours présent sur le poste.

Il est dans ce cas nécessaire de rétablir la cohérence de la base de données. Cette opération peut être automatiquement réalisée :

- lors du prochain appel à la fonction **HTransactionAnnule** ou **HTransactionDébut**.
- avec WDTRANS, utilitaire de gestion des transactions livré avec WinDev.

Pour savoir si la restauration de la cohérence de la base est nécessaire, testez le résultat de la fonction **HTransactionInterrompue** dans le code d'initialisation du projet par exemple.

Pour plus de détails, consultez l'aide en ligne (mot-clé : "Transactions Hyper File").



Notes

Transactions et Hyper File Client / Serveur

Les transactions sont disponibles en Hyper File Client / Serveur. Pour plus de détails sur les spécificités des transactions en mode Client / Serveur, consultez l'aide en ligne.

LEÇON 4.6. CRYPTAGE DE DONNÉES

Ce que vous allez apprendre dans cette leçon ...

- A quoi sert le cryptage des données ?
- Utiliser le cryptage de données.



Durée estimée : 15 mn

A quoi sert le cryptage des données ?

Une des nombreuses fonctionnalités de WinDev sur les fichiers Hyper File est de garantir la confidentialité des données contenues dans les fichiers Hyper File.

Le cryptage des fichiers permet d'assurer cette confidentialité.

Le cryptage des fichiers permet d'en rendre le contenu illisible pour toute personne ne possédant pas la clé de décryptage ("Ô Sésame, ouvres toi !"). En effet, un fichier est crypté selon une clé (appelée également "mot de passe").

Ce mot de passe est (et doit être) connu de vous seul.

C'est lors de la description du fichier sous l'éditeur d'analyses que vous avez la possibilité de définir si un cryptage doit être réalisé :

- sur le fichier lui-même
- sur l'index
- sur les fichiers mémos dépendants du fichier

Il existe plusieurs types de cryptage :

- cryptage 128 bits (performant)
- cryptage RC5 (128 bits, 12 boucles, moins performant mais plus sécurisé)
- cryptage RC5 (128 bits, 16 boucles, idem)

La clé (ou "mot de passe") que vous allez définir sera valable uniquement sur le fichier lui-même, et non pas pour l'analyse en totalité. Vous avez alors la liberté de définir un cryptage sur tout ou partie des fichiers de votre analyse.

Les données contenues dans le fichier (et/ou dans l'index et/ou dans les fichiers mémos) seront "codées" en fonction de la clé (ou "mot de passe") que vous aurez définie lors de la génération des fichiers.

Lors du décodage du fichier, le mot de passe pourra être :

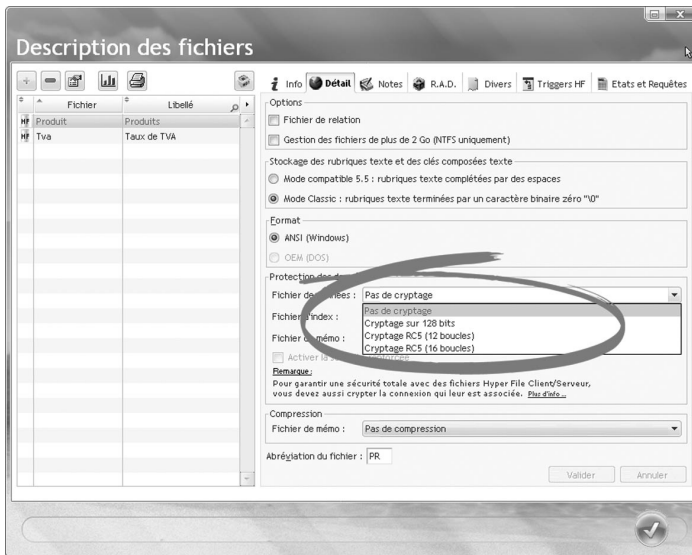
- demandé à l'utilisateur par l'intermédiaire d'une boîte de dialogue automatiquement gérée par WinDev. Dans ce cas, l'utilisateur devra connaître le mot de passe pour utiliser le fichier.
- présent directement dans le code du programme. Toute personne possédant le programme pourra utiliser le fichier, mais seulement pour les traitements que vous aurez prévus. Dans ce cas l'utilisateur n'aura pas connaissance du mot de passe.
- activé par un mode opératoire avancé : mot de passe lu dans un fichier, activé par un traitement spécial ...

La gestion de ce mot de passe incombe totalement au développeur, c'est-à-dire ... vous ! WinDev se charge de coder ou décoder en fonction du mot de passe que vous indiquez.

Comment crypter les fichiers ?

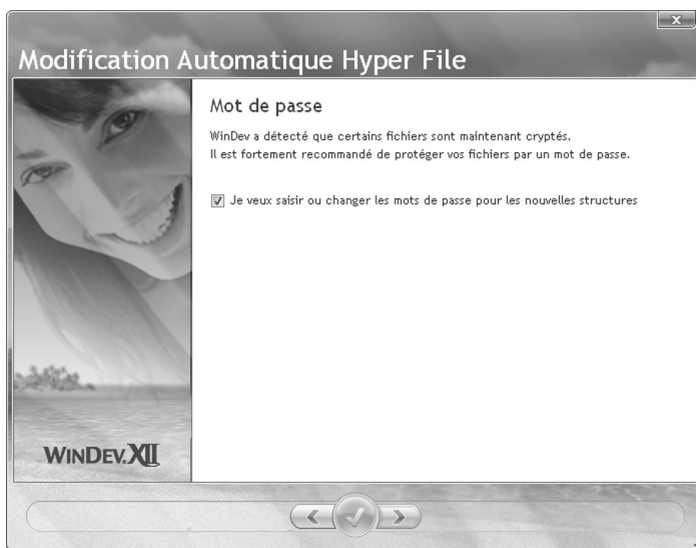
Mettre en place le cryptage des fichiers

- ▶ Pour utiliser le cryptage sur fichier :
 1. Chargez l'analyse de votre projet avec l'éditeur d'analyses (option "Projet .. Charger l'analyse").
 2. Sélectionnez le fichier que vous souhaitez crypter.
 3. Dans la description du fichier (option "Structure de fichiers .. Description", onglet "Détail"), choisissez le type de cryptage pour votre fichier, votre mémo ou votre index.



Vous avez également la possibilité d'activer la sécurité renforcée. Grâce à ce mode, le mot de passe de cryptage devra être saisi à chaque modification automatique des données (nouvelle installation, mise à jour, génération ...).

Lors de la génération de votre analyse, l'assistant vous propose de saisir ou de modifier (si vous avez déjà crypté votre fichier) le mot de passe de cryptage.



Vous obtiendrez alors la liste des fichiers pour lesquels vous avez demandé un cryptage. Vous avez la possibilité de sélectionner les fichiers à crypter, et de saisir les mots de passe pour chacun des fichiers.



Gestion du cryptage en WLangage

Pour gérer un mot de passe de cryptage en WLangage, vous pouvez :

- soit indiquer le mot passe avec les fonctions d'ouverture et de création des fichiers (fonctions **HCréation**, **HCréationSiInexistant**, **HOuvre**).

Exemple d'utilisation de la fonction **HCréationSiInexistant** :

```
HCréationSiInexistant(CLIENT, "Motdepasse")
SI HErreurMotDePasse() ALORS
    Erreur("Mot de passe invalide")
FIN
```

- soit utiliser la fonction **HPasse** avant la première opération de lecture ou d'écriture dans le fichier crypté.

Exemple d'utilisation de la fonction **HPasse** :

```
// Ouverture d'un fichier avec mot de passe et contrôle d'erreur
HPasse(CLIENT, "MotDePasse")
HCréationSiInexistant(CLIENT)
SI ErreurDétectée ALORS
    Erreur("Erreur Hyper File : " + HErreurInfo())
    RETOUR
FIN
```

Pour plus de détails sur le cryptage des données, consultez l'aide en ligne (mots-clés : "HPasse", "CryptageFIC", "CryptageMMO", "CryptageNDX").

LEÇON 4.7. LA RÉPLICATION DE DONNÉES

Ce que vous allez apprendre dans cette leçon ...

- A quoi sert la réplication de données ?
- Mise en œuvre de la réplication de données.



Durée estimée : 30 mn

A quoi sert la réplication de données ?

La réplication de données est une fonctionnalité très puissante. La réplication est l'opération permettant de maintenir à jour des bases de données distantes de structures identiques. Chacune de ces bases de données évolue indépendamment.

Grâce à la réplication, les opérations effectuées sur chacune des bases de données sont reportées sur toutes les autres bases de données.

WinDev permet de réaliser ces opérations très simplement.

WinDev propose deux types de réplication :

- **La réplication journalée** (basée sur la journalisation). Ce type de réplication permet de répliquer uniquement des bases de données Hyper File entre elles. Ce type de réplication peut être mis en place grâce aux fonctions WLangage ou grâce à l'outil WDRéplic.
- **La réplication universelle** qui permet de répliquer n'importe quel type de base de données (par exemple une base de données Hyper File avec une base de données Oracle).

Nous ne présenterons ici que la réplication universelle.

Mise en œuvre de la réplication de données

Le but de la réplication universelle est de conserver plusieurs bases de données synchronisées. Ces bases de données peuvent être de types différents ou non. Il est par exemple possible de réaliser une réplication entre une base de données Hyper File et une base de données Oracle, ou entre deux bases de données Hyper File.

La réplication universelle utilise un modèle centralisé : toutes les bases de données se synchronisent avec une base de données maître. La base de données maître répercute ensuite les modifications vers les autres bases de données.

La réplication universelle utilise plusieurs types de fichiers :

- **Fichier .RPM** : fichier permettant de décrire une base maître, et les bases qui lui sont abonnées.
- **Fichier .RPL** : fichier décrivant une base abonnée. Pour chaque base abonnée, un fichier RPL est créé. Ce fichier est présent sur le poste abonné.
- **Fichier .RPA** : fichier journal contenant les informations de réplication. Ce fichier est échangé entre la base de données maître et la base de données abonnée.
- **Fichier .SYN** : fichier contenant les informations sur la situation de la base distante. Ce fichier permet d'optimiser la taille des fichiers de synchronisation. Ce fichier est présent sur le poste maître et sur chaque poste abonné.

Remarque : Pour mettre en place la réplication universelle sur des bases de données différentes de Hyper File, il est obligatoire de créer une rubrique de type **DateHeure** dans chaque fichier. Cette rubrique sera mise à jour par l'application lors de la modification ou lors de l'ajout d'un enregistrement. Si les bases de données utilisent différents fuseaux horaires, il est conseillé d'utiliser un format universel (date et heure GMT par exemple).

Mise en place

1. Activation

Pour activer la réplication universelle, il suffit d'utiliser la fonction **HGèreRéplication** avec le paramètre **rplRéplicationUniverselle**.

Cette fonction permet de désactiver le mode de réplication standard (s'il était actif) et d'activer la réplication universelle.

2. Déclaration de la base maître

Cette opération n'est à faire qu'une seule fois sur la base maître.

Pour déclarer la base maître, il suffit d'utiliser la fonction **HCréeRéplicaMaître**.

Remarque : Si le répertoire courant permet de stocker des données, la ligne de code suivante peut être utilisée :

```
HCréeRéplicaMaître (" ")
```

Cette ligne de code crée le fichier RéplicaMaître.RPM sur le disque. Il ne reste plus qu'à inscrire les abonnés dans ce fichier.

3. Déclaration des bases abonnées

Cette opération n'est à faire qu'une seule fois pour chaque base abonnée. Cette opération doit être effectuée sur la base maître.

Pour déclarer un nouvel abonné, utilisez la fonction **HCréeRéplicaAbonné**. Cette fonction crée un abonné (fichier RPL) avec le nom fourni. Cette fonction renvoie également un numéro d'abonné.

Remarque : la fonction **HCréeRéplicaAbonné** utilise des paramètres spécifiques pour la réplication universelle. Pour plus de détails, consultez l'aide de la fonction.

4. Réplication

La fonction **HCréeRéplicaTransportable** crée un fichier journal (fichier .RPA).

Ce fichier journal est transmis et exécuté grâce à la fonction **HSynchroniseRéplica**.

Attention : Par défaut lors de la synchronisation (HSynchroniseRéplica), le maître est prioritaire : dans le cas d'une réplication abonné vers maître, les données du maître ne seront pas mises à jour. Pensez à utiliser une autre constante (rplPlusRécentPrioritaire par exemple).

Deux fonctions spécifiques peuvent également être utilisées :

HRplDéclareLiaison	Permet de signaler au moteur de réplication une liaison entre deux fichiers. Le moteur suivra alors la liaison pour obtenir la liste des enregistrements à répliquer dans le second fichier.
HRplProcédureFiltre	Permet de spécifier une procédure de filtrage spécifique lorsqu'un fichier donné sera répliqué.

LEÇON 4.8. LES TRIGGERS

Ce que vous allez apprendre dans cette leçon ...

- A quoi sert un trigger ?
- Utilisation d'un trigger.



Durée estimée : 15 mn

A quoi sert un trigger ?

Un trigger permet de déclencher une action automatiquement lors d'une opération d'ajout, de modification et de suppression sur un fichier de données Hyper File.

Lors d'une opération (ajout, modification ou suppression) sur un fichier Hyper File, il est possible d'exécuter une action avant ou après l'exécution de cette opération. Par exemple, lors d'une suppression dans un fichier, il est possible de demander la confirmation de suppression de l'enregistrement.

Mise en place

Les triggers sont définis par programmation. Les fonctions du WLangage utilisables sont :

- **HDécritTrigger** : Décrit le trigger. Cette fonction est généralement utilisée dans le code d'initialisation du projet. Le trigger déclenche l'appel d'une procédure. Cette procédure est associée au trigger lors de l'utilisation de la fonction **HDécritTrigger**.
- **HActiveTrigger** : Active un trigger décrit par la fonction **HDécritTrigger**. Dès sa création, le trigger est automatiquement activé.
- **HDésactiveTrigger** : Désactive un trigger décrit par la fonction **HDécritTrigger**.



Attention !

La fonction **HDésactiveTrigger** ne détruit pas le trigger. Le trigger existe toujours mais n'est plus actif.

- **HDétruitTrigger** : Détruit un trigger décrit par la fonction **HDécritTrigger**.

Etape 1 : décrire le trigger (ici dans le code d'initialisation du projet)

Exemple :

```
- - code d'initialisation du projet - -  
HDécritTrigger("CLIENT, FOURNISSEUR", "HSUPPRIME", ...  
    "ProcConfirmeSuppression", hTriggerAvant)
```

Dans cet exemple, le trigger va déclencher l'appel de la procédure "ProcConfirmeSuppression" lors de la suppression d'un enregistrement dans l'un des fichiers "CLIENT" et/ou "FOURNISSEUR". Cette suppression pourra avoir lieu n'importe où dans le projet.



Attention !

La procédure "ProcConfirmeSuppression" sera appelée AVANT (**hTriggerAvant**) que la suppression ait lieu.

Etape 2 : créer une procédure globale liée au trigger

Exemple :

```
PROCEDURE PROCONFIRMESUPPRESSION ()  
  // Demande la confirmation à l'utilisateur  
  SI OuiNon("Confirmez-vous la suppression ? ") = Non ALORS  
    // Initialisation de H.Afaire pour annuler l'opération  
    H.Afaire = "A"  
  FIN
```



EXEMPLE

Exemple

Pour plus de détails sur l'utilisation des triggers, consultez l'exemple "Trigger" (exemple didactique), livré en standard avec WinDev. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

Nous avons abordé simplement le fonctionnement d'un trigger. Pour obtenir plus de détails, consultez l'aide en ligne (mot-clé : "HDécritTrigger")

LEÇON 4.9. BASES EXTERNES

Ce que vous allez apprendre dans cette leçon ...

- Paramétrer la connexion à une base de données dans l'analyse
- Paramétrer la connexion à une base de données par programmation



Durée estimée : 15 mn

Présentation

On nommera "Base de données externe" tout système de gestion de base de données autre qu'Hyper File. Tout programme réalisé avec WinDev peut accéder très facilement à une base de données autre qu'Hyper File.

Il faut pour cela décrire la base de données dans une analyse (comme pour Hyper File).

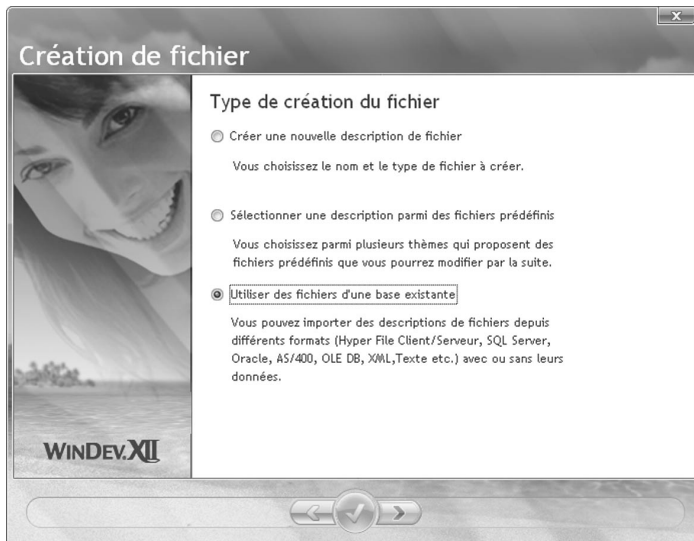
Les modes d'accès aux bases de données sont :

- accès natif
- accès via ODBC
- accès via OLE DB
- accès ODBC via OLE DB

La programmation sera effectuée avec les fonctions du WLangage (HLit*) et/ou en utilisant le langage SQL. Consultez le tableau comparatif en sélectionnant l'option "? .. Aide à l'utilisation des bases de données" dans le menu de WinDev.

Connexion à une base de données dans l'éditeur d'analyse

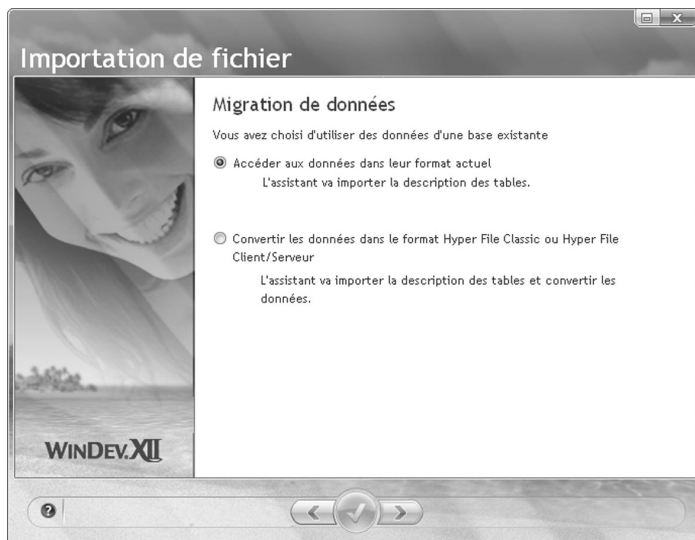
Lors de la création d'un nouveau fichier (option "Insertion .. Fichier" de l'éditeur d'analyse), choisissez l'option "Utiliser des fichiers d'une base existante" :



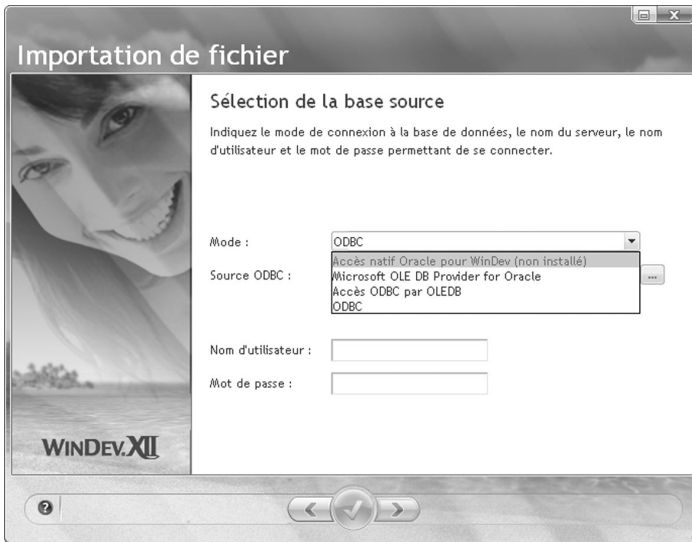
Sélectionnez ensuite le type de base de données :



Puis, définissez si les données seront converties en Hyper File Classic ou bien si elles restent dans leur format actuel :



Les paramètres de connexion à cette base de données externe doivent être décrits (nom du serveur ou service, base de données, login et mot de passe et le type d'accès (Natif, OLE DB, ODBC)).



Toutes ces informations permettent de définir une connexion dans l'analyse. Cette connexion sera automatiquement utilisée lors de la génération d'un RAD (RAD application complète ou RAD fenêtre). Tous les fichiers sont liés à cette connexion.

L'utilisation d'une connexion spécifiée dans l'analyse est très confortable pour le développement de l'application.

Mais bien souvent, les paramètres utilisés en développement correspondent aux paramètres d'un administrateur, et l'application cliente doit utiliser un mode de connexion plus sécurisé (avec moins de droits).

Il faut donc redéfinir la connexion. Deux solutions sont envisageables :

- Solution 1 : Redéfinir la connexion spécifiée dans l'analyse. Cette solution oblige à conserver des données "en dur" dans l'application, et non paramétrables.
- Solution 2 : Redéfinir la connexion spécifiée par programmation. Cette solution implique l'utilisation de deux fonctions du WLanguage **HOuvreConnexion** et **HChangeConnexion**.

Connexion à une base de données par programmation

La connexion à une base de données par programmation se fait à l'aide de deux fonctions du WLanguage :

- **HOuvreConnexion**, qui permet de décrire les différents paramètres de la connexion et de l'ouvrir.
- **HChangeConnexion**, qui permet d'associer une connexion aux différents fichiers de l'analyse.

Etudions en détail les manipulations nécessaires à la mise en place de la connexion par programmation.

Création de la connexion : HOuvreConnexion

Le plus simple pour utiliser la fonction **HOuvreConnexion** est d'utiliser l'assistant de cette fonction : vous répondez aux différentes questions et le code correspondant est automatiquement généré

dans votre traitement.

Pour utiliser l'assistant de la fonction **HOuvreConnexion** :

1. Dans l'éditeur de code, saisissez le nom de la fonction suivi de la parenthèse ouvrante :

```
HOuvreConnexion (
```

2. Cliquez sur l'option "Assistant de la fonction HOuvreConnexion" qui apparaît. L'assistant se lance automatiquement.

L'assistant demande les différents paramètres de la connexion :

- Nom de la connexion
- Provider
- Cryptage
- Utilisateur et mot de passe
- Serveur et base de données



Attention !

L'utilisateur et le mot de passe donnés dans l'assistant apparaîtront directement dans le code généré et seront "en dur".

Il peut être nécessaire de faire une fenêtre de login afin d'identifier l'utilisateur et d'utiliser des variables pour le nom de l'utilisateur et son mot de passe. Ces opérations peuvent être faites après avoir utilisé l'assistant.

3. L'assistant est terminé. Validez. Le code correspondant est automatiquement intégré dans le traitement en cours.

La fonction **HOuvreConnexion** a permis de définir une connexion, mais cette connexion n'est pas encore associée aux fichiers de données.



Attention !

Pensez à tester le résultat de la fonction **HOuvreConnexion**. Cette fonction renvoie Faux si la connexion ne peut pas être ouverte.

Association de la connexion aux fichiers de données : **HChangeConnexion**

Pour associer une connexion à un fichier, il suffit d'utiliser la fonction **HChangeConnexion**. La syntaxe de cette fonction est très simple :

- le premier paramètre correspond au nom du fichier ("*" pour tous les fichiers de l'analyse)
- le second paramètre correspond au nom de la connexion à utiliser

Exemple :

```
HChangeConnexion ("*", "MaNouvelleConnexion")
```



Notes

Il est possible de décrire autant de connexions que nécessaires.

Un fichier peut utiliser une connexion différente de la connexion utilisée par les autres fichiers présents dans l'analyse.



DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 5

**Administrer, mettre en place
une base Hyper File Client /
Serveur**

XIII

Express

LEÇON 5.1. INTRODUCTION

Ce que vous allez apprendre dans cette leçon ...

- Principe du Client / Serveur
- Pourquoi passer une application en Hyper File Client / Serveur ?

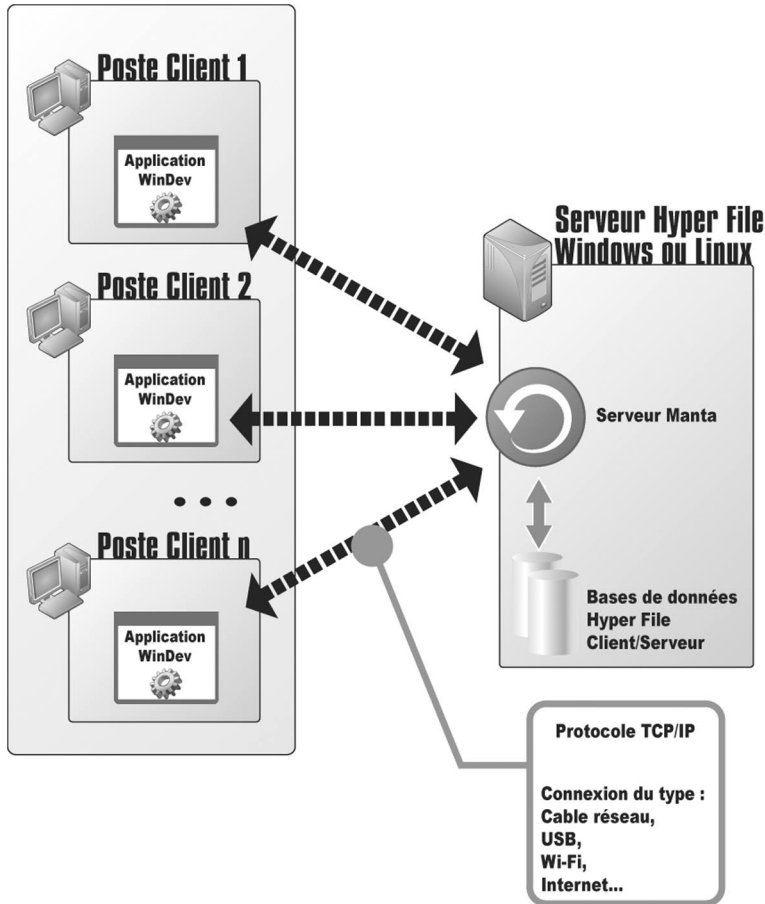


Durée estimée : 5 mn

Présentation

WinDev permet de créer des applications accédant à des bases de données Hyper File Client / Serveur.

Une application Hyper File Client / Serveur consiste à exécuter l'application sur différents postes utilisateur (appelés machines clientes) et à déporter la ou les bases de données et les traitements sur un poste serveur. Ce mode de fonctionnement permet des temps de réponses plus rapides et plus fiables, ainsi qu'une maintenance de la base de données facilitée.



WinDev permet de :

- créer entièrement une application Hyper File Client / Serveur.
- modifier une application WinDev existante en une application Hyper File Client / Serveur.

Pourquoi passer une application en mode Hyper File Client / Serveur ?

Les principaux avantages d'une application en mode Hyper File Client / Serveur par rapport à une application en mode Hyper File Classic :

- L'utilisation de Hyper File Client / Serveur est plus sécurisée (utilisation d'un login, d'un mot de passe, et définitions de droits associés aux utilisateurs).
- Pas de gestion de répertoires : tous les fichiers de la base de données sont regroupés au même endroit.
- Les clients finals ne voient pas les fichiers de données dans leur explorateur et ne peuvent pas y accéder directement.
- Les bases de données en mode Client / Serveur peuvent être utilisées par une connexion Internet.

LEÇON 5.2. MISE EN PLACE D'UNE APPLI- CATION CLIENT / SERVEUR

Ce que vous allez apprendre dans cette leçon ...

- Installation d'un serveur Hyper File Local
- Créer une application en mode Client / Serveur
- Migrer une application en mode Client / Serveur
- Utiliser le centre de contrôle Hyper File
- Fonctionnalités disponibles en mode Client / Serveur



Durée estimée : 10 mn

Présentation

Dans cette leçon, nous allons effectuer toutes les opérations nécessaires au développement et au déploiement d'une application Hyper File Client / Serveur.

Installation d'un serveur Hyper File local

La première opération à faire avant de commencer à développer une application Hyper File Client / Serveur consiste à installer un serveur Hyper File.

Sur le poste de développement, ce serveur peut être installé en local (c'est ce que nous allons faire). En déploiement, ce serveur peut être installé sur un poste spécifique.

Le programme d'installation du serveur Hyper File est disponible sur le CD de WinDev. Si vous ne possédez pas ce CD, l'installation du serveur Hyper File est également disponible sur le site de PC SOFT (www.pcsoft.fr).

Pour installer le serveur Hyper File en local :

1. Lancez le programme d'installation de WinDev.
2. Choisissez l'option "Installer ou mettre à jour un serveur Hyper File Client / Serveur".
3. Acceptez la licence.
4. Choisissez la plate-forme (option "Pour Windows sur cette machine").
5. Choisissez l'option "Installer un nouveau serveur".
6. Sélectionnez le répertoire d'installation et indiquez le nom du serveur et le port. Par défaut, le port utilisé est 4900. Vous pouvez installer le Centre de Contrôle Hyper File si celui-ci n'est pas présent ou accessible depuis votre poste.



Attention !

Le Centre de Contrôle Hyper File est nécessaire à la gestion de la base Hyper File Client / Serveur.

7. L'installation est effectuée. Par défaut, pour se connecter au serveur en mode administrateur, il faut utiliser l'utilisateur "admin" sans mot de passe.

Créer une application en mode Hyper File Client / Serveur

La création d'une application WinDev Hyper File Client / Serveur se fait de la même manière que pour toute application WinDev.

Il est nécessaire de :

1. Créer le projet en demandant à créer une nouvelle base de données.
2. Créer l'analyse en spécifiant que les bases de données utilisées par le projet seront de type "Hyper File Client / Serveur".
3. Indiquer les caractéristiques de la connexion au serveur Hyper File Client / Serveur à utiliser.
4. Lors de la création d'un fichier sous l'analyse, indiquer que ce fichier est en mode Client / Serveur et préciser la connexion utilisée.

Migrer une application Hyper File Classic en mode Client / Serveur

Présentation

La migration d'une application Hyper File Classic en mode Client / Serveur est l'opération la plus courante.

WinDev propose plusieurs solutions pour réaliser cette migration sur le poste de développement :

- réaliser la migration depuis l'éditeur d'analyse (grâce à l'option "Analyse .. Hyper File Client/Serveur .. Passage en Hyper File Client/Serveur").
- réaliser la migration depuis le centre de contrôle Hyper File.

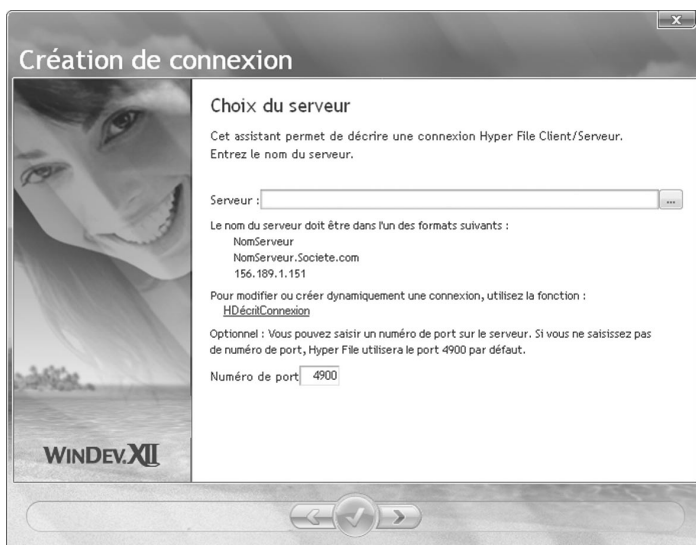
Pour mieux se rendre compte des différentes étapes, nous allons migrer l'application de gestion de comptes que nous avons réalisé dans la partie 2 de ce livre, en utilisant la première méthode.

Migration de l'exemple

Une version corrigée du projet étudié dans la partie 2 est présente dans le sous-répertoire de WinDev "Autoformation\corrigés\MesComptes". Nous allons migrer ce projet pour le tester en mode Client / Serveur.

Pour migrer le projet :

1. Ouvrez le projet "MesComptes.wdp" sous l'éditeur (option "Fichier .. Ouvrir un projet"). Ce projet est disponible dans le sous-répertoire de WinDev "Autoformation\corrigés\MesComptes"
2. Affichez l'analyse du projet (option "Projet .. Charger l'analyse"). L'éditeur d'analyse s'affiche.
3. Dans l'éditeur d'analyse, sélectionnez l'option "Analyse .. Hyper File Client / Serveur .. Passage en Hyper File Client / Serveur". Un assistant s'ouvre, permettant de créer une connexion au serveur Hyper File installé sur le poste.



4. Indiquez dans les plans suivants :

- le nom du serveur (nom de votre machine par exemple) et le numéro de port,
- le nom de l'utilisateur (laissez cette information vide pour utiliser l'administrateur)

- le nom de la base de données (MesComptes dans notre exemple)
 - le nom de la connexion (conservez le nom proposé)
5. Passez à l'écran suivant. La connexion à la base est automatiquement créée.
 6. L'assistant propose ensuite d'associer les différents fichiers présents dans l'analyse à la connexion qui vient d'être créée.



Passez à l'écran suivant.

7. L'assistant propose ensuite de créer les fichiers de données sur le serveur. Validez (option "Copier maintenant").
8. Sélectionnez les fichiers de l'analyse à copier sur le serveur : dans notre cas, ce sont tous les fichiers de données du répertoire EXE. Passez à l'écran suivant.
9. Les fichiers de l'analyse sont automatiquement transformés en fichiers Hyper File Client / Serveur et associés à la connexion choisie.
10. Générez l'analyse.



Attention !

Vérifiez le code de votre projet : en mode Hyper File Client / Serveur, les instructions HSubstRep, HChangeRep, ... sont inutiles.



Notes

Selon les paramètres indiqués lors de la création de la connexion, il est possible de modifier la connexion définie dans l'analyse grâce aux fonctions **HOuvreConnexion** et **HChangeConnexion**. Cette manipulation a été vue dans le chapitre "Bases externes", page 277.



Notes

La fonction **HOUvreConnexion** permet toujours de repasser en mode Hyper File Classic : il suffit de lui préciser le chemin du répertoire contenant les fichiers de données Hyper File Classic.

11. Vous avez migré le projet de développement. Il peut être également nécessaire de migrer l'application déployée (par exemple si l'application déployée utilise des fichiers Hyper File Classic). Cette opération se paramètre lors de la création du programme d'installation de l'application.

Fonctionnalités disponibles en mode Hyper File Client / Serveur

Hyper File Client / Serveur propose de nombreuses fonctionnalités propres au mode Client / Serveur :

- Transactions
- Journal
- Procédures stockées
- Triggers

Nous ne détaillerons pas ici ces fonctionnalités (certaines ont été abordées dans ce cours en mode Hyper File Classic). Nous vous conseillons de consulter l'aide en ligne pour plus de détails sur ces sujets.

LEÇON 5.3. ADMINISTRER

Ce que vous allez apprendre dans cette leçon ...

- Le centre de contrôle Hyper File
- Créer un utilisateur dans le Centre de Contrôle Hyper File
- Fonctions WLangage permettant d'administrer
- Sauvegarder la base



Durée estimée : 20 mn

Présentation

Maintenant que nous savons créer / migrer une application en mode Hyper File Client / Serveur, nous allons voir comment l'administrer.

En effet, une base de données Client / Serveur nécessite :

- une configuration spécifique des postes (installation d'un serveur Hyper File, ...)
- une administration réalisée grâce au Centre de Contrôle Hyper File.

Configuration des postes

Pour utiliser une base Hyper File Client / Serveur, il est nécessaire d'installer un serveur Hyper File sur le poste serveur. Il est possible d'utiliser plusieurs serveurs Hyper File sur le même poste, utilisant des ports différents. Sur chaque serveur, une ou plusieurs bases de données peuvent être installées.

Par exemple, il est possible d'installer **sur le même poste** un serveur Hyper File de tests, avec une base de données de tests, et un serveur Hyper File de production, utilisant un port différent.

Le centre de contrôle Hyper File

Le Centre de Contrôle Hyper File permet de réaliser toutes les opérations d'administration des serveurs et des bases de données Hyper File Client / Serveur.

Nous allons voir les fonctionnalités les plus importantes.

Pour lancer le centre de contrôle Hyper File et accéder aux données :


1. Dans l'éditeur, sélectionnez l'option "Outils .. Centre de Contrôle Hyper File".

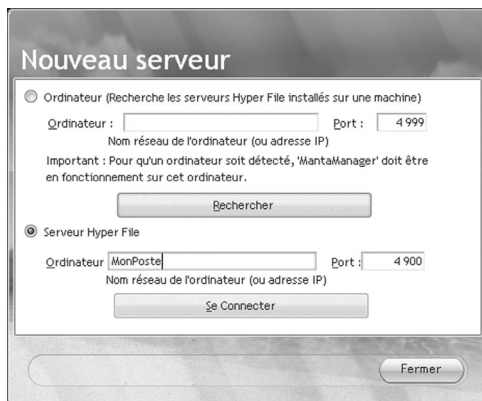


2. Cliquez sur "Hyper File Client/Serveur". La zone qui apparaît peut être vide si le Centre de Contrôle Hyper File n'a jamais été utilisé depuis ce poste.

Depuis cet écran, vous pouvez :

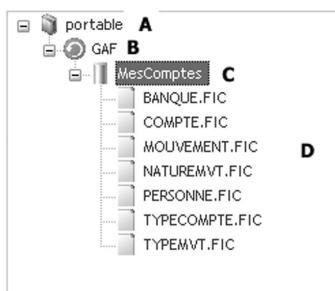
- Rechercher les serveurs Hyper File accessibles depuis le réseau (opération relativement longue)
- Rechercher les serveurs Hyper File accessibles depuis un ordinateur. C'est ce que nous allons faire.

3. Cliquez sur l'icône . Dans l'écran qui apparaît, sélectionnez par exemple "Serveur Hyper File" et indiquez le nom de votre poste.



Cliquez ensuite sur "Se Connecter". Le poste apparaît dans le Centre de Contrôle Hyper File.

4. Sélectionnez votre poste, votre serveur Hyper File et votre base "MesComptes". Donnez les paramètres de connexion (Admin sans mot de passe par défaut). La liste des fichiers s'affiche.
5. L'organisation des données est la suivante :



A - Nom du poste

B - Nom du serveur

C - Nom de la base

D - Fichiers de données de la base.

6. Quelques exemples des informations accessibles (pour plus de détails, consultez l'aide en ligne) :

- L'onglet "Description" présente les informations sur les fichiers de données (nombre d'enregistrements, ...)
- Si vous sélectionnez un fichier de données, l'onglet "Contenu" présente les données enregistrées, ...

Depuis le Centre de Contrôle Hyper File, il est possible d'administrer toute la base de données Hyper File Client / Serveur.

Créer un compte utilisateur dans le Centre de Contrôle Hyper File

Nous avons vu que lors de l'installation d'un serveur Hyper File et lors de la création d'une base de données, un seul compte utilisateur est créé : le compte de l'administrateur (login "Admin" sans mot de passe).

L'utilisation de compte utilisateur permet de sécuriser l'accès aux données. En effet, tous les utilisateurs de l'application ne sont pas des administrateurs. A chaque utilisateur (ou groupe d'utilisateurs), il est possible d'associer des droits.



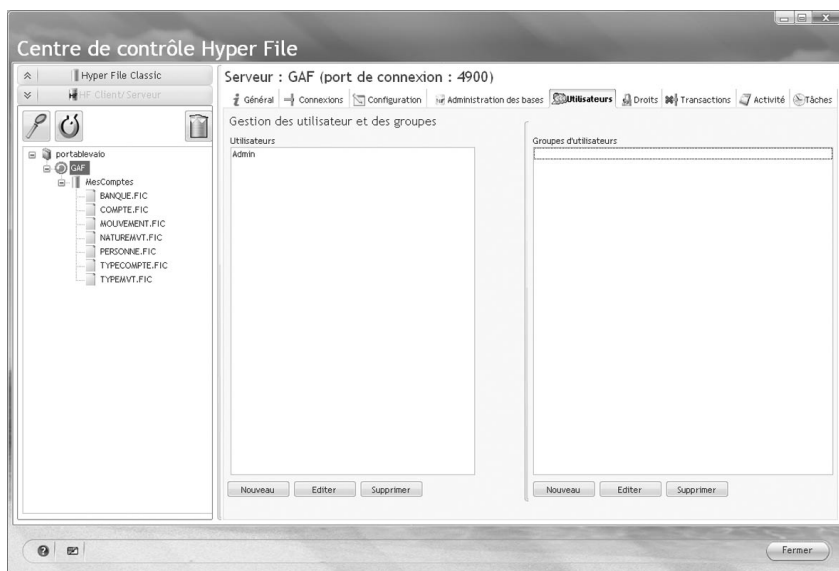
Attention !

Les droits des utilisateurs spécifiés dans le Centre de Contrôle Hyper File sont des droits de base de données et pas des droits d'application.

Il ne faut pas confondre la gestion des droits des bases de données Client / Serveur avec le groupware utilisateur.

Certains utilisateurs peuvent par exemple ne pas avoir le droit d'écrire dans certains fichiers. Pour faire un test simple, nous allons créer un utilisateur et lui permettre de consulter les enregistrements du fichier Compte.

1. Dans le Centre de Contrôle Hyper File, positionnez-vous sur le serveur Hyper File.
2. Dans la partie droite de l'écran, sélectionnez l'onglet "Utilisateurs". Cet onglet permet de gérer les utilisateurs du serveur. Seul pour le moment l'utilisateur "Admin" est présent.



3. Pour créer un nouvel utilisateur, cliquez sur "Nouveau". L'écran permettant de définir les

caractéristiques de l'utilisateur apparaît. Saisissez par exemple les informations suivantes :



Notes

Plusieurs caractéristiques peuvent être notées :

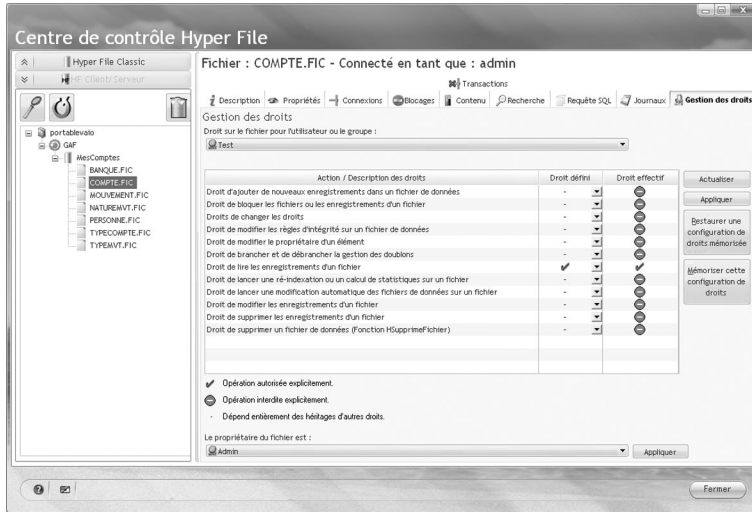
- **Super Utilisateur** : Les utilisateurs désignés comme "Super utilisateur" sont autorisés à effectuer toutes les actions sur le serveur, les bases de données et sur tous les fichiers.
- **Compte actif** : Si cette option n'est pas cochée, l'utilisateur existe mais n'est pas actif (cas des utilisateurs en congés par exemple).
- **Expiration du mot de passe** : Il est possible d'avoir un mot de passe valide pour quelques jours.

4. Validez la création de l'utilisateur. Par défaut, cet utilisateur n'a aucun droit.

Nous allons maintenant attribuer des droits à l'utilisateur : l'utilisateur TEST peut consulter le fichier COMPTE.

1. Dans le Centre de Contrôle Hyper File, sélectionnez la base de données MESCOMPTE, puis le fichier COMPTE.
2. Cliquez sur l'onglet "Droits".
3. Sélectionnez l'utilisateur "TEST" dans la liste des utilisateurs.
4. Dans la liste des droits, sélectionnez "Droit de lire les enregistrements d'un fichier" et sélectionnez la coche verte dans la colonne "Droit défini".

5. Cliquez sur le bouton "Appliquer". Le droit devient effectif.



De cette façon, il est possible de définir des droits :

- sur le serveur Hyper File
- sur la base de données
- sur les fichiers de la base de données.

Dans notre exemple, l'utilisateur TEST pourra uniquement parcourir les enregistrements du fichier COMPTE. Si cet utilisateur tente une autre action, un message s'affichera : "L'utilisateur TEST n'a pas les droits suffisants pour " .

Une fois le compte créé, il peut être utilisé lors de la connexion de l'application au serveur (quand la fonction **HOUvreConnexion** est utilisée).



Notes

La gestion des utilisateurs et de leurs droits peut également être effectuée par programmation grâce aux fonctions WLanguage. Pour plus de détails, consultez l'aide en ligne.

Sauvegarde de la base

La sauvegarde de la base de données peut être réalisée directement grâce à l'onglet "Sauvegarde" disponible lors de la sélection de la base de données dans le volet gauche du Centre de Contrôle Hyper File.

Conclusion

Le centre de contrôle Hyper File est un outil complet d'administration de base de données, permettant entre autre :

- d'arrêter ou de redémarrer un serveur en cas de problème.
- de gérer les utilisateurs et leurs droits
- de réindexer les fichiers de données si nécessaire
- de faire des sauvegardes de la base.

Le Centre de contrôle Hyper File est un outil redistribuable qui doit être installé chez les clients possédant des bases de données Hyper File Client / Serveur. Le centre de Contrôle Hyper File doit être utilisé par l'administrateur de la base de données.

LEÇON 5.4. INSTALLATION EN CLIENTÈLE

Ce que vous allez apprendre dans cette leçon ...

- Comment installer une application Client / Serveur en clientèle ?



Durée estimée : 5 mn

WinDev 12 version "Express" :

Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

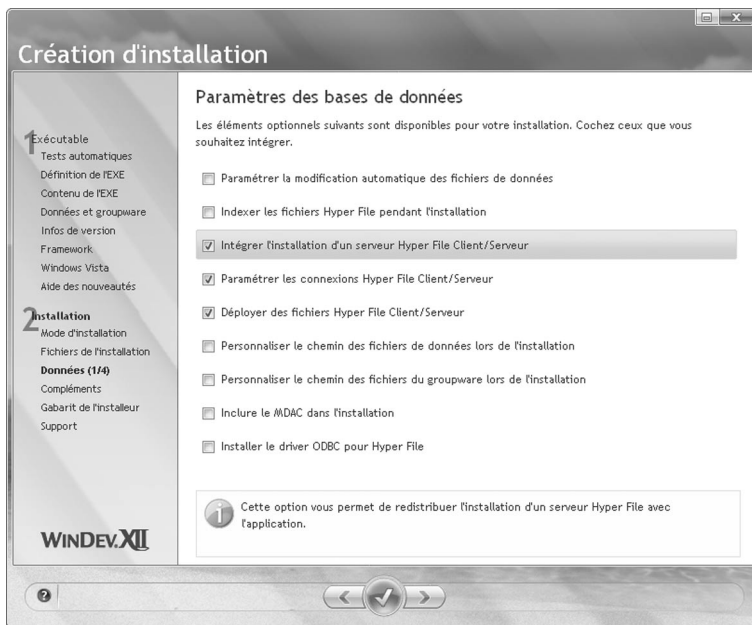
Présentation

L'installation d'une application Client / Serveur nécessite plusieurs options spécifiques. Ces options sont prises en charge par l'assistant de création d'installation de WinDev.
Nous allons créer la procédure d'installation de notre application "Mes Comptes".

Lancement de l'assistant de création d'installation

Pour lancer la procédure d'installation d'une application Hyper File Client / Serveur :

1. Sélectionnez l'option "Atelier .. Créer la procédure d'installation".
2. Créez l'exécutable et validez la création de l'aide.
3. Allez jusqu'à l'écran "Paramètres des bases de données". Dans ce plan, il est possible de :
 - Intégrer l'installation d'un serveur Hyper File Client / Serveur
 - Paramétrer les connexions Hyper File Client / Serveur
 - Déployer les fichiers Hyper File Client / Serveur



Si ces options sont cochées, les écrans suivants permettent de paramétrer les différents éléments à prendre en compte pour l'installation en clientèle.
Pour plus de détails, consultez l'aide en ligne.



DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 6

Les requêtes

XIII

Express

LEÇON 6.1. CRÉEZ UNE REQUÊTE DE SÉLECTION

Ce que vous allez apprendre dans cette leçon ...

- Manipulations de l'éditeur de requêtes.
- Etape par étape, la réalisation d'une requête.



Durée estimée : 30mn

Introduction

Une requête permet de définir une interrogation sur des fichiers de données. L'éditeur de requêtes vous permettra par exemple de décrire sans aucune programmation des interrogations du type :

- la liste des fournisseurs appartenant au département 75.
- le nombre d'annonces par ville.
- la liste des clients qui ont le plus commandé.

Et ce, sur tous les types de bases (Hyper File, Oracle, ...) !

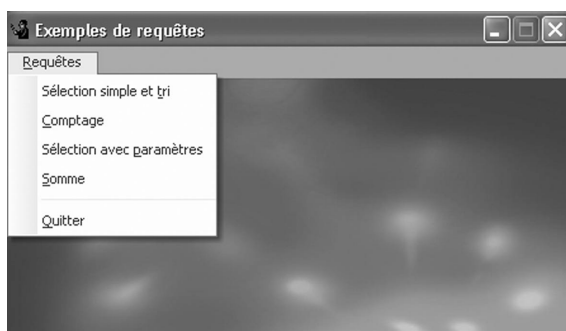
Nous allons voir avec un exemple simple comment manipuler l'éditeur de requêtes. Plus loin dans ce cours, nous verrons une utilisation avancée de l'éditeur de requêtes.

Dans cette leçon, vous verrez comment :

- Faire une requête simple correspondant à une liste avec une sélection et un tri.
- Faire une requête de comptage.
- Faire une requête multi-fichiers.

- ▶ Dans le menu de WinDev, sélectionnez l'option "? .. Guide d'Auto formation .. Faire des requêtes".

Le programme se lance. La fenêtre ci-dessous apparaît.



Vous pouvez parcourir les différentes options du menu et choisir des exemples de requêtes.



EXEMPLE

Exemple

L'exemple "Requêtes" (exemple didactique) livré avec WinDev illustre l'utilisation des requêtes avec WinDev. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

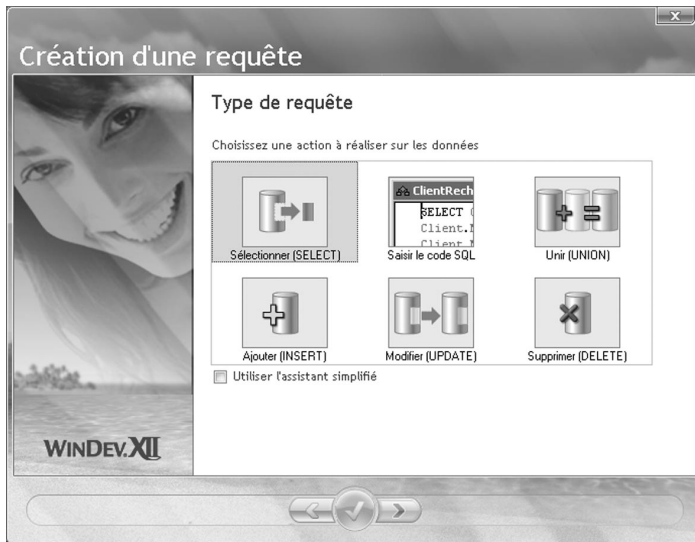
- ▶ Ouvrez le projet "Faire des requêtes.WDP" présent dans le sous-répertoire "Auformation\Exercices\Faire des requêtes" du répertoire d'installation de WinDev 12.

Votre première requête

Nous allons lister les fournisseurs du département 75.

- ▶ Créez une nouvelle requête :
 1. Dans le menu, sélectionnez "Fichier" puis "Nouveau".
 2. Sélectionnez "Requête".

L'assistant de création de requêtes se lance :



Vous avez la possibilité d'utiliser l'assistant de création de requêtes ou de saisir directement le code SQL de la requête (option "Saisir le code SQL").

Avec l'assistant, vous pouvez réaliser des requêtes de type :

- "Sélectionner (SELECT)" : interrogation
- "Ajouter (INSERT)" : ajout dans un fichier
- "Modifier (UPDATE)" : modification dans un fichier
- "Supprimer (DELETE)" : suppression dans un fichier
- "Unir (UNION)" : combinaison de requêtes de sélection



Attention !

La saisie directe de code SQL est destinée aux utilisateurs maîtrisant le langage SQL ! Consultez l'aide en ligne pour obtenir la liste des fonctions SQL reconnues par WinDev (mot-clé : "SQL, Langage SQL").

Vous allez apprendre à utiliser l'éditeur de requêtes en effectuant des requêtes simples : sélections de clients, comptages ... vous pourrez ensuite approfondir vos connaissances en manipulant et en testant tous les types de requêtes.

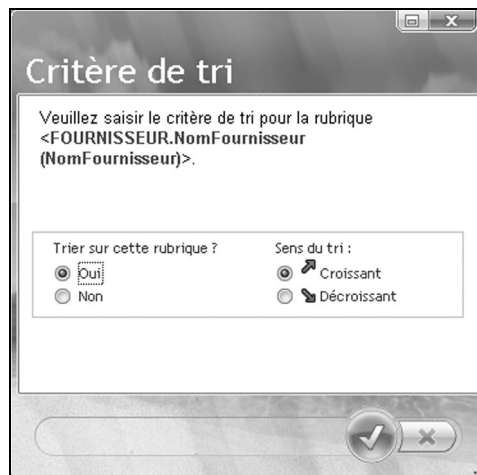
- ▶ Pour lister les fournisseurs du département 75, nous allons réaliser une requête de sélection. Passez à l'écran suivant.
- ▶ Sélectionnez les rubriques qui constitueront le résultat de la requête. Sur la gauche, dans les "Eléments disponibles",
 1. Sélectionnez le fichier "Fournisseur".
 2. Double-cliquez sur les rubriques "SOCIETE", "NOMFOURNISSEUR", "CODEPOSTAL" et "VILLE". Ces rubriques apparaissent au milieu de la fenêtre.



- ▶ Pour obtenir les fournisseurs du département 75, nous allons appliquer une condition de sélection sur le code postal. La condition de sélection est la suivante : "Nous désirons les fournisseurs du département 75". C'est-à-dire les fournisseurs dont le code postal **commence** par 75. En effet, la rubrique "Département" n'existe pas dans notre fichier.
- ▶ Sélectionnez la rubrique "CODEPOSTAL" au centre. Dans les "Actions" sur la droite, cliquez sur le bouton "Condition de sélection" puis sur "Nouvelle condition". La rubrique est donc : "FOURNISSEUR.CODEPOSTAL" La condition est : "COMMENCE PAR" La valeur est : "75"

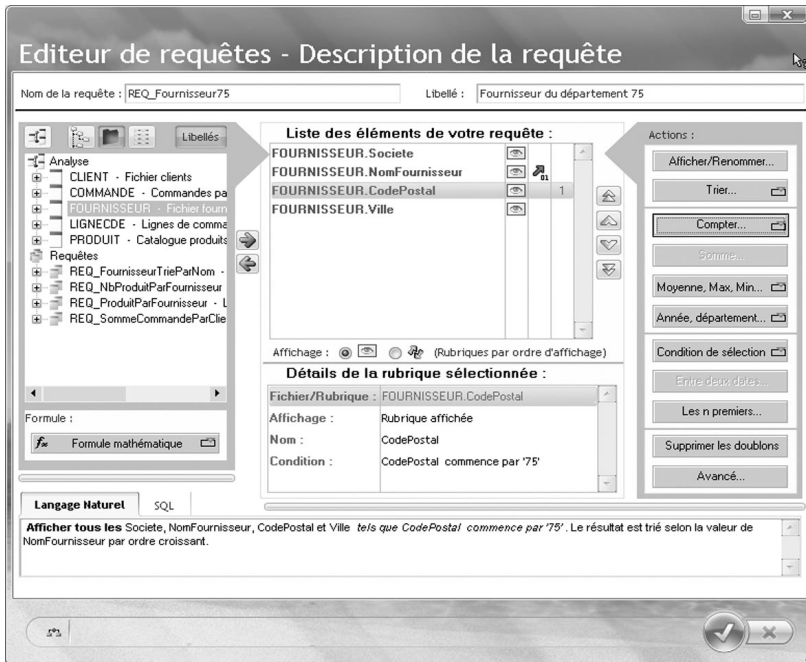


- ▶ Saisissez la valeur et cliquez sur "OK".
- ▶ Nous allons maintenant indiquer le critère de tri.
Nous allons trier le résultat par "NOMFOURNISSEUR".
- ▶ Sélectionnez la rubrique "NOMFOURNISSEUR" au centre puis cliquez dans les "Actions" sur le bouton "Trier" et sélectionnez l'option "Tri sur la rubrique sélectionnée".
Le tri sera croissant :

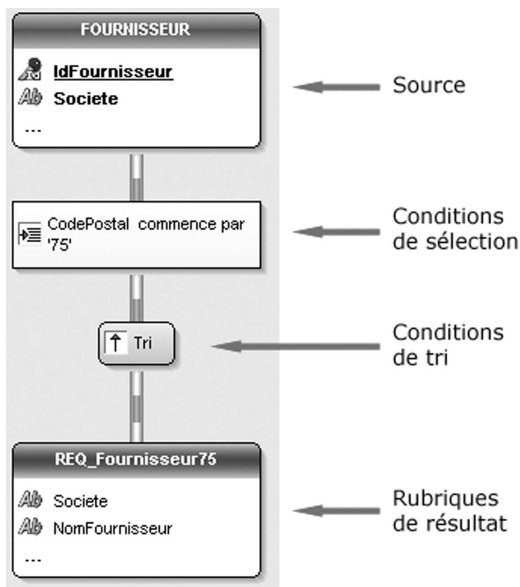



- ▶ Validez en cliquant sur le bouton "OK".
Une flèche rouge avec le numéro 01 apparaît à la droite de la rubrique "NOMFOURNISSEUR". Cette flèche indique qu'un tri croissant est fait sur cette rubrique. Le chiffre "01" indique que ce sera le premier tri réalisé.

- ▶ La requête est créée. Nous allons lui donner un nom et la sauvegarder.
 1. En haut de l'écran, saisissez le nom "REQ_Fournisseur75".
 2. Saisissez le libellé "Fournisseurs du département 75".
 3. Cliquez sur "OK".



La représentation graphique de la requête est la suivante :



► Cliquez sur  pour sauvegarder la requête.

Il ne reste plus qu'à tester la requête.

► Cliquez sur le bouton "GO" . Le résultat s'affiche dans une table :

Aperçu de REQ_Fournisseur75

4 enregistrements correspondent à votre requête.

Societe	NomFournisseur	CodePostal	PAI
MERGATE	CONNER ALFRED	75000	PAI
WINNER	GAYANT CHRISTOPHE	75012	PAI
YELLOW	LEDEUF DOMINIQUE	75015	PAI
WORLD COMPANY	SYLVESTRE	75000	PAI

Utiliser le clic droit pour imprimer, exporter... vos enregistrements.

Exporter vers un fichier Hyper File...

Fermer

Si vous faites un clic droit sur la table affichant le résultat de la requête, un menu contextuel apparaît. Vous avez alors la possibilité d'exporter le résultat vers :

- Microsoft Excel.
- un fichier XML (*eXtensible Markup Language*)
- Microsoft Word.

LEÇON 6.2. REQUÊTES AVEC COMPTAGE

Ce que vous allez apprendre dans cette leçon ...

- Manipulations de l'éditeur de requêtes.
- Etape par étape, la réalisation d'une requête avec comptage.



Durée estimée : 20mn

Réalisation d'une requête avec comptage

Nous allons maintenant créer une requête qui effectue un comptage. Cette requête permet de comptabiliser le nombre d'occurrences d'une valeur dans un fichier.

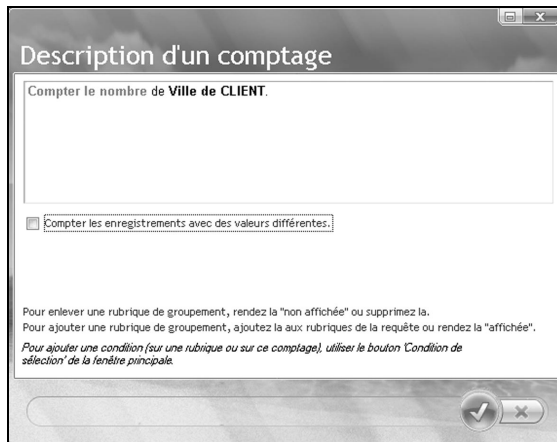
Nous allons compter le nombre de clients par ville.

- ▶ Ouvrez le projet "Faire des requêtes.WDP" présent dans le sous-répertoire "Auformation\Exercices\Faire des requêtes" du répertoire d'installation de WinDev 12.
- ▶ Créez une nouvelle requête :
 1. Dans le menu, sélectionnez "Fichier" puis "Nouveau".
 2. Sélectionnez "Requête".
 3. Nous allons réaliser une requête de "sélection" (Option "Sélectionner (SELECT)"). Passez à l'écran suivant.



Nous allons sélectionner les rubriques à récupérer dans le résultat de la requête.

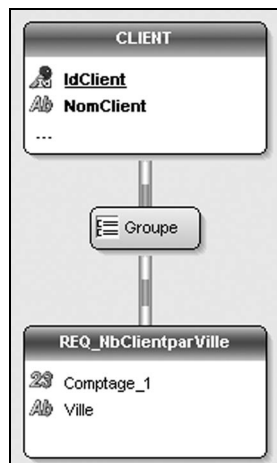
- ▶ Nous devons afficher la ville. Sur la gauche, dans les "Éléments disponibles", sélectionnez la rubrique "VILLE" du fichier "CLIENT" (par un double-clic).
- ▶ Pour créer une rubrique de comptage :
 1. Sélectionnez au centre la rubrique "VILLE"
 2. Cliquez dans les "Actions" sur le bouton "Compter".
 3. Sélectionnez dans le menu : "Sur la rubrique sélectionnée".
 4. Validez en cliquant sur le bouton "OK".





L'action "Compter les enregistrements" est ajoutée dans le résultat de la requête. Comme nous voulons afficher toutes les villes, sélectionnez à nouveau la rubrique "VILLE" du fichier "CLIENT" (par un double-clic). La rubrique est affichée au centre de l'écran.

- ▶ La requête est créée. Nous allons lui donner un nom et la sauvegarder.
 1. En haut de l'écran, saisissez le nom "REQ_NbClientparVille".
 2. Saisissez le libellé "Nombre de clients par ville".
 3. Cliquez sur "OK".

La représentation graphique de la requête est la suivante :



La rubrique "Comptage_1" contient le comptage des enregistrements décrit plus haut.

- ▶ Cliquez sur  pour sauvegarder la requête. Il ne reste plus qu'à tester la requête.
- ▶ Cliquez sur le bouton .

LEÇON 6.3. REQUÊTES SOMME

Ce que vous allez apprendre dans cette leçon ...

- Manipulations de l'éditeur de requêtes.
- Etape par étape, la réalisation d'une requête "Somme".



Durée estimée : 20mn

Réalisation d'une requête "Somme"

Nous allons maintenant créer une requête qui effectue une somme. Nous allons faire un cumul d'une valeur d'un fichier.

Nous allons calculer le total de toutes les commandes par client (le Chiffre d'Affaires par client).

- ▶ Ouvrez le projet "Faire des requêtes.WDP" présent dans le sous-répertoire "Auformation\Exercices\Faire des requêtes" du répertoire d'installation de WinDev 12.
- ▶ Créez une nouvelle requête :
 1. Dans le menu, sélectionnez "Fichier .. Nouveau".
 2. Sélectionnez "Requête".
 3. Nous allons réaliser une requête de sélection. Sélectionnez l'option "Sélectionner (SELECT)". Passez à l'écran suivant.

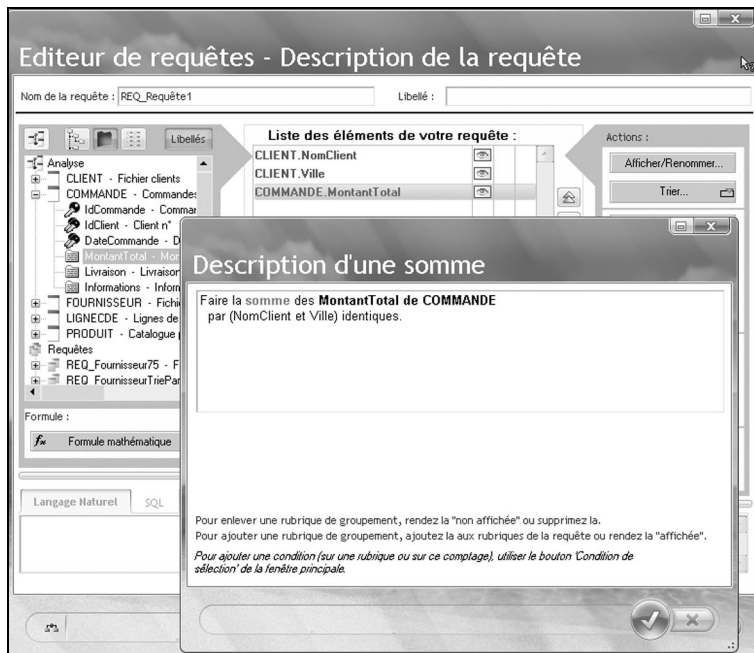
Nous allons sélectionner les rubriques à récupérer dans le résultat de la requête.

Nous afficherons tout d'abord le nom du client et la ville du client.

- ▶ Sur la gauche, dans les "Eléments disponibles", sélectionnez les rubriques "NOMCLIENT" et "VILLE" du fichier "CLIENT" (par un double-clic).

Ensuite, nous devons sélectionner la rubrique sur laquelle nous allons faire la **somme** (MONTANT-TOTAL dans cet exemple).

- ▶ Sur la gauche, dans les "Eléments disponibles", sélectionnez la rubrique "MONTANTTOTAL" du fichier "COMMANDE" (par un double clic).
- ▶ Pour effectuer la somme des valeurs de "MONTANTTOTAL" :
 1. Sélectionnez la rubrique "MONTANTTOTAL" au centre.
 2. Dans les "Actions", sur la droite, sélectionnez "Somme".
 3. Validez en cliquant sur le bouton "OK" :



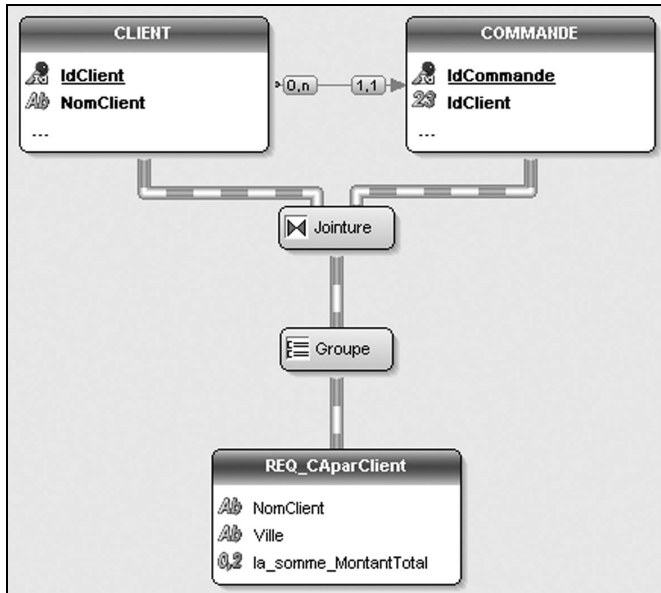
La somme des "MONTANTTOTAL" a été ajoutée dans la liste du résultat de la requête. Au fur et mesure de la création de la requête, l'éditeur de requêtes de WinDev crée les requêtes en langage naturel (et également en SQL).





Vous pouvez ainsi vérifier que vous ne vous êtes pas trompé par rapport à l'objectif de votre requête.

- ▶ La requête est créée. Nous allons lui donner un nom et la sauvegarder.
 1. En haut de l'écran, saisissez le nom "REQ_CAprClient".
 2. Saisissez le libellé "Somme des commandes par client".
 3. Cliquez sur "OK".

La représentation graphique de la requête est la suivante :



La représentation graphique correspond à une "jointure". La requête utilise deux sources de données (les fichiers "CLIENT" et "COMMANDE").

- ▶ Cliquez sur  pour sauvegarder la requête. Il ne reste plus qu'à tester la requête.
- ▶ Cliquez sur le bouton .

LEÇON 6.4. LE LANGAGE SQL

Ce que vous allez apprendre dans cette leçon ...

- Les principaux ordres du langage SQL.



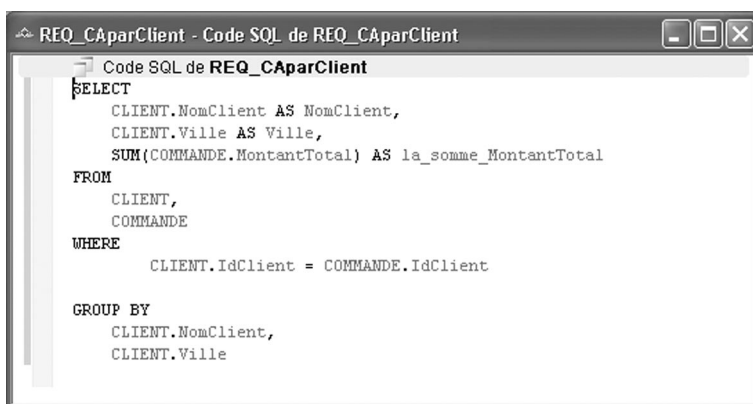
Durée estimée : 15mn

Code source d'une requête

Dans WinDev, la requête est représentée dans l'éditeur sous une forme graphique. Nous allons découvrir maintenant le code source correspondant à une requête. Le langage utilisé pour présenter une requête est le langage SQL. Il s'agit ici d'une présentation de SQL et non pas d'un cours complet (il existe des livres entiers sur le sujet !).

Nous allons visualiser le code source SQL de la requête créée dans la leçon précédente.

- ▶ Ouvrez la requête "REQ_CAprClient".
Si vous n'avez pas précédemment créé cette requête, ouvrez la requête "REQ_CAprClient" présente dans le répertoire "\\Autoformation\\Corrigés\\Faire des requêtes".
- ▶ Dans le menu, sélectionnez l'option "Requête .. Code SQL".



```
Code SQL de REQ_CAprClient
SELECT
  CLIENT.NomClient AS NomClient,
  CLIENT.Ville AS Ville,
  SUM(COMMANDE.MontantTotal) AS la_somme_MontantTotal
FROM
  CLIENT,
  COMMANDE
WHERE
  CLIENT.IdClient = COMMANDE.IdClient

GROUP BY
  CLIENT.NomClient,
  CLIENT.Ville
```

La requête en langage SQL est composée de différents mots clés :

- l'instruction SELECT
- l'instruction FROM
- la clause WHERE
- la clause GROUP BY



Attention !

Veillez à bien respecter l'enchaînement de ces ordres. Vous ne pouvez pas inverser cet enchaînement.

WinDev permet de passer très simplement de la description graphique d'une requête au code SQL de cette requête et inversement.

Pour afficher le code SQL d'une requête créée sous l'éditeur de requêtes, sélectionnez l'option "Requête .. Code SQL".

Pour afficher la représentation graphique d'une requête créée en code SQL, sélectionnez "Requête .. Rétro-analyse de la requête".

Les ordres SQL

L'instruction SELECT

L'instruction SELECT permet de définir la liste des rubriques et des calculs à visualiser dans le résultat. La syntaxe est la suivante :

```
SELECT NOMRUBRIQUE1, NOMRUBRIQUE2 ...
```

Chaque rubrique peut être renommée (on parle d'alias) en utilisant le mot-clé AS. Par exemple :

```
CLIENT.NOMCLIENT AS NOM
```

Dans cet exemple, la rubrique *NomClient* du fichier CLIENT est renommée en *Nom*.

NomRubriqueXX peut être soit une rubrique associée à un fichier, soit une formule de calcul.

L'instruction FROM

L'instruction FROM permet de définir la liste des fichiers à partir desquels seront extraites les rubriques et les calculs. La syntaxe est la suivante :

```
FROM FICHIER1, FICHIER2 ...
```

FichierXX correspond à un des fichiers de l'analyse. L'utilisation de **AS** est possible.

La clause WHERE

La clause WHERE contient les conditions de sélection ainsi que les conditions de jointure. Ces conditions permettent de filtrer les enregistrements issus des fichiers de la base de données. Seuls les enregistrements vérifiant la ou les conditions seront récupérés dans le résultat.

Une jointure permet de faire correspondre à un enregistrement d'un fichier l'enregistrement d'un fichier relié. Par exemple, une jointure entre les fichiers CLIENT et COMMANDE permettra de connaître toutes les commandes de chaque client.

La syntaxe est la suivante :

```
WHERE CONDITION1 ET/OU CONDITION2
```

Les opérateurs ET et OU permettent de réaliser une opération logique entre deux conditions.

- L'opérateur ET indique que les deux conditions doivent être vérifiées en même temps.
- L'opérateur OU indique qu'une des deux conditions ou les deux conditions doivent être vérifiées.

La clause WHERE n'est pas obligatoire. Si ce mot-clé n'est pas indiqué, tous les enregistrements de tous les fichiers seront sélectionnés.

La clause GROUP BY

La clause GROUP BY permet d'indiquer les rubriques ou formules de regroupement lorsqu'il y a un calcul. Par exemple, dans la requête que nous avons réalisée précédemment, le calcul portait sur la somme des commandes par client; c'est-à-dire que les commandes étaient cumulées pour chaque client. Le regroupement des commandes était donc réalisé par client.

La syntaxe est la suivante :

```
GROUP BY RUBRIQUE1, RUBRIQUE2 ...
```

RubriqueXX correspond à une rubrique d'un des fichiers ou à une formule.

La clause ORDER BY

La clause ORDER BY permet d'indiquer les critères de tri de la requête. Les critères de tri permettent de classer le résultat de la requête.

La syntaxe est la suivante :

```
ORDER BY RUBRIQUE1, RUBRIQUE2 ...
```

RubriqueXX correspond à une rubrique d'un des fichiers ou à une formule.



Notes

L'éditeur de requêtes génère automatiquement le code SQL. C'est également un moyen didactique de se former à ce langage !

LEÇON 6.5. UTILISATION DES REQUÊTES

Ce que vous allez apprendre dans cette leçon ...

- Utilisation des requêtes dans les états.
- Utilisation des requêtes dans les champs (listes, combos, tables, zones répétées).
- Utilisation des requêtes en WLangage.



Durée estimée : 15mn

Utilisation d'une requête depuis un état

Lors de la création d'un état, vous pouvez choisir la source de données. La source de données peut être une requête, un fichier de données, ... Il est possible de faire un état qui travaille directement sur le résultat d'une requête (pour plus de détails, consultez la leçon sur les états, page 329).

Utilisation d'une requête depuis un champ

Comme pour les états, lors de la création d'un champ table, d'une liste, d'une combo ou d'une zone répétée, il est possible de définir la source de données pour le remplissage. Cette source peut être un fichier ou une requête.

Utilisation d'une requête depuis le WLangage

Vous pouvez également utiliser une requête comme si vous utilisiez un fichier de données. Vous devez uniquement initialiser l'exécution de votre requête avec la fonction **HExécuteRequête**. Lorsque votre parcours est terminé, utilisez la fonction **HAnnuleDéclaration**. Cette fonction optionnelle permet d'annuler la définition de la requête. Votre parcours peut s'effectuer normalement avec les fonctions **HLitPremier** et **HLitSuivant** :

```
// Initialise la requête
HExécuteRequete (MaRequête)
SI ErreurDétectée ALORS
    Erreur("Erreur d'initialisation de la requête"+RC+...
        HErreurInfo())
    RETOUR
FIN
// lecture du premier enregistrement de la requête
HLitPremier (MaRequête)
TANTQUE PAS HEnDehors()
    // Traitement sur l'enregistrement de la requête
    // lecture de l'enregistrement suivant
    HLitSuivant()
FIN
HAnnuleDéclaration (MaRequête) //optionnel
```



Notes

Offrez la possibilité aux utilisateurs de vos applications de créer eux-mêmes leurs états et requêtes (ou de modifier des états ou des requêtes que vous mettez à leur disposition) !

Il suffit de livrer le logiciel "Etats et requêtes" avec votre application. Pour plus de détails, consultez le chapitre "Les états", page 366.



DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 7

Les états

XIII ***Express***

LEÇON 7.1. VOTRE PREMIER ETAT

Ce que vous allez apprendre dans cette leçon ...

- Vocabulaire de base.
- Etape par étape, la construction d'un état.



Durée estimée : 30mn

Vocabulaire de base

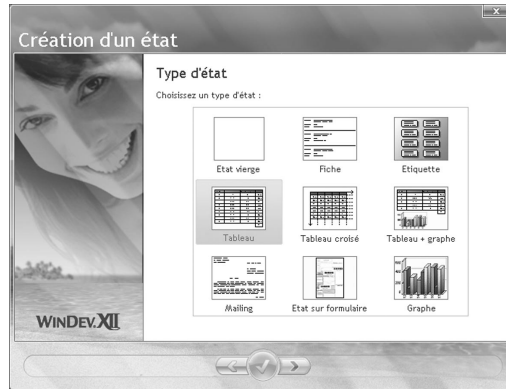
Avant d'aborder les méthodes d'impression, il est important de rappeler le vocabulaire de base pour faire bonne impression !

- **Fonte** : La fonte définit la forme du caractère (épaisseur de trait, espacement entre caractères, ...). Par exemple, ce document est imprimé en fonte "Franklin Gothic" (les exemples de programme en fonte "Courier").
Les fontes sont fréquemment appelées **polices** par "abus" de langage : nous parlerons donc plutôt de police que de fonte.
- **Style** : Le style peut être *italique*, **gras**, souligné, ... Il ne modifie pas la hauteur des caractères.
- **Hauteur du caractère** : La hauteur du caractère s'exprime en "point Didot". Un point vaut 0,3759 mm. C'est une unité utilisée en imprimerie. La hauteur des caractères de ce manuel est par exemple de 10 points. Les hauteurs de 10 et 12 points sont des hauteurs courantes pour des documents habituels. La hauteur des caractères est indépendante de la définition de l'imprimante.
Une hauteur de 16 ou 24 points permet de mettre en valeur un titre.
Une hauteur de 6 points sera difficilement lisible, surtout si la définition de l'imprimante est mauvaise
- **Police** : La police est la forme avec laquelle le texte sera imprimé. Elle correspond à la combinaison de la fonte, du style, de la hauteur et de la couleur du caractère.
- **Définition de l'imprimante** : Chaque imprimante a une définition propre. Plus la définition est importante, plus convenable sera l'impression. Par exemple, une imprimante à 600 DPI (dots per inch) fournira des impressions plus nettes qu'une imprimante à 300 DPI. Le type d'impression (laser, jet d'encre, matricielle 8 ou 24 aiguilles,...) influe également sur la qualité d'impression. La qualité du papier également, il ne faut pas l'oublier !
- **Proportionnelle / Fixe** : Une police "fixe" présentera toujours la même largeur de caractère, quel que soit le caractère. Un "i" aura la même taille qu'un "m". Une police fixe bien connue est "Courier".
Une police "proportionnelle" utilisera une taille différente en fonction des lettres. Cette documentation est écrite dans une police proportionnelle.

Un **Etat** est le nom donné à la représentation d'une édition. WinDev vous permet de construire des états. Lors de l'exécution (ou de l'édition) de l'état, un document est créé.

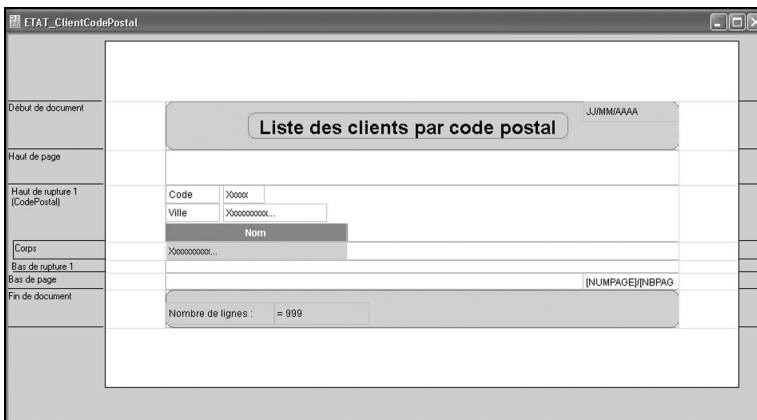
Il existe différentes formes d'états :

- Vierge
- Tableau
- Fiche
- Etiquette
- Mailing
- Formulaire
- Graphe
- Tableau avec graphe
- Tableau croisé



Un **Bloc** est le nom donné à une partie de l'état. Un état est constitué d'un ensemble de blocs. Les différents blocs d'un état sont les suivants :

- début de document
- haut de page
- haut de rupture
- corps
- bas de rupture
- bas de page
- fin de document



Il existe également d'autres blocs que nous ne détaillerons pas ici :

- bloc d'itération
- complément de corps

Ces différents blocs ne sont pas tous obligatoires. Ils peuvent être supprimés lors de la conception de l'état. Seul le bloc "Corps" est obligatoire.

Vous avez également la possibilité d'imprimer des états avec code-barres (automatiquement générés par WinDev), des états avec étiquettes, ...

Présentation

Imprimer avec l'éditeur d'états ou imprimer en WLanguage

WinDev propose deux méthodes d'impression :

- l'éditeur d'états qui permet de créer des états imprimés d'une façon "conviviale".
- par programmation en WLanguage.

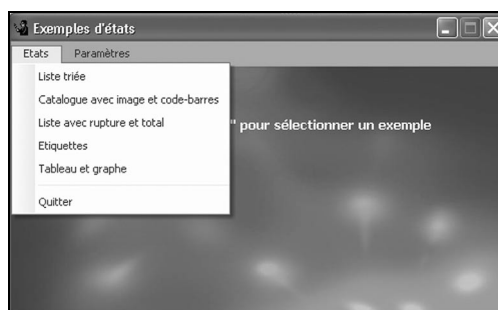
Par exemple, il est courant d'imprimer une liste dont les données proviennent d'un fichier.

Pour imprimer le contenu d'un fichier de données, l'utilisation de l'éditeur d'états est conseillée. Vous découvrirez plus loin dans ce guide comment effectuer une impression par programmation.

Nous allons tout d'abord étudier quelques exemples d'états réalisés avec l'éditeur d'états, ensuite nous verrons en détail comment les réaliser.

Quelques exemples d'états

- ▶ Sélectionnez l'option "? .. Guide d'Auto formation .. Faire des éditions". Le programme se lance. La fenêtre ci dessous apparaît.



Parcourez les différentes options du menu et testez les différentes éditions.

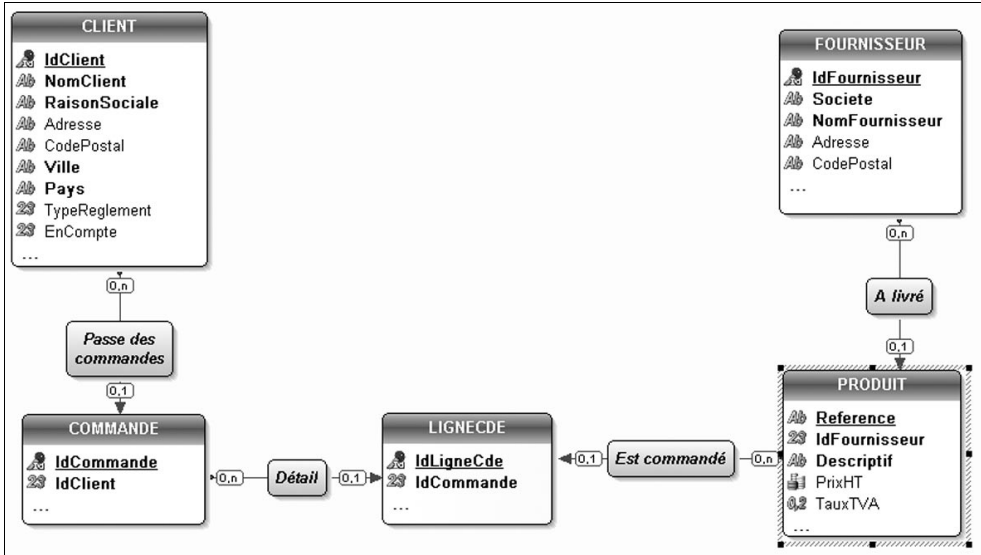
Maintenant, passons à la réalisation !

- ▶ Pour apprendre à utiliser l'éditeur d'états, nous allons créer quelques éditions dans un projet exemple préparé à cet effet.
 1. Sélectionnez l'option "Fichier .. Ouvrir un projet".
 2. Sélectionnez le sous-répertoire "AutoFormation\Exercices\Faire des éditions\" du répertoire d'installation de WinDev 12.
 3. Sélectionnez le fichier "Faire des éditions.WDP".
 Le projet se charge.

Ce projet utilise des fichiers de données. Afin de vous familiariser rapidement avec les données, nous avons pris pour exemple des fichiers simples :

- un fichier CLIENT
- un fichier COMMANDE
- un fichier LIGNECDE
- un fichier PRODUIT
- un fichier FOURNISSEUR

Ci-dessous la modélisation logique (MERISE) des données utilisées (appelé MLD) :



Les états à créer

- ▶ Pour apprendre à utiliser l'éditeur d'états, nous allons créer les exemples suivants :
 - Liste des clients triée par code postal
 - Impression des commandes par client

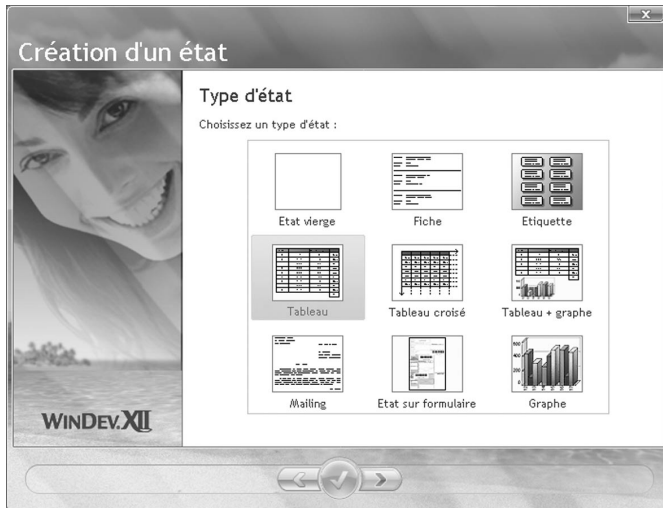
Création de votre premier état

Création de l'état : Liste des clients triée par code postal

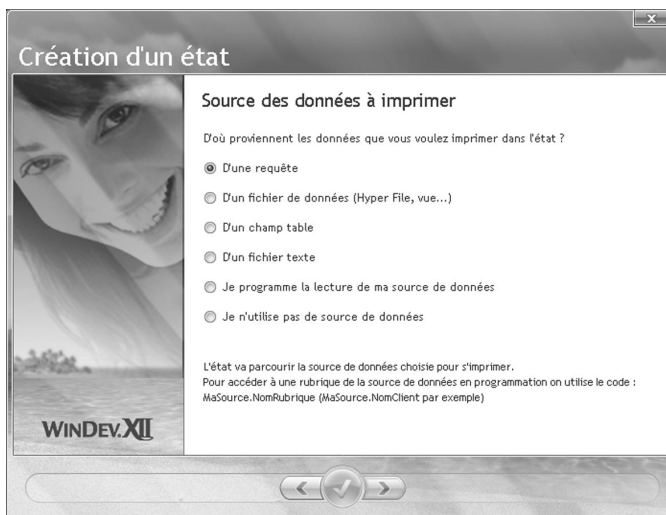
Nous allons imprimer la liste des clients triée par code postal. Pour cela, nous allons utiliser un état de type tableau, qui permet de représenter clairement une liste de données.

- ▶ Pour créer un état, sélectionnez l'option "Fichier .. Nouveau". Dans la fenêtre qui apparaît, choisissez "Etat".

La fenêtre suivante propose plusieurs présentations d'un état :

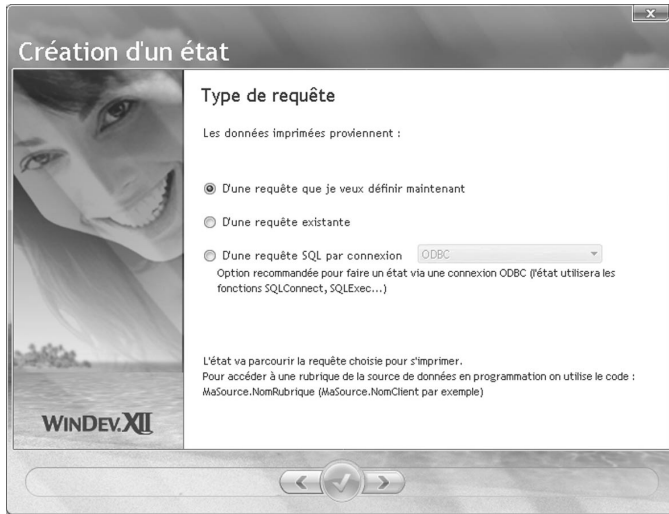


- ▶ Sélectionnez "Tableau". Passez à l'écran suivant.
- ▶ Sélectionnez la source de provenance des données de l'état.



- ▶ Sélectionnez "D'une requête". Passez à l'écran suivant.

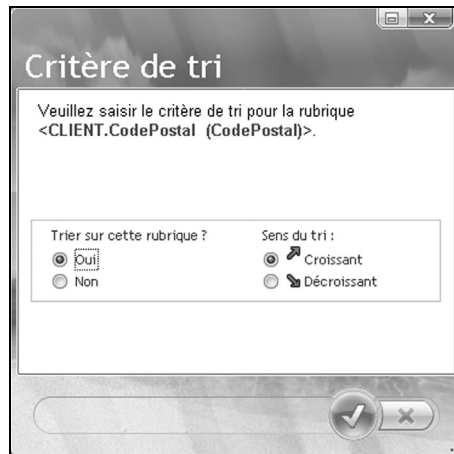
La requête n'existant pas, nous allons la créer.



- ▶ Sélectionnez "Une requête que je veux définir maintenant". Passez à l'écran suivant.
- ▶ L'assistant de description d'une requête s'affiche. Cet écran permet de sélectionner les fichiers contenant les données à imprimer. Nous allons imprimer des données provenant du fichier CLIENT.
- ▶ Cliquez sur le signe "+" à gauche du fichier CLIENT. Sélectionnez les rubriques à imprimer (par un double-clic) :
 - NomClient
 - CodePostal
 - Ville
 - idClient



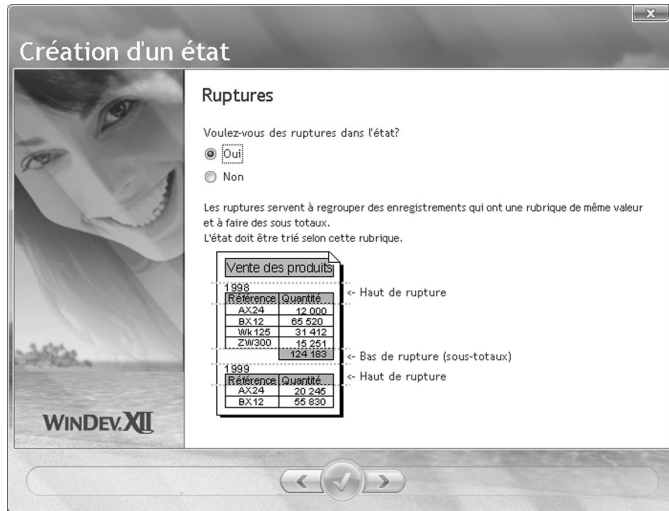
- Pour trier par code postal, il faut choisir un critère de tri :
 1. Sélectionnez la rubrique à trier : cliquez sur le "CodePostal" au milieu de l'écran.
 2. Cliquez en haut à droite sur le bouton "Trier". Choisissez l'option "Trier sur la rubrique sélectionnée".



3. Choisissez "Ordre croissant" et validez. Une flèche rouge apparaît à droite de la rubrique "CodePostal" ainsi que le numéro "01". Cette flèche indique qu'un tri croissant est fait sur cette rubrique. Le numéro "01" indique que ce tri sera le premier tri effectué.

- Nous allons valider cet écran. . Passez à l'écran suivant.

L'assistant demande d'indiquer s'il y a une rupture.



Qu'est-ce qu'une rupture?

Une **Rupture** est une opération qui consiste à regrouper des enregistrements (ou lignes) suivant un ou plusieurs critères. Attention, les enregistrements (ou lignes) sont bien entendu imprimés.

Une rupture est OBLIGATOIREMENT liée à un tri.



Notes

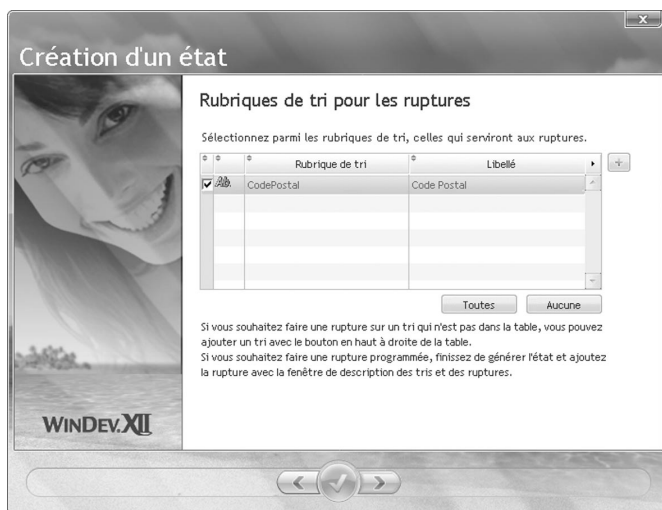
Notez la présence d'un en-tête de rupture et d'une fin de rupture.

Les informations communes aux différentes lignes sont présentes dans l'en-tête de rupture.

Les totaux, compteurs, ... sont présents dans la fin de rupture.

Cette rupture permet de regrouper une série d'enregistrements par rapport à un même critère. Dans notre exemple, la rupture concerne le code postal et permet de regrouper tous les clients d'une même commune.

► Répondez "Oui". Passez à l'écran suivant.



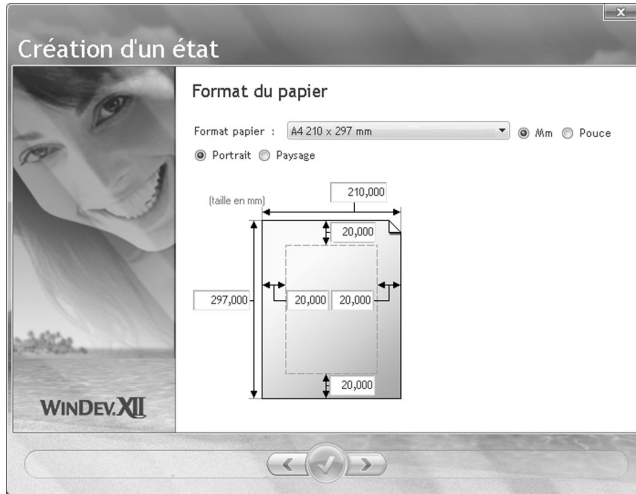
- ▶ La rupture s'effectue sur le code postal. Passez à l'écran suivant.
- ▶ Vous allez ensuite indiquer dans quel ordre sont imprimées les rubriques et leur répartition dans les différents blocs.
Dans notre exemple, le code postal et la ville seront dans le bloc "haut de rupture 1" et le nom dans le corps.
Dans la colonne de droite, sélectionnez "haut de rupture 1" pour "Ville".
Décochez la rubrique "idClient".



Les flèches bleues à droite permettent de modifier l'ordre des rubriques. Vous pouvez par exemple afficher la ville avant le nom.

► Passez à l'écran suivant.

Cet écran permet de définir la mise en page de l'état.



Nous garderons les valeurs par défaut avec l'orientation "Portrait".

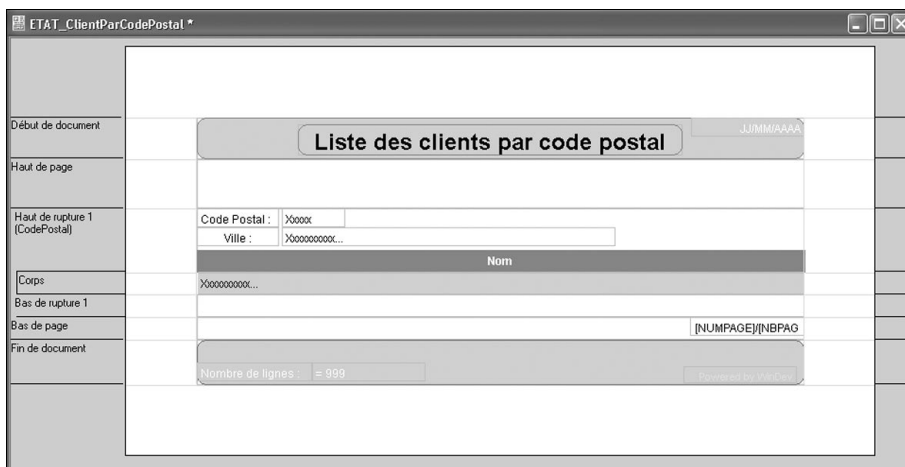
Lors du choix des marges d'impression, n'oubliez pas de tenir compte des marges physiques des imprimantes. Les marges physiques sont les marges réservées par l'imprimante dans lesquelles il n'est pas possible d'imprimer. De plus, les marges physiques diffèrent en fonction des modèles d'imprimante.

► Passez à l'écran suivant.


► Sélectionnez par exemple le gabarit "Media Center". Passez à l'écran suivant.



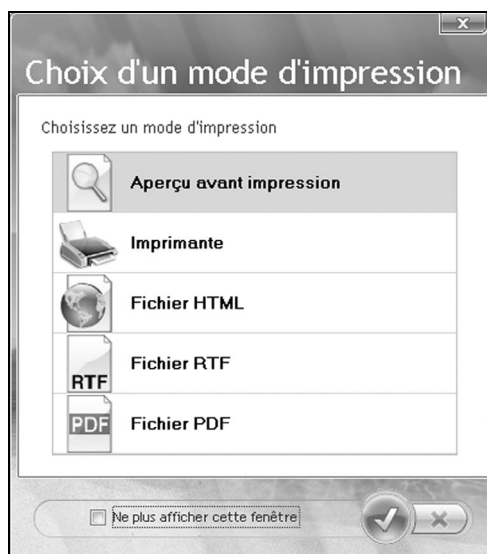
- ▶ Il ne nous reste plus qu'à donner un nom et un libellé à l'état et à le sauvegarder.
 1. Saisissez le nom "ETAT_ClientParCodePostal".
 2. Saisissez le titre "Liste des clients par code postal".
 3. Validez.



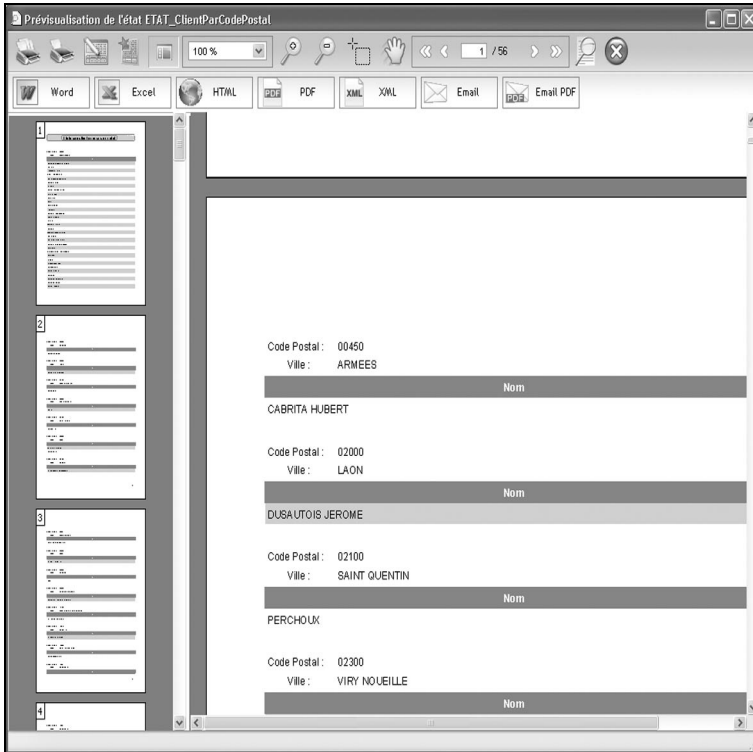
4. Enregistrez l'état. L'état est terminé. Nous allons immédiatement le tester pour voir le résultat.

- ▶ Exécutez cet état en cliquant sur l'icone "GO"  de la barre d'icônes.


La destination de l'impression peut être au choix :




- Choisissez "Aperçu avant impression" et cliquez sur le bouton "OK".
L'exécution de l'état s'affiche à l'écran.



Vous pouvez :

- Directement imprimer la page en cours ou la totalité du document en cliquant sur les imprimantes .

- Sélectionner un niveau de zoom plus ou moins important avec les icônes .
- Enregistrer l'état sous forme d'un document Word (au format RTF).
- Enregistrer l'état au format HTML.
- Enregistrer l'état au format PDF.
- Enregistrer l'état au format XML.
- Créer un e-mail avec l'état au format HTML en corps du message.
- Créer un e-mail avec l'état au format PDF en document lié.

LEÇON 7.2. ETATS STATISTIQUES

Ce que vous allez apprendre dans cette leçon ...

- Lier plusieurs fichiers de données dans un état.
- Faire des regroupements dans un état.



Durée estimée : 30mn

Présentation

Nous allons maintenant éditer la liste des commandes par client. Cette liste sera triée sur le nom du client. Pour chaque client, nous imprimons la commande et le détail de la commande (lignes) avec un sous-total pour chaque commande. Nous cherchons à obtenir un résultat semblable à la représentation ci-dessous :

Prévisualisation de l'état Etat_CommandesParClient

100 %

Word Excel HTML PDF XML XML Email Email PDF

Liste des commandes par client

Nom : AIC CARLE Client enregistré
 Raison Sociale : AEER
 Code Postal : 65000
 Ville : BARBAZAN DEBAT
 Commande n° : 350

Date Commande	Descriptif	Ref.	Qté	Montant
17/09/1995	Désign auto34005M-F	34005M-F	1	
17/09/1995	Désign auto344791 CLA	344791 CLA	3	

Nom : ALEMBIK NATHALIE Client enregistré
 Raison Sociale : TDF ES
 Code Postal : 36000
 Ville : CHATEAUROUX
 Commande n° : 976

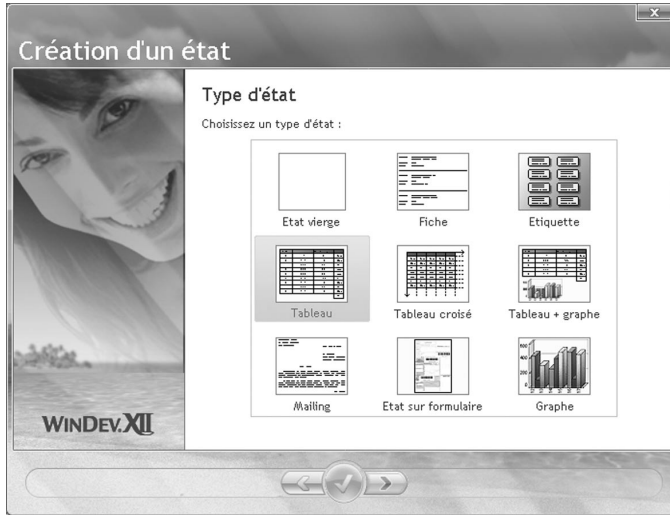
Date Commande	Descriptif	Ref.	Qté	Montant
13/09/1997	Général FAM452-0021 SIMON	004150MON	2	
13/09/1997	Général FAM195-0021 75BCV	002175BCV	3	
13/09/1997	121g foie de motue fumé	000071TRH	2	
13/09/1997	Désign auto344095-I	344095-I	2	

L'état que nous allons réaliser utilise plusieurs fichiers de données comme source.

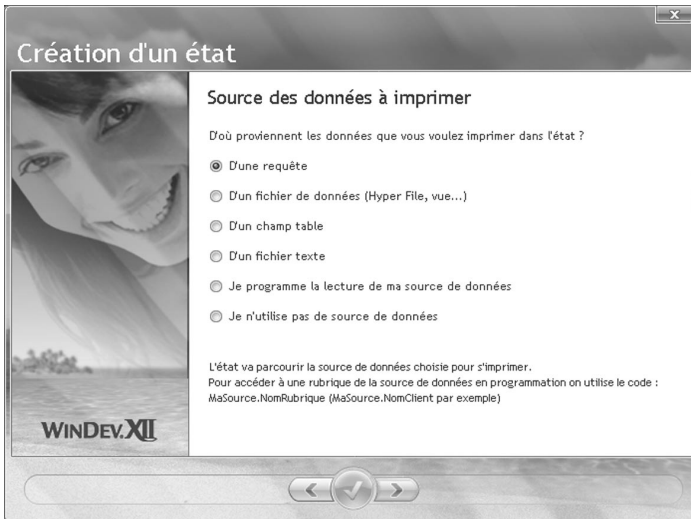
Etape par étape

- Ouvrez le projet "Faire des éditions.WDP". Ce projet est présent dans le sous-répertoire "Auto-Formation\Exercices\Faire des éditions\" du répertoire d'installation de WinDev 12.

- Pour créer un nouvel état, sélectionnez l'option "Fichier .. Nouveau .. Etat".

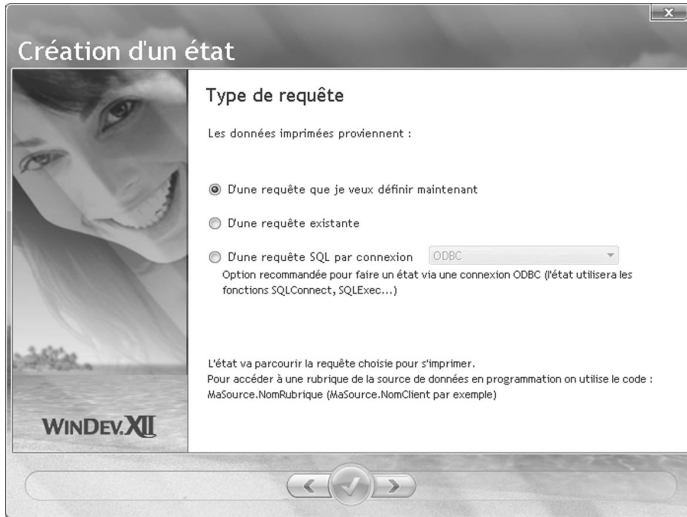


- Sélectionnez "Tableau" et passez à l'écran suivant.
- Sélectionnez "D'une requête" et passez à l'écran suivant.



La requête n'existant pas, nous allons la créer.

► Sélectionnez "Une requête que je veux définir maintenant". Passez à l'écran suivant.



Nous allons imprimer des données provenant des fichiers CLIENT, COMMANDE, PRODUIT et LIGNECDE.

► Sélectionnez les rubriques suivantes à imprimer (par un double-clic) :

- Le fichier CLIENT :
 - NomClient
 - RaisonSociale
 - CodePostal
 - Ville

- Le fichier COMMANDE :
 - IdCommande
 - DateCommande

- Le fichier PRODUIT :
 - Descriptif

- Le fichier LIGNECDE :
 - Référence
 - QteCommandee
 - Montant



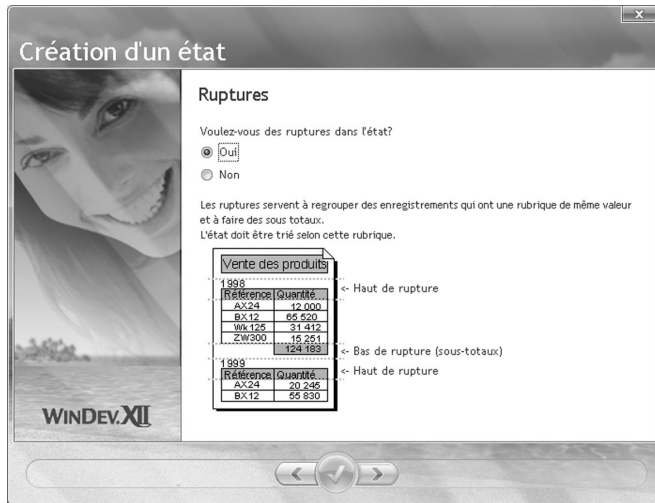
Nous allons trier les résultats par nom de client et par numéro de commande.

- ▶ Dans la liste au centre, sélectionnez la première rubrique sur laquelle vous désirez effectuer un tri :
 1. Cliquez sur la rubrique "NomClient". Dans le menu contextuel (clic droit de la souris), sélectionnez "Trier la rubrique .. Trier par ordre croissant".
 2. Cliquez ensuite sur la rubrique "IdCommande". Dans le menu contextuel (clic droit de la souris), sélectionnez "Trier la rubrique .. Trier par ordre croissant".
 3. Validez.

Une flèche rouge apparaît à droite des rubriques "NomClient" et "IdCommande". Le chiffre situé à côté de chaque flèche indique l'ordre de tri réalisé.



- ▶ Nous allons valider cet écran. Passez à l'écran suivant.
- ▶ Nous allons maintenant indiquer les ruptures. Choisissez "Oui". Passez à l'écran suivant.



Par défaut, une rupture est proposée sur les rubriques "NomClient" et "IdCommande", car ce sont les critères de tri déjà définis.

- ▶ Désélectionnez la rubrique NomClient (cliquez sur la coche). Passez à l'écran suivant.



Vous allez ensuite définir la répartition des rubriques.

Les informations concernant le client et la commande seront dans le haut de rupture 1. Le corps de l'édition concerne le détail de chaque commande.

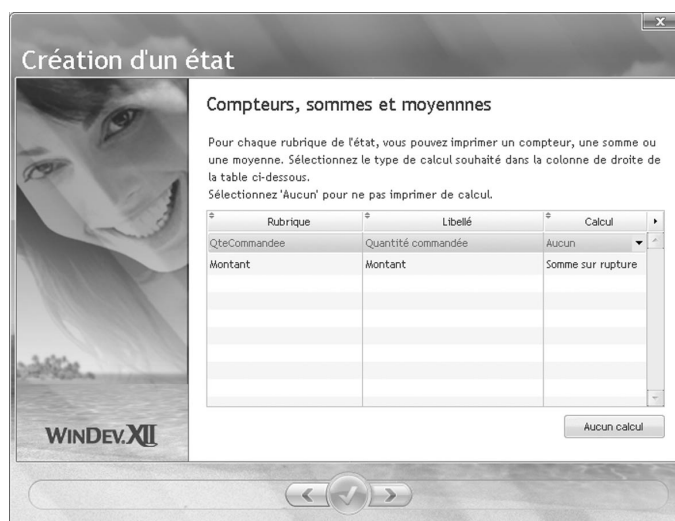
- Répartissez les rubriques comme indiqué dans l'image ci-dessous :



- Passez à l'écran suivant

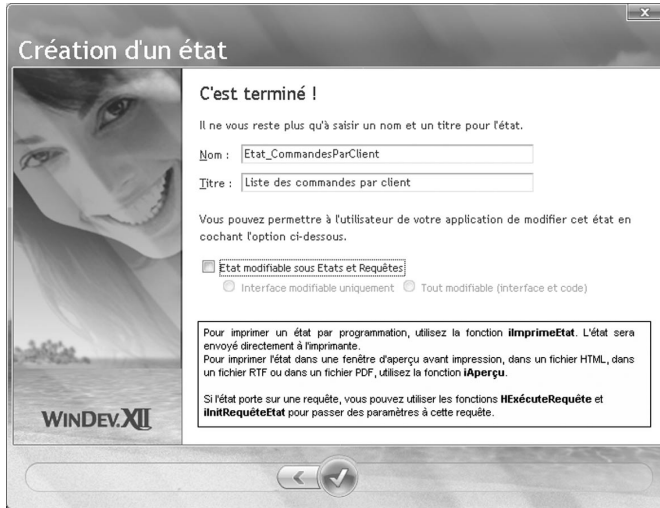
Il faut maintenant indiquer s'il y a des sommes ou des comptages dans l'état. Nous allons additionner le montant des lignes de commandes.

Cette somme sera placée dans une rupture. Nous allons conserver uniquement la somme sur le montant. Vous allez indiquer qu'il n'y a aucun calcul sur la quantité.

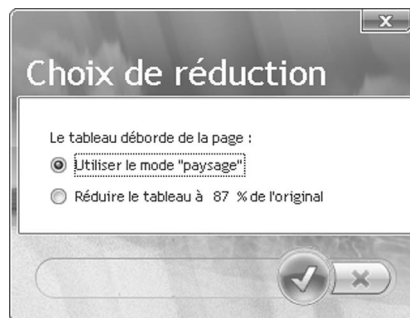


- Passez à l'écran suivant.
- Nous conserverons les paramètres par défaut pour le format de la page.

- ▶ Passez à l'écran suivant.
- ▶ Choisissez un gabarit et passez à l'écran suivant
- ▶ Donnez un nom à l'état pour terminer : "ETAT_CommandesParClient" par exemple.

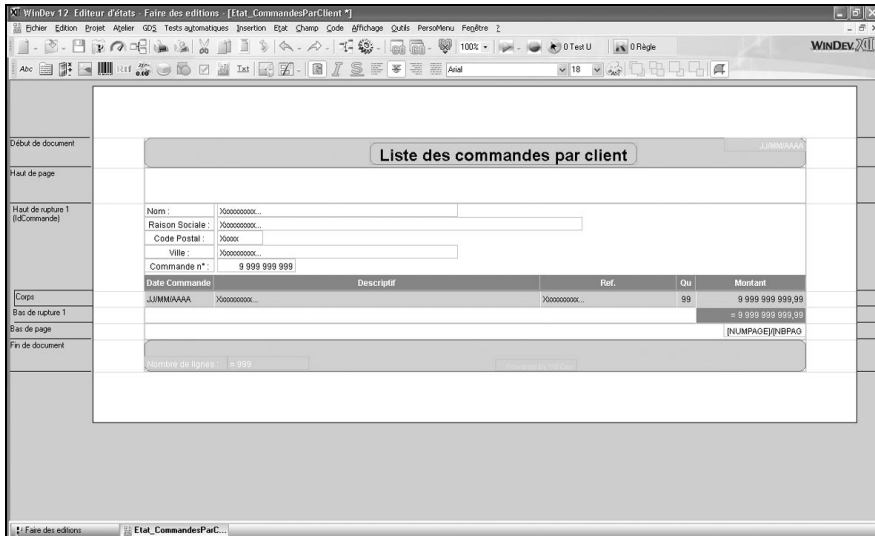


- ▶ Validez en cliquant sur le bouton "Terminer". L'état étant trop grand, l'éditeur d'états propose de changer le format de l'état ou de le réduire.



- ▶ Sélectionnez le mode paysage et validez.

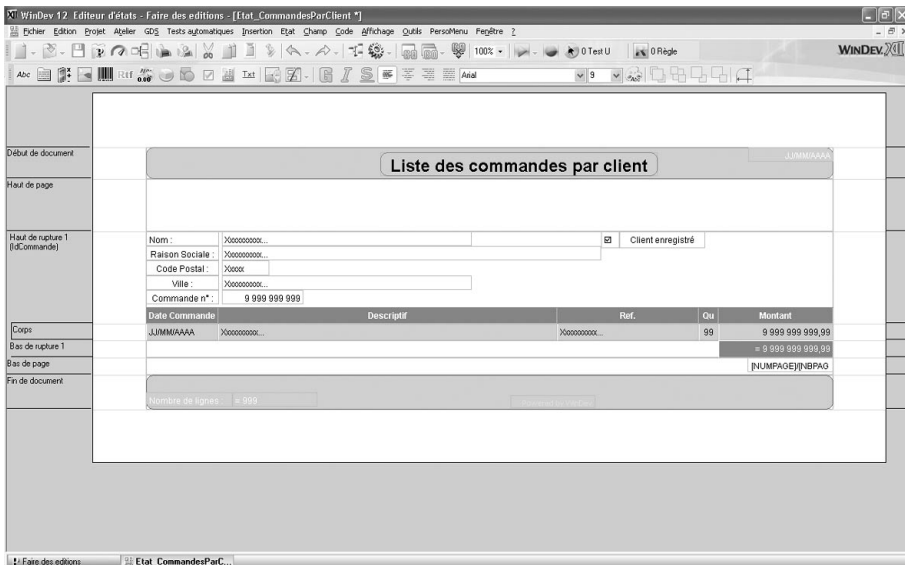
L'état apparaît sous l'éditeur :



- ▶ Modifiez si nécessaire la mise en forme des champs.
- ▶ Nous allons maintenant ajouter un interrupteur dans cet état.
Les interrupteurs dans les états permettent notamment de reproduire des formulaires (par exemple, les cases à cocher "Monsieur", "Madame", ...).
Une rubrique du fichier "CLIENT" indique si le client possède un compte chez le fournisseur (en l'occurrence chez nous). Cette rubrique peut prendre la valeur "Oui" ou "Non".
- ▶ Sélectionnez l'option "Insertion .. Nouveau champ .. Interrupteur". Placez l'interrupteur sur l'état dans le bloc "Haut de rupture 1".
- ▶ Dans l'onglet "Général" des propriétés de l'interrupteur (clic droit, "Description") :
 1. Saisissez le nom du champ, "INT_ENCOMPTE".
 2. Cliquez sur le bouton "Autres..." et sélectionnez la rubrique liée "EnCompte" du fichier "CLIENT".
 3. Validez.



- ▶ Ajoutez un libellé que vous positionnerez à côté de l'interrupteur et saisissez le texte "Client Enregistré".
- ▶ Voici l'état tel qu'il apparaît sous l'éditeur :



- ▶ Enregistrez l'état, et testez-le avec le bouton .

Prévisualisation de l'état Etat_CommandesParClient

100 % 1 / 70

Word Excel HTML PDF XML XML Email Email PDF

Liste des commandes par client

Nom : A/C CARLE Client enregistré
 Raison Sociale : AEER
 Code Postal : 65990
 Ville : BARBAZAN DEBAT
 Commande n° : 358

Date Commande	Descriptif	Ref.	Qté	Montant
17/09/1995	Désign auto340051M-F	340051M-F	1	
17/09/1995	Désign auto344791 CLA	344791 CLA	3	

Nom : ALEMBIK NATHALIE Client enregistré
 Raison Sociale : TDFES
 Code Postal : 38000
 Ville : CHATEAULOUX
 Commande n° : 976

Date Commande	Descriptif	Ref.	Qté	Montant
13/09/1997	Général FAM452-004158MON	004158MON	2	
13/09/1997	Général FAM195-002175BCW	002175BCW	3	
13/09/1997	121g foie de morue fumé	000071THH	2	
13/09/1997	Désign auto344089i-1	344089i-1	2	

LEÇON 7.3. ETATS TABLEAU CROISÉ

Ce que vous allez apprendre dans cette leçon ...

- Etape par étape, la création d'un état tableau croisé.



Durée estimée : 20mn

Présentation

L'éditeur d'états offre également la possibilité de créer des états de type "Tableau Croisé". L'état est alors constitué uniquement d'un tableau à deux entrées.

Exemple :

Contrairement aux autres types d'états, l'état "Tableau croisé" est toujours basé sur une **requête intégrée**. Cette requête est créée grâce à l'assistant de création de l'état "Tableau croisé". Vous pourrez par ailleurs modifier cette requête une fois votre état créé.

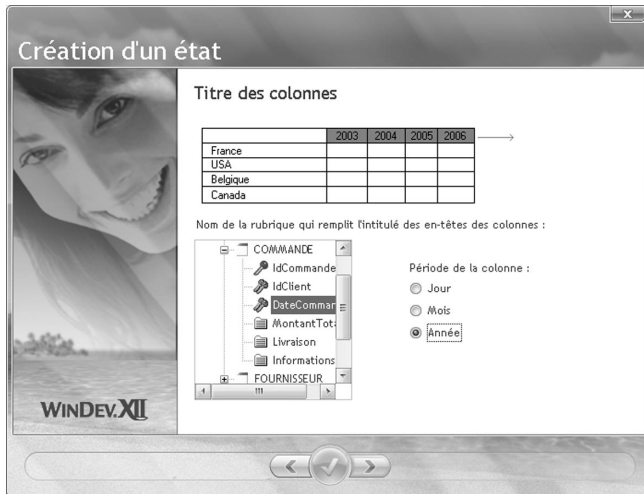
Etape par étape

Nous allons réaliser un état de synthèse sur le chiffre d'affaires par client et par année. Ce tableau peut être représenté de plusieurs façons :

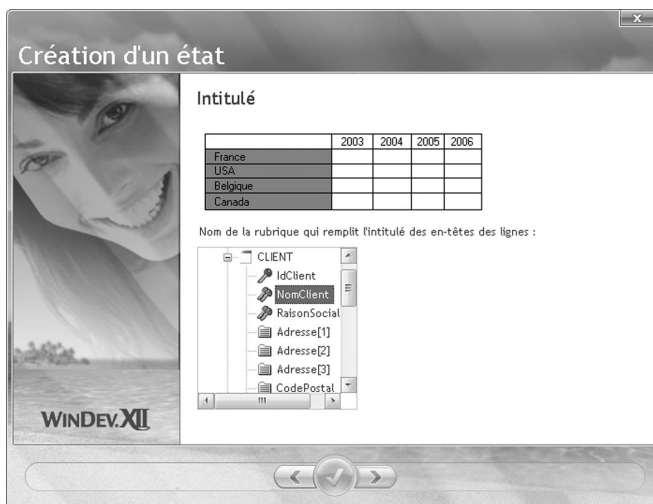
- en indiquant, pour chaque client, les différentes années considérées et le chiffre d'affaires correspondant.
 - en réalisant un tableau croisé, de façon à trier les résultats à la fois par client (vertical) et par année (horizontal). C'est cette méthode que nous allons utiliser.
- ▶ Ouvrez si nécessaire le projet "Faire des éditions.WDP". Ce projet est présent dans le sous-répertoire "AutoFormation\Exercices\Faire des éditions\" du répertoire d'installation de WinDev 12.
 - ▶ Sélectionnez dans le menu "Fichier .. Nouveau .. Etat".
 - ▶ Sélectionnez "Tableau Croisé". Passez à l'écran suivant.

L'assistant vous propose alors de choisir la rubrique qui servira de source de données pour les entêtes de colonnes. Vous pouvez ainsi définir la première entrée (ligne) de votre tableau croisé.

- ▶ Sélectionnez la rubrique "DateCommande" dans le fichier "COMMANDE".
L'assistant vous propose alors trois choix (jour, mois et année).
Choisissez "Année". Passez à l'écran suivant.



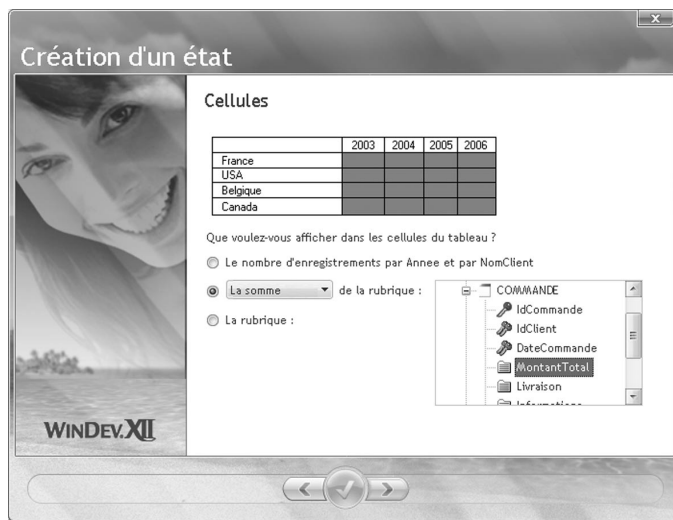
- ▶ L'assistant vous demande si vous souhaitez "borner" les dates. Ne cochez rien, nous allons prendre toutes les commandes. Passez à l'écran suivant.
- ▶ L'assistant vous demande maintenant de choisir la rubrique qui servira de source de données pour les lignes. En fait, vous allez choisir la seconde entrée (colonne) de votre tableau croisé. Sélectionnez la rubrique "NOMCLIENT" dans le fichier "CLIENT".



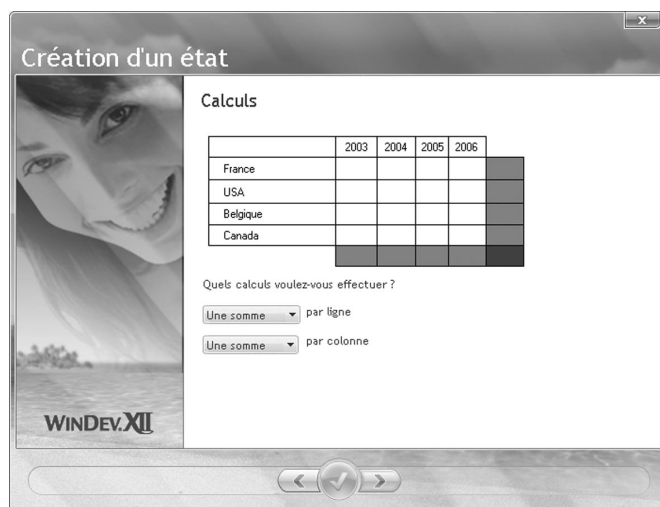
- ▶ Passez à l'écran suivant.
L'assistant vous demande maintenant de sélectionner le type d'information à afficher dans les

cellules du tableau croisé. Comme nous voulons afficher le Chiffre d'Affaires par client et par date, nous allons demander une somme des montants totaux de chaque commande passée pour chaque client chaque année.

- Sélectionnez "La somme de la rubrique :". Dans la liste qui apparaît sur le côté, choisissez la rubrique "MONTANTTOTAL" du fichier "COMMANDE" :

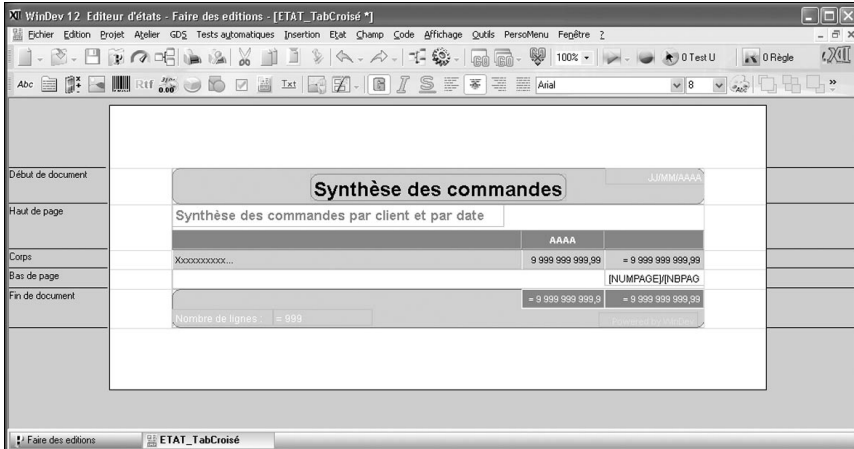


- Passez à l'écran suivant. L'assistant vous propose de calculer une somme par ligne et une somme par colonne. Nous allons accepter ce calcul :




- Passez à l'écran suivant. Ne changez rien à la mise en page. Passez à l'écran suivant.
- Sélectionnez un gabarit (par exemple "Media Center"). Passez à l'écran suivant.

- ▶ Vous allez donner un nom à cet état :
 1. Saisissez le nom : "ETAT_TabCroisé".
 2. Saisissez la description : "Synthèse des commandes par client et par date".
 3. Validez. L'état étant trop grand, l'éditeur d'états propose de changer le format de l'état ou de le réduire : sélectionnez "Réduire le tableau". L'état apparaît sous l'éditeur d'états.



- ▶ Modifiez si nécessaire la taille des colonnes pour obtenir la même mise en page que ci-dessus.

Enregistrez l'état. Il ne reste plus qu'à tester en cliquant sur l'icone .



Vous venez de réaliser facilement un état tableau croisé. Ce type d'état peut bien entendu être uti-

lisé pour d'autres applications, notamment pour effectuer des comparatifs, des statistiques, ...

LEÇON 7.4. CHAMPS SPÉCIFIQUES

Ce que vous allez apprendre dans cette leçon ...

- La création d'un lien dans un état.
- La création d'un champ cliquable (dans l'aperçu)



Durée estimée : 20mn

Présentation

L'éditeur d'états de WinDev propose deux fonctionnalités originales au niveau des champs :

- **Les champs Lien**, utilisables aussi bien dans les états générés en HTML que dans les états affichés dans l'aperçu avant impression.

Les champs lien permettent d'ouvrir directement un site à partir d'un état. Il est ainsi possible d'imaginer un récapitulatif de commande envoyé par e-mail. Cet état peut contenir un lien vers le site de la société, un lien pour envoyer un e-mail à la société, ...

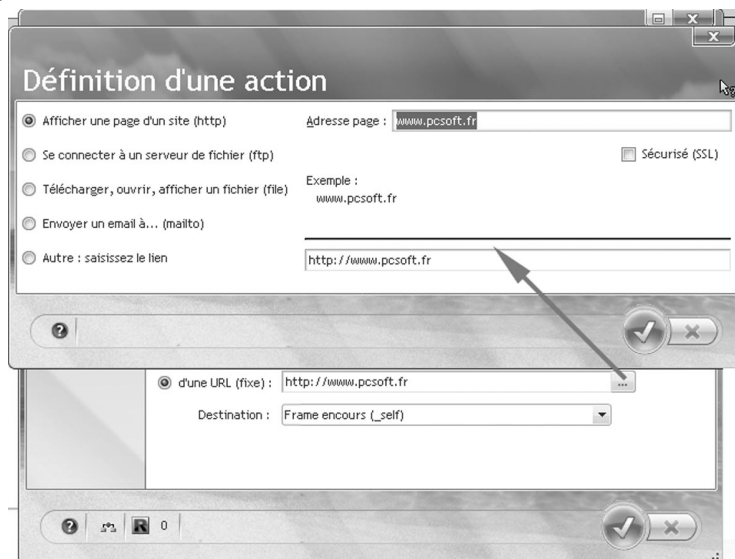
- **Les champs clicables**, utilisables uniquement dans l'aperçu avant impression.

Dans ce cas, le clic sur un des champs de l'état déclenche un traitement spécifique, par exemple l'affichage de la fiche complète du client sélectionné.

Dans l'aperçu avant impression, ces deux types de champs sont facilement repérables grâce à des petites étoiles.

Manipulations pour créer un champ Lien

- ▶ Ouvrez si nécessaire le projet "Faire des éditions.WDP". Ce projet se trouve dans le sous-répertoire "Autoformation\Exercices\Faire des éditions" de WinDev.
- ▶ Ouvrez l'état "ETAT_FactureAvecLien.WDE".
- ▶ Affichez la description du champ "Lien". Ce lien permet à l'internaute de visiter le site de la société qui le facture.



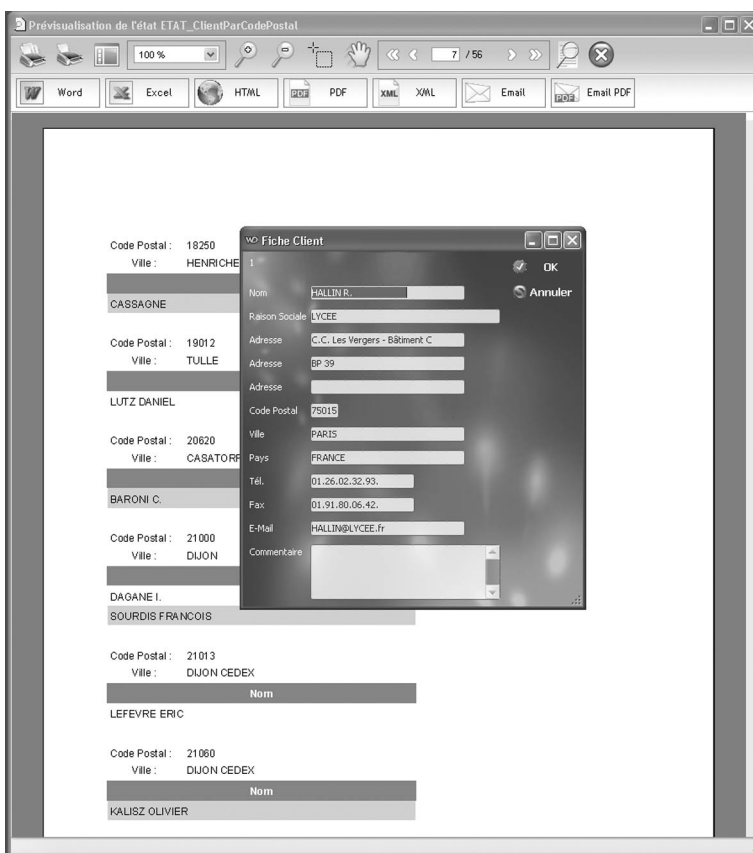
Remarque : Un champ Lien dans un état peut afficher le contenu d'une rubrique ou une URL. Dans ce cas, il est possible de :

- afficher une page d'un site
- se connecter à un serveur FTP
- télécharger, ouvrir, afficher un fichier
- envoyer un e-mail

- ▶ Testez l'état. Ce lien fonctionne en aperçu avant impression (une étoile près du lien indique que ce lien est cliquable) ou en mode HTML.

Manipulations pour créer un champ cliquable

- ▶ Ouvrez si nécessaire le projet "Faire des éditions.WDP". Ce projet se trouve dans le sous-répertoire "Autoformation\Exercices\Faire des éditions" de WinDev 12.
- ▶ Ouvrez l'état créé précédemment "ETAT_ClientParCodePostal" (ou l'état "ETAT_ClientCodePostal" si vous ne l'avez pas créé). Cet état liste les clients par code postal. Nous allons permettre à l'utilisateur de cliquer sur le nom du client pour voir le détail des informations le concernant dans une fenêtre fiche.



- ▶ Le projet contient déjà la fenêtre à ouvrir : "FEN_Fiche_Client". Ouvrez cette fenêtre sous l'éditeur de fenêtres (option "Fichier .. Ouvrir"). Cette fenêtre a été créée par le RAD. Son code de déclaration des globales a été modifié pour permettre d'afficher le client dont l'identifiant est passé en paramètre.

```

CAS ~= "Modif"
  SI IdClient<>0 ALORS
    HLitRecherchePremier (CLIENT, IdClient, IdClient)
  FIN
SI HNbEnr (CLIENT)=0 OU HEnDehors (CLIENT) ALORS
  // ouvre une boîte de dialogue pour informer l'utilisateur
  Info("Le fichier ne contient aucun enregistrement.")
  // ferme la fenêtre
  Ferme
FIN

// On remplit les champs
FichierVersEcran()
FIN

```

- Pour ouvrir directement la fenêtre "FEN_Fiche_Client" depuis l'aperçu avant impression :
1. Sous l'éditeur d'états, cliquez sur le champ "RUB_NomClient".
 2. Affichez le code de ce champ (option "Code" du menu contextuel).
 3. Saisissez le code suivant dans le traitement "Clic sur RUB_NomClient":

```
Ouvre (FEN_Fiche_Client, "Modif", MaSource.IdClient)
```

Ce code permet de :

- ouvrir la fenêtre "FEN_Fiche_Client" en mode Modification.
 - passer en paramètre à la fenêtre l'identifiant du client cliqué (cet identifiant est présent dans la requête intégrée à l'état).
4. Enregistrez et testez votre état.

LEÇON 7.5. EXÉCUTION DES ETATS

Ce que vous allez apprendre dans cette leçon ...

- Exécution des états en WLangage.



Durée estimée : 10mn

Exécuter les états en WLangage

Vous avez créé des états qui ont été testés depuis l'éditeur. Nous allons voir comment lancer l'exécution d'un état depuis un programme.



Notes

WinDev permet d'exécuter directement les états depuis une application grâce aux fonctions du WLangage. Sachez que vous avez également la possibilité de rajouter du code pour effectuer des traitements spécifiques à l'intérieur de chaque bloc de l'état (par exemple : faire un calcul, afficher une boîte de dialogue ...).

La syntaxe générale d'exécution d'un état est la suivante :

```
// prévisualisation de l'état "Etiquetteclient"
iAperçu(100)
// Impression de l'état
iImprimeEtat(Etiquetteclient)
```

En exécutant ce code, un écran d'aperçu vous sera proposé. Pour imprimer directement l'état sans passer par une fenêtre de prévisualisation, supprimez la ligne :

```
iAperçu(100)
```

Fonction **iAperçu**

La fonction **iAperçu** permet de sélectionner le mode d'édition. La syntaxe de la fonction **iAperçu** est la suivante :

```
iAperçu(<format de sortie>)
```

Le paramètre <format de sortie> peut prendre différentes valeurs :

- *ilprimante* : pour une sortie directe sur l'imprimante courante
- *iPage* : pour afficher à l'écran la page complète
- *i2Page* : pour afficher 2 pages à l'écran
- *i100* : pour afficher la page en zoom 100%
- *iFichier* : pour imprimer l'état dans un fichier texte
- *iHTML* : pour imprimer l'état au format HTML
- *iHTMLSansCSS* : pour imprimer l'état au format HTML sans feuille de styles
- *iPDF* : pour imprimer l'état au format PDF
- *iRTF* : pour imprimer l'état au format RTF
- *iXML* : pour imprimer l'état dans un fichier XML
- *iFax* : pour une sortie directe sur un fax.

Pour plus de détails sur la fonction **iAperçu**, consultez l'aide en ligne (mot-clé : "iAperçu").

Fonction `ilImprimeEtat`

La fonction `ilImprimeEtat` permet d'exécuter un état. La syntaxe de cette fonction est la suivante :

```
ilImprimeEtat(<Nom de l'état>)
```

Remarque

D'autres syntaxes sont possibles pour ces deux fonctions, consultez l'aide en ligne pour plus de détails (mots-clés : "Impression d'un état", "iAperçu", "ilImprimeEtat").



Notes

Offrez la possibilité aux utilisateurs de vos applications de créer eux-mêmes leurs états et requêtes (ou de modifier des états ou des requêtes que vous mettez à leur disposition) !

Il suffit de diffuser avec votre application le logiciel "Etats et Requêtes". Pour plus de détails, consultez la leçon suivante.

LEÇON 7.6. DIFFUSEZ "ETATS ET REQUÊTES" AVEC VOS APPLICATIONS

Ce que vous allez apprendre dans cette leçon ...

- A quoi sert le logiciel "Etats et Requêtes".
- Comment diffuser le logiciel "Etats et Requêtes".
- Comment utiliser le logiciel "Etats et Requêtes".



Durée estimée : 10mn

WinDev 12 version "Express" :

Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

Présentation du logiciel "Etats et Requêtes"

Le logiciel "Etats et Requêtes" permet aux utilisateurs de vos applications de modifier et de créer à leur convenance les états de vos applications.

Un utilisateur trouve qu'il manque une information sur l'état fourni en standard ? Avec le logiciel "Etat et Requêtes", il peut ajouter l'information directement depuis l'aperçu avant impression. Cette modification peut être intéressante pour d'autres utilisateurs ? L'état peut être mis à disposition de tous les utilisateurs utilisant l'application en réseau.




Notes

Le logiciel "Etats et Requêtes" remplace le logiciel "WDETAT Utilisateur". Ce logiciel est désormais gratuit et librement redistribuable avec vos applications créées avec WinDev. Consultez la licence pour plus de détails sur les conditions de redistribution.

Lancement du logiciel Etats et Requêtes

Pour que les utilisateurs finals personnalisent les états de votre application ou créent leurs propres états et leurs propres requêtes, il suffit de lancer le logiciel Etats et Requêtes.

Pour permettre à vos utilisateurs finals de lancer le logiciel Etats et Requêtes :

- soit affichez vos états sous la fenêtre d'aperçu avant impression. Les utilisateurs finals pourront lancer le logiciel Etats et Requêtes en cliquant l'icone .
- soit ajoutez le menu "?" dans la fenêtre principale de votre application (option "Fenêtres .. Menu principal .. Ajouter le menu '?'"). Dans l'assistant de création de ce menu, cochez les options "Créer, modifier un état" et "Créer, modifier une requête".
- soit utilisez la fonction **LanceEtatsEtRequêtes** dans votre application.

Diffuser le logiciel "Etats et Requêtes" avec votre application

Pour diffuser le logiciel "Etats et Requêtes" avec votre propre application WinDev, il est nécessaire de :

- indiquer au niveau du projet que l'application autorise le lancement du logiciel "Etats et Requêtes".
- définir si nécessaire au niveau de l'analyse les fichiers et rubriques qui seront utilisables avec le logiciel "Etats et Requêtes".
- définir les états et les requêtes modifiables sous le logiciel "Etats et Requêtes".
- créer l'exécutable et le programme d'installation intégrant le logiciel "Etats et Requêtes".

Nous allons détailler ces différentes étapes. Vous pouvez réaliser les manipulations suivantes sur le projet "Faire des éditions.WDP" que nous avons manipulé jusqu'à présent.

Paramétrage du projet

Pour utiliser le logiciel "Etats et Requêtes" à partir de votre application, il est nécessaire de configurer le projet associé à votre application.

- ▶ Ouvrez si nécessaire le projet "Faire des éditions.WDP". Ce projet se trouve dans le sous-répertoire "Autoformation\Exercices\Faire des éditions" de WinDev.
- ▶ Affichez l'onglet "Avancé" de la description du projet (option "Projet .. Description du projet").
- ▶ Cochez l'option "Autoriser le lancement de "Etats et Requêtes"". Validez.

Paramétrage de l'analyse

Nous allons maintenant ouvrir l'analyse pour paramétrer les fichiers et rubriques utilisables sous le logiciel "Etats et Requêtes".

- ▶ Ouvrez l'analyse (option "Projet .. Charger l'analyse").

Par défaut, tous les fichiers et toutes leurs rubriques sont visibles et manipulables avec le logiciel "Etats et Requêtes". Si votre application contient des informations sensibles (mots de passe, ...) vous pouvez rendre ces fichiers ou rubriques invisibles sous le logiciel "Etats et Requêtes".



Notes

A partir du logiciel "Etats et Requêtes", l'utilisateur ne peut pas ajouter, modifier ou supprimer des enregistrements (les fonctions **HAjoute**, **HModifie** et **HSupprime** sont interdites). Il n'est également pas possible de créer ou de re-créeer des fichiers de données (les fonctions **HCréation** et **HCréationSilnexistent** sont interdites).

Pour définir un fichier invisible sous le logiciel "Etats et Requêtes" :

1. Sous l'éditeur d'analyses, sélectionnez le fichier voulu.
2. Affichez la fenêtre de description du fichier (option "Structure de fichiers .. Description" ou option "Description du fichier" du menu contextuel).
3. Dans l'onglet "Etats et Requêtes", décochez l'option "Visible par l'utilisateur final (Etats et Requêtes)".
4. Validez.

Pour définir une rubrique invisible sous le logiciel "Etats et Requêtes" :

1. Sous l'éditeur d'analyses, sélectionnez le fichier voulu.
2. Affichez la fenêtre de description du fichier (option "Structure de fichiers .. Rubriques" ou option "Description des rubriques" du menu contextuel).
3. Sélectionnez dans la table, la rubrique voulue.
4. Dans l'onglet "Etats et Requêtes", décochez l'option "Visible par l'utilisateur final (Etats et Requêtes)".
5. Validez.

- ▶ Pour prendre en compte les modifications effectuées dans l'analyse, il est nécessaire de régénérer l'analyse (option "Analyse .. Génération").

Paramétrage des états

Lors de la création d'un état, il est possible de spécifier si cet état est modifiable sous le logiciel "Etats et Requêtes".

Pour rendre un état de votre application modifiable sous le logiciel "Etats et Requêtes" :

1. Ouvrez l'état sous l'éditeur d'états.
2. Affichez la description de l'état (option "Etat .. Description").
3. Dans l'onglet "Détail", cochez "Etat modifiable sous Etats et Requêtes".
4. Cochez :
 - l'option "Interface modifiable uniquement" pour que l'interface de l'état uniquement soit modifiable sous Etats et Requêtes.
 - l'option "Tout modifiable (interface et code)" pour que l'état soit entièrement modifiable sous Etats et Requêtes.
5. Validez et enregistrez l'état.

Paramétrage des requêtes

Par défaut, une requête n'est pas modifiable sous le logiciel "Etats et Requêtes". Lors de la création d'une requête, il est possible de spécifier si cette requête est modifiable sous le logiciel "Etats et Requêtes".

Pour rendre une requête de votre application modifiable sous le logiciel "Etats et Requêtes" :

1. Ouvrez la requête sous l'éditeur de requêtes.
2. Affichez la description de la requête (option "Requête .. Description de la requête").
3. Cliquez sur le bouton "Avancé".
4. Dans l'onglet "Etats et Requêtes", cochez "Visible sous Etats et Requêtes".
5. Validez et enregistrez la requête.

Création de l'exécutable et diffusion de l'application

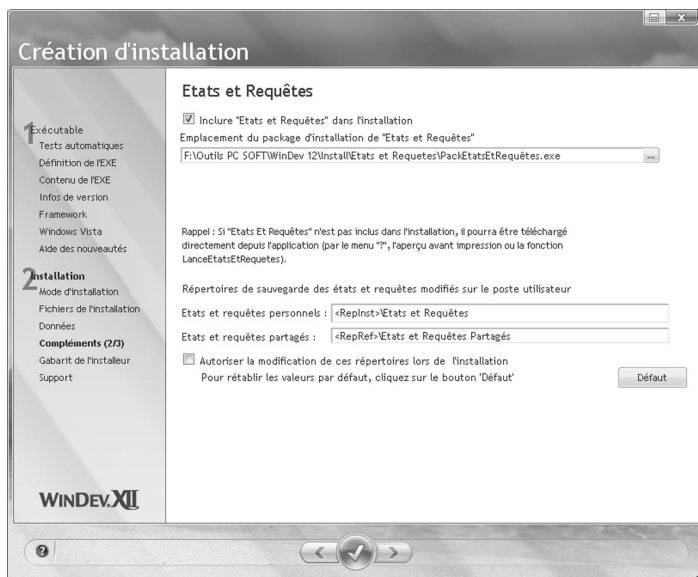
La création et la diffusion d'une application utilisant le logiciel "Etats et Requêtes" se fait de la même façon que pour une application classique. C'est lors de la création de la procédure d'installation, que vous pourrez indiquer les paramètres spécifiques au logiciel "Etats et Requêtes".

- Pour créer le programme d'installation de votre application, sélectionnez l'option "Atelier .. Créer la procédure d'installation". L'assistant de création d'installation se lance. Cliquez sur le lien "Installation".
 1. Sélectionnez "Continuer sans créer l'exécutable".
 2. Choisissez une "Installation autonome". Passez à l'écran suivant.
 3. Choisissez une "Installation standard". Passez à l'écran suivant.
 4. Dans l'écran "Fichiers installés", sélectionnez les fichiers de données (.fic, .mmo et .ndx) présent dans le répertoire de l'exécutable. Pour cela, cliquez sur "Ajouter" et sélectionnez ces fichiers. Passez à l'écran suivant.
 5. Dans l'écran "modules complémentaires", vérifiez que l'option "Intégrer l'installation de Etats et Requêtes" est cochée.
 6. Dans l'écran intitulé "Etats et Requêtes", spécifiez si vous voulez incorporer l'installation de

"Etats et Requêtes" dans votre programme d'installation.

Dans ce cas, le package d'installation du logiciel "Etats et Requêtes" est recherché. Par défaut, ce pack d'installation est présent dans le sous-répertoire "Etats et Requêtes" du répertoire d'installation de WinDev.

Attention : Incorporer le logiciel "Etats et Requêtes" augmente la taille de votre installation de façon non négligeable (environ 135 Mo).



7. Spécifiez si nécessaire les répertoires correspondant aux états et requêtes modifiés ou créés par l'utilisateur final.

Le répertoire des états et requêtes personnels correspond au répertoire des états et requêtes visibles uniquement par l'utilisateur qui les a créés.

Le répertoire des états et requêtes partagés correspond au répertoire des états et requêtes visibles par tous les utilisateurs de l'application.

Ces répertoires peuvent ou non être modifiés lors de l'installation de l'application.

8. Continuez l'assistant jusqu'à la génération du programme d'installation de votre application (dans notre exemple, l'application "Faire des éditions").

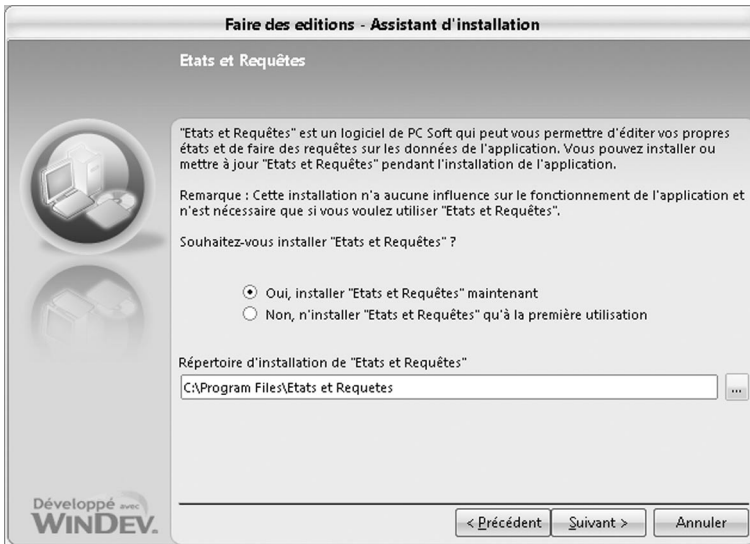
Installer et utiliser le logiciel "Etats et Requêtes"

Installation de l'application

Pour tester l'installation et l'utilisation du logiciel "Etats et Requêtes", nous allons installer l'application "Faire des éditions".

Le programme d'installation est présent dans le sous-répertoire "Autoformation\Exercices\Faire des éditions\Install"

- ▶ Lancez le programme "Install.exe". L'installation se lance.
- ▶ L'écran "Etats et Requêtes" permet d'installer le logiciel "Etats et Requêtes" (uniquement si ce programme n'est pas déjà installé sur le poste en cours).



Passez à l'écran suivant.

- ▶ L'écran suivant permet de définir les répertoires utilisés pour l'enregistrement des états.



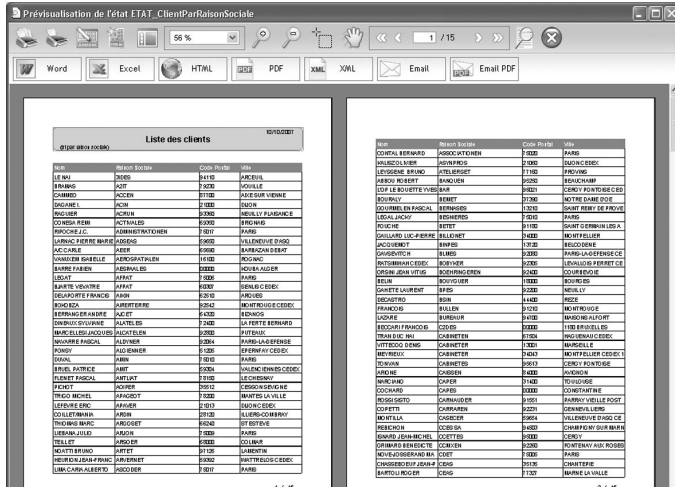
Passez à l'écran suivant et terminez l'installation de l'application.

- ▶ L'installation de l'application, puis l'installation du logiciel "Etats et Requêtes" se lance.

Test de l'application

Nous allons maintenant nous placer dans la peau de l'utilisateur final de votre application (ce n'est pas évident, mais pourquoi pas).

- ▶ Lancez l'application "Faire des éditions".
- ▶ Sélectionnez l'option "Edition .. Liste triée". L'état apparaît en mode "Aperçu avant impression".



Dans cet écran, deux nouveaux icônes font leur apparition :



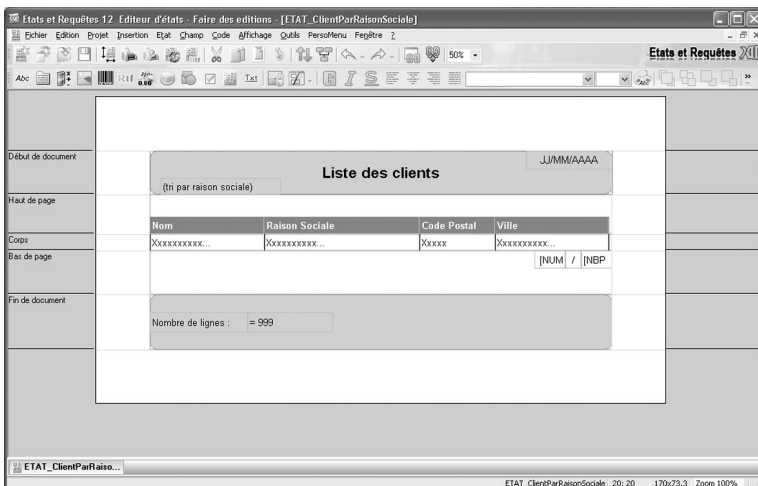
Permet de modifier l'état en cours dans l'aperçu avant impression.



Permet de créer un nouvel état.

- ▶ Nous allons modifier cet état :

1. Cliquez sur l'icône .
2. L'état apparaît en édition sous le logiciel "Etats et Requetes".




3. L'utilisateur peut alors faire les modifications qu'il souhaite :

- sur le style des informations affichées (par exemple changer la couleur d'un champ)
- sur le contenu de l'état (par exemple ajouter un champ, ...). Si l'état est lié à une requête, l'utilisateur peut modifier cette requête.

Si l'état est lié à une requête intégrée, la modification est effectuée directement dans l'état. Si l'état est lié à une requête indépendante, la requête est également modifiée. La modification sera prise en compte uniquement dans le cadre de l'exécution de l'état (si une fenêtre utilise également cette requête, la modification ne sera pas prise en compte par la fenêtre).

► Dans cet exemple, nous allons mettre la date en rouge. Pour cela :

1. Sélectionnez le champ date dans le début de document.
2. Dans la barre des champs, cliquez sur l'icone "Palette" () et sélectionnez la couleur rouge.
3. Enregistrez votre état. Par défaut, le répertoire proposé est le répertoire d'installation de votre application.
4. Fermez le logiciel "Etats et Requêtes" ainsi que l'aperçu en cours.
5. Sélectionnez l'option "Edition .. Liste triée" dans l'application. L'état apparaît en mode "Aperçu avant impression" et la modification réalisée est effective.

Dans cet exemple, l'état que vous avez modifié est disponible pour vous seul. Dans le cas d'une application réseau, l'utilisateur a la possibilité de mettre à disposition de tous les utilisateurs un état modifié.

Nous avons fini notre découverte du logiciel "Etats et Requêtes". Pour plus de détails, consultez l'aide en ligne (mot-clé : Etats et Requêtes).



DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 8

Programmation avancée

XIII
Express

LEÇON 8.1. IMPORT / EXPORT

Ce que vous allez apprendre dans cette leçon ...

- Fichier CSV et fichiers texte
- Fichier XML
- Fichier XLS



Durée estimée : 20mn

Présentation

WinDev permet de manipuler simplement les formats les plus courants de données : fichiers texte (fichiers INI, CSV, ...), fichiers XML, fichiers XLS, ... De nombreuses familles de fonctions WLangage permettent aussi bien de lire ces fichiers que de créer ces fichiers. Vous pouvez ainsi lire facilement dans vos applications WinDev des données générées par d'autres logiciels ou encore créer des fichiers nécessitant un formatage particulier.

Dans ce chapitre, nous nous consacrerons plus particulièrement à trois formats de fichiers :

- les fichiers Texte (fichiers texte, fichiers INI et fichiers CSV)
- les fichiers XML
- les fichiers XLS

Manipulation des fichiers texte, CSV, ...

Les fichiers externes sont souvent appelés fichiers texte (ou ASCII) ou encore fichiers à accès direct. Notons que ce type de fichier contient généralement du texte, mais peut également contenir des informations binaires telles que des images, des sons, ...

Dans cette leçon comme dans toute la documentation de WinDev, nous parlerons de **fichiers externes**.

WinDev permet de manipuler en programmation les fichiers externes. Les fonctions du WLangage permettent de créer, lire ... des fichiers externes. Toutes les fonctions peuvent être très facilement identifiées : elles commencent toutes par la lettre "F".

Dans un fichier externe, il n'y a pas nécessairement de notion "d'enregistrement". Pour pouvoir manipuler un fichier externe, il faut en connaître la structure, c'est-à-dire connaître l'agencement des données qui se trouvent à l'intérieur du fichier.

Les fichiers CSV sont simplement des fichiers texte utilisant une structure spécifique. Ils sont donc manipulables comme les fichiers textes.

Les fichiers .INI sont également des fichiers texte utilisant une structure spécifique. Pour les manipuler, WinDev dispose de deux fonctions spécifiques : INILit et INIEcrit.

A propos des caractères accentués

Si le fichier a été créé sous MS-DOS, les caractères accentués sont au standard OEM. Si le fichier a été créé sous Windows, les caractères accentués sont au standard ANSI.

Dans les deux standards, les caractères accentués n'ont pas le même "code" et n'ont pas la même apparence.

Ainsi sous Windows, pour les fichiers externes créés sous MS-DOS, les caractères accentués s'afficheront sous la forme de "■".

Les fonctions **OemVersAnsi** et **AnsiVersOem** permettent de convertir les caractères d'un standard vers l'autre.

Exemple

Pour simplifier la présentation des fonctions de gestion des fichiers texte, nous avons préparé un exemple contenant les principales manipulations possibles sur les fichiers texte et les répertoires

- ▶ Ouvrez le projet "WDImport_Export.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\WDImport_Export" de WinDev.
- ▶ Ouvrez la fenêtre "FEN_FichierTexte.WDW" et testez-la. Cette fenêtre regroupe de nombreuses manipulations possibles avec WinDev.

Nous vous conseillons également d'étudier l'exemple "Fichiers Texte" (exemple didactique) livré en standard avec WinDev : vous aurez ainsi un aperçu des différentes manipulations possibles sur les fichiers externes :

- créer un fichier externe
- lire (par bloc, par ligne ou en une seule fois) un fichier externe
- écrire dans un fichier externe
- obtenir des informations sur un fichier
- bloquer un fichier
- tester l'existence d'un fichier
- sélectionner un fichier
- copier un fichier
- supprimer un fichier
- lister les fichiers d'un répertoire

Cet exemple présente également les manipulations possibles sur les disques et les répertoires :

- créer un répertoire
- connaître le répertoire en cours
- tester l'existence d'un répertoire
- lister des répertoires
- copier un répertoire
- supprimer un répertoire



EXEMPLE

Exemple

L'exemple "Fichiers Texte" livré en standard avec WinDev (exemple didactique), illustre les manipulations possibles sur les fichiers, disques et répertoires avec le WLangage. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

Pour plus de détails sur les fonctions de gestion des fichiers externes, consultez l'aide en ligne (mot-clé : "Fichier externe").

Manipulation des fichiers XML

XML (Extensible Markup Language, ou Langage Extensible de Balisage) est le langage destiné à succéder à HTML sur le World Wide Web. Comme HTML (Hypertext Markup Language), le XML est un langage de balisage, c'est-à-dire un langage qui présente de l'information encadrée par des balises. Mais contrairement à HTML, qui présente un jeu limité de balises, XML est un **métalangage**, qui permet d'inventer de nouvelles balises pour isoler toutes les informations élémentaires que peut contenir une page Web.

XML permet de structurer un document contenant des données. Un fichier de données Hyper File contenant plusieurs rubriques et enregistrements peut par exemple être exporté dans un fichier XML (fonctions **XML***).



EXEMPLE

Exemple

Pour plus de détails sur l'utilisation du XML, consultez l'exemple "XML" (exemple didactique), livré en standard avec WinDev. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

WinDev gère en standard :

- les fichiers au format XML grâce à un accès natif fourni avec WinDev. Pour plus de détails, consultez l'aide en ligne.
- les exportations vers le XML (fonctions **TableVersXML**, **TexteVersXML**, **HExporteXML**)
- l'import de données XML (fonction **HImporteXML**)
- la manipulation d'un document XML. Les fonctions permettant de manipuler un document au format XML commencent toutes par **XML***. Il est possible :
 - Créer un document XML manipulable par WinDev (fonction **XMLDocument**)
 - Lire et naviguer dans un document XML (fonctions **XMLFils**, **XMLParent**, **XMLPremier**, ...)
 - Modifier un document XML (fonctions **XMLInsèreDocument**, **XMLInsèreElément**, **XMLModifie**, ...)
 - Exécuter une requête XPath dans un document XML (fonction **XMLExécuteXPath**)

Exemple de lecture d'un fichier XML

Nous allons travailler sur un fichier exemple "Exemple.XML".

- ▶ Ouvrez le projet "WDImport_Export.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\WDImport_Export" de WinDev 12.

Le fichier "Exemple.XML" contient les informations suivantes :

```
<COMMANDE>
  <LIGNECDE NUMERO="1">
    <PRODUIT> Choux </PRODUIT>
    <PRIXUNI MONNAIE="Euro"> 2.06 </PRIXUNI>
    <QUANTITE> 2 </QUANTITE>
  </LIGNECDE>
  <LIGNECDE NUMERO="2">
    <PRODUIT> Carotte </PRODUIT>
    <PRIXUNI MONNAIE="Euro"> 1.93 </PRIXUNI>
    <QUANTITE UNITE="KG"> 1 </QUANTITE>
  </LIGNECDE>
  <LIGNECDE NUMERO="3">
    <PRODUIT> Ail </PRODUIT>
    <PRIXUNI MONNAIE="Euro"> 0.65 </PRIXUNI>
    <QUANTITE UNITE="Gousse"> 1 </QUANTITE>
  </LIGNECDE>
</COMMANDE>
```

Ce document XML représente une commande ("**COMMANDE**") de trois lignes de produits :

- 2 choux à 2.06 Euros pièce (<LIGNECDE NUMERO="1">)
- 1 kilo de carottes à 1.93 Euro le kilo (<LIGNECDE NUMERO="2">)
- 1 gousse d'ail à 0.65 Euro la gousse (<LIGNECDE NUMERO="3">)

- ▶ Ouvrez la fenêtre "FEN_ExempleXML.WDW".

Traiter un document XML

Afin de pouvoir effectuer des manipulations sur le document XML, il est nécessaire de "créer" un document XML avec la fonction WLanguage **XMLDocument**. Voici la syntaxe d'utilisation de cette fonction :

```
-- Code du bouton Créer un document
SAI_OPERATIONS = ""
GR_TraiteXML..Etat = Gris 
CodeXML est une cha ne
CodeXML = fChargeTexte("Exemple.XML")
//Fermeture du document s'il s'agit du deuxi me clic
XMLTermine("Commande")
//Cr ation du document
XMLDocument("Commande", CodeXML)
```

```

SI ErreurDétectée ALORS
    Erreur("Le document n'est pas au format XML")
    RETOUR
SINON
    SAI_OPERATIONS = CodeXML
    GR_TraiteXML..Etat = Actif
    Info("Le document XML a bien été créé")
FIN

```

Ce code permet de charger le code XML (présent dans un document texte) dans un document XML nommé "Commande". Ce code est récupéré grâce à la fonction **fChargeTexte** du WLanguage. Le document XML sera ensuite manipulé par ce nom "Commande".

Parcourir un document XML

Le parcours d'un document XML ressemble beaucoup au parcours d'un fichier de données (avec les fonctions **HLit***).

Les fonctions **XMLPremier**, **XMLSuivant**, **XMLDernier**, **XMLPrécédent** et **XMLEnDehors** permettent de parcourir un document XML en se positionnant sur les balises contenues dans ce document. Les fonctions **XMLNomElément** et **XMLDonnée** permettent entre autre de récupérer le nom de la balise en cours de lecture et sa valeur.



Notes

Vous devez toujours avoir à l'esprit la notion de **niveau** dans un document XML. En effet, un document XML est un document structuré avec une racine, des noeuds "parents" et des noeuds "enfants".

Dans notre exemple, les balises "LIGNECDE" sont toutes de même niveau. "COMMANDE" est la balise de niveau supérieure par rapport aux balises "LIGNECDE". Les balises "PRODUIT", "PRIXUNI" et "QUANTITE" sont des balises de niveaux inférieurs par rapport aux balises "LIGNECDE".

La fonction **XMLEnDehors** permet de détecter la fin d'un document XML, mais aussi la fin d'un niveau.

```

SI XMLEnDehors ("Commande") ALORS
    Erreur("Il n'y a plus de balise à lire sur ce niveau")
FIN

```

Se positionner sur les balises

Pour se positionner sur la première balise du document, il faut utiliser la fonction **XMLPremier** :

```

XMLPremier ("Commande")
Info ("Première balise du parcours : " + ...
    XMLNomElement ("Commande") )

```

Pour se positionner sur la dernière balise d'un niveau, il faut procéder de la même façon avec la fonction **XMLDernier**.

Pour se positionner sur la balise suivante **de même niveau**, il faut utiliser la fonction **XMLSuivant** :

```
XMLSuivant ("Commande")

SI XMLEnDehors ("Commande") ALORS
    Info ("Il n'y a plus de balise à lire sur ce niveau")
SINON
    Info ("Balise suivante du parcours : " + ...
        XMLNomElement ("Commande"))
FIN
```

Pour se positionner sur la balise précédente, il faut procéder de la même façon avec la fonction **XMLPrécédent**.

Le message d'erreur vous indique que vous êtes arrivé en fin de parcours. Cela est tout à fait normal, puisqu'il n'y a pas d'autres balises de même niveau que "<COMMANDE>".

Pour se positionner sur la balise suivante sur un autre niveau, il est nécessaire d'utiliser les fonctions **XMLFils** (pour se positionner sur la première balise fille de la balise en cours) et **XMLParent** (pour se positionner sur la balise mère de la balise en cours).



Notes

Si une balise possède un attribut (par exemple "<LIGNECDE NUMERO=1>"), la fonction **XMLFils** se positionne sur l'attribut et le considère comme une balise.

Récupérer des valeurs

La fonction permettant de récupérer le nom de la balise courante est **XMLNomElement** :

```
Info ("La balise courante est " + XMLNomElement ("Commande"))
```

La fonction permettant de récupérer la valeur de la balise courante est **XMLDonnée** :

```
Info ("La valeur de la balise courante est " + ...
    XMLDonnée ("Commande"))
```

La fonction **XMLLit** permet aussi de récupérer directement le nom d'une balise ou la valeur d'une balise en exécutant une requête XPATH.

```
// Affiche les informations concernant la ligne de commande 2
Info ("Les informations concernant le second produit " + ...
    "de la ligne de commande sont " + RC + ...
    XMLLit ("Commande", "/COMMANDE/LIGNECDE [2] /PRODUIT"), ...
    XMLLit ("Commande", ...
        "/COMMANDE/LIGNECDE [2] /PRIXUNI/@MONNAIE"), ...
    XMLLit ("Commande", "/COMMANDE/LIGNECDE [2] /PRIXUNI"))
```

Effectuer une recherche

Vous avez la possibilité d'effectuer une recherche dans un document XML. L'élément recherché peut être une balise, un attribut ou une valeur

La fonction permettant d'effectuer des recherches dans un document XML est **XMLRecherche**.

Nous ne détaillerons pas cette fonction. Consultez l'aide en ligne (mot-clé : "XMLRecherche").

Pour en savoir plus sur les fonctionnalités XML de WinDev, consultez l'aide en ligne (mot-clé : "XML").

Manipulation des fichiers XLS

Pour manipuler les fichiers Excel, WinDev met à votre disposition les fonctions xlsXXX. Ces fonctions permettent de lire les données présentes sur les différentes feuilles d'un fichier Excel.

Un test rapide des possibilités de ces fonctions ?

- ▶ Ouvrez la fenêtre "FEN_Exemple_XLS" et testez cette fenêtre.
- ▶ Le fichier proposé contient 3 feuilles. Il est possible d'afficher le contenu des trois feuilles.
- ▶ Ouvrez le fichier (bouton "Ouvrir le fichier sous Excel") pour vérifier les informations affichées par WinDev.

Les principales fonctionnalités utilisées dans ces fenêtres sont :

- l'ouverture d'un fichier Excel (fonction **xlsOuvre**)
- la récupération du nombre de feuilles du fichier
- la récupération du nombre de lignes et de colonnes de chaque feuille
- la lecture des données de chaque feuille.

LEÇON 8.2. COMPILATION DYNAMIQUE

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Tracer une courbe en compilation dynamique



Durée estimée : 20mn

Présentation

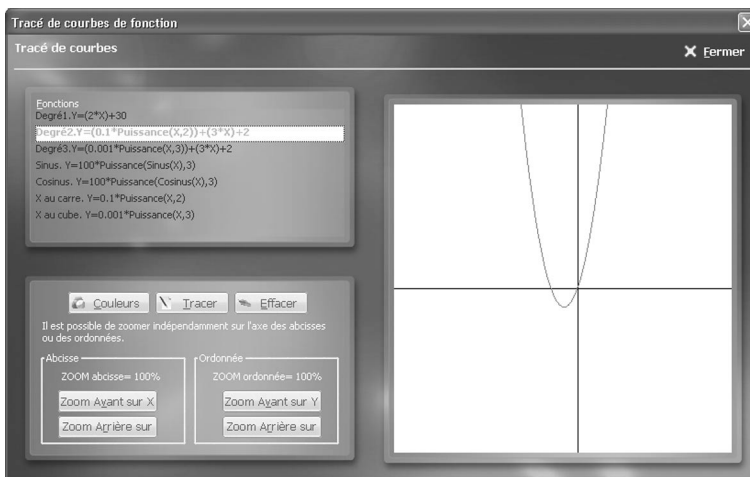
La compilation dynamique permet de compiler un code à n'importe quel moment de l'application. Un exemple d'utilisation courant ? Votre application contient une formule paramétrable. Pour changer les paramètres, il n'est pas nécessaire de re-livrer l'exécutable. Seule la modification par exemple d'un fichier texte suffit.

Exemple

Dans l'exemple Programmation, la compilation dynamique est utilisée pour réaliser un dessin de courbes mathématiques.

Nous allons tout d'abord tester cet exemple :

1. Ouvrez le projet "Programmation.WDP" présent dans le sous-répertoire "Autoformation\Corrigés\Programmation" de WinDev.
2. Ouvrez la fenêtre "FEN_Dessine" (par exemple en utilisant le raccourci CTRL + E).
3. Testez cette fenêtre :
 - Choisissez la fonction à dessiner
 - Choisissez les couleurs du dessin (bouton "Couleurs").
 - Validez grâce au bouton "Tracer"



4. Retournons sous l'éditeur de code pour étudier le code permettant de dessiner cette courbe. Fermez la fenêtre.
5. Affichez le code du bouton "Tracer". Ce code appelle la procédure AfficheCourbe.
6. Positionnez le curseur de la souris sur le nom de la procédure et utilisez la touche [F2]. Le code de la procédure apparaît automatiquement sous l'éditeur de code.

Le code de cette procédure peut être décomposé en plusieurs parties :

- initialisation des variables.
- dessin du fond du graphe et des axes.

- gestion des erreurs (nous verrons ce point plus loin dans ce chapitre).
- déclaration du code compilé dynamiquement.

```
// Code de la fonction compilée dynamiquement
CodeFonction est une chaîne = [
Fonction ValeurY(Fonct,X)
Y est un entier
%1
Renvoyer PartieEntiere(Y)
]
```

Ce code contient plusieurs points à remarquer :

- Une fonction compilée dynamiquement est contenue dans une chaîne.
 - Le code utilisé manipule une chaîne multiligne. Cette chaîne n'utilise pas de guillemets comme les chaînes classiques. Les caractères = [et] permettent de définir le début et la fin de cette chaîne. Sous l'éditeur de code, cette chaîne apparaît surlignée en rose.
 - Le code qui sera compilé est une fonction ValeurY, qui attend deux paramètres : Fonct (la fonction choisie par l'utilisateur), et X (l'abscisse du point à dessiner). La fonction renverra l'ordonnée correspondante.
- compilation de la fonction et son utilisation.

```
// On compile la fonction de calcul de la valeur de Y
ErrCompile = Compile("ValeurY", ...
ChaîneConstruit(CodeFonction, Fonct))
SI ErrCompile= "" ALORS
Xold = -201
Yold = ValeurY(Fonct, -201)
POUR X=-200 A 200
Y = ValeurY(Fonct,X)
dLigne((EchelleX*Xold)+X0,...
-(EchelleY*Yold)+Y0,(EchelleX*X)+X0,-(EchelleY*Y)+Y0)
Xold = X
Yold = Y
FIN
SINON
Erreur("La fonction n'a pu être compilée"+ErrCompile)
FIN
```

La compilation de la fonction est effectuée avec la fonction WLangage **Compile**. Dès que cette fonction est utilisée (et si aucune erreur n'est renvoyée), la fonction compilée dynamiquement devient utilisable.

LEÇON 8.3. TIMER

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Timer automatique
- Timer programmé



Durée estimée : 20mn

Présentation

Un timer permet d'exécuter en parallèle une tâche qui doit être lancée périodiquement (acquisition de données en tâches de fond, rafraîchissement automatique de données, déclencher un programme à une heure donnée ...). Seules les procédures (globales ou locales à une fenêtre) et les méthodes globales de classe peuvent être appelées par un timer.

Avec WinDev, les timers peuvent être gérés :

- soit directement depuis l'interface de l'éditeur de code : ce sont les timers automatiques.
- soit avec les fonctions WLangage : ce sont les timers programmés.

Dans cette leçon, nous verrons en détail uniquement les timers automatiques.

Créer un timer automatique

Pour simplifier la manipulation des timers, nous vous avons préparé une fenêtre utilisant un timer automatique pour afficher la date et l'heure dans une fenêtre.


- ▶ Ouvrez si nécessaire le projet "Programmation.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\Programmation" de WinDev.
- ▶ Ouvrez la fenêtre "FEN_Timer.WDW". Cette fenêtre gère un timer pour afficher l'heure dans une fenêtre. Testez cette fenêtre.



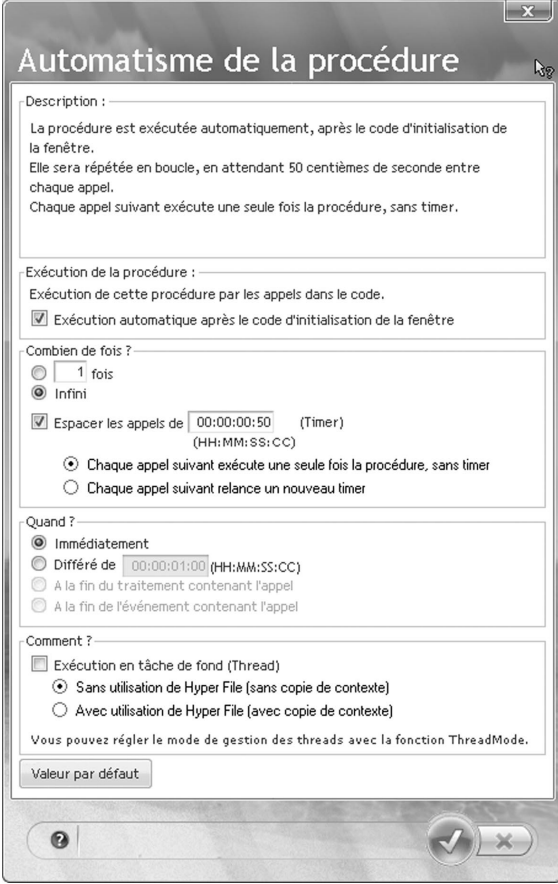
- ▶ Revenez sous l'éditeur.
- ▶ Affichez la procédure locale "MiseAHeure" (double-cliquez sur le nom de la procédure dans le volet "Code"). Cette procédure est une procédure locale à la fenêtre "FEN_Timer". Cette procédure met à jour le champ "SAI_Heure" avec la nouvelle heure. A chaque changement d'heure, des bips vont être émis et l'image de l'horloge va être affichée :

```

FEN_Timer - Procédure locale MiseAHeure
-----
Procédure locale MiseAHeure
PROCEDURE MiseAHeure()
// Mise à jour du champ HEURE
SAI_Heure=HeureSys()
// Emettre des bips chaque changement d'heure
SI SAI_Heure[[3 A 4]]="00" ALORS
  nCompteur est un entier
  POUR nCompteur = 1 _A_ Val{SAI_Heure[[1 A 2]]}
    Bip()
  FIN
  IMG_Timer.Visible=Vrai
SINON
  IMG_Timer.Visible=Faux
FIN
  
```

Le chronomètre vert situé dans l'en-tête de la procédure signale que cette procédure est une procédure automatique. Pour définir les propriétés d'une procédure, il suffit de cliquer sur l'icone  en haut à droite.

► Cliquez sur l'horloge verte pour visualiser les propriétés de la procédure :



Automatisme de la procédure

Description :

La procédure est exécutée automatiquement, après le code d'initialisation de la fenêtre.
Elle sera répétée en boucle, en attendant 50 centièmes de seconde entre chaque appel.
Chaque appel suivant exécute une seule fois la procédure, sans timer.

Exécution de la procédure :

Exécution de cette procédure par les appels dans le code.

Exécution automatique après le code d'initialisation de la fenêtre

Combien de fois ?

1 fois
 Infini

Espacer les appels de 00:00:00:50 (Timer)
(HH:MM:SS:CC)

Chaque appel suivant exécute une seule fois la procédure, sans timer
 Chaque appel suivant relance un nouveau timer

Quand ?

Immédiatement
 Différé de 00:00:01:00 (HH:MM:SS:CC)
 A la fin du traitement contenant l'appel
 A la fin de l'événement contenant l'appel

Comment ?

Exécution en tâche de fond (Thread)
 Sans utilisation de Hyper File (sans copie de contexte)
 Avec utilisation de Hyper File (avec copie de contexte)

Vous pouvez régler le mode de gestion des threads avec la fonction ThreadMode.

Valeur par défaut

Cette fenêtre est impressionnante ? Ne vous affolez pas, elle est très simple à comprendre.

La procédure MiseAHeure est exécutée après le code d'initialisation de la fenêtre (1), à l'infini (2) en espaçant les appels de 50 centièmes de seconde (3). Chaque appel de la fonction MiseAHeure exécute la fonction MiseAHeure.(4)



Notes

Arrêter une procédure automatique

Pour arrêter une procédure automatique, utilisez la fonction *FinAutomatismeProcédure*.

Créer un timer programmé

Remarque : pour compatibilité avec les versions précédentes, il est également possible de gérer les timers avec les fonctions WLangage.

La fonction **TimerSys** permet d'utiliser un timer géré par Windows. Ce timer pourra être arrêté grâce à la fonction **FinTimerSys**.

LEÇON 8.4. ÉVÉNEMENT WINDOWS

Ce que vous allez apprendre dans cette leçon ...

- Programmation des événements Windows.



Durée estimée : 10mn

Introduction

Toute action effectuée par Windows correspond à un événement Windows. Différents types d'événements peuvent être rencontrés, par exemple :

- Le survol d'une fenêtre par la souris
- L'arrêt du système
- L'affichage d'une boîte de dialogue
- Une erreur logicielle
- Etc ...

Lorsque ces événements se produisent, il est possible de les intercepter afin de préparer ou d'exécuter un traitement spécifique.

WinDev propose par défaut la gestion automatique des événements les plus utilisés. Par exemple, les événements suivants sont proposés pour un champ de saisie :

- Initialisation du champ
- Entrée dans le champ
- Modification du champ
- Sortie du champ

De nombreux autres événements peuvent être gérés facilement. Une barre d'outils est disponible dans l'éditeur de code :



Il suffit d'ajouter l'événement souhaité. Exemple : "Touche Enfoncée" .

Pour la gestion d'événements plus "spécifiques", il existe la fonction WLangage **Evénement**. La fonction **Evénement** permet de déclencher une procédure en réaction à un événement Windows. Vous pouvez ainsi associer à tout type d'événement Windows une procédure en WLangage.



Notes

Pour utiliser la fonction **Événement**, il faut maîtriser la programmation Windows et notamment les événements Windows.

Pour connaître la liste (non exhaustive) des événements Windows, consultez l'aide en ligne de WinDev (mot-clé : "Valeur des constantes de l'API Windows").

Exemple : Détecter le clic sur une liste

- ▶ Ouvrez si nécessaire le projet "Programmation.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\Programmation" de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_Evenement" et testez-la. Cette fenêtre détecte si la liste est manipulée avec la souris ou avec le clavier.



- ▶ Regardons le code utilisé, principalement dans le code de déclaration des globales de la fenêtre FEN_Evenement.

```
EXTERN "WINCONST.WL"
```

Tout d'abord, le fichier WINCONST.WL est intégré dans le code de l'application grâce au mot-clé EXTERNE. Ce fichier contient la déclaration et les valeurs des constantes Windows. A la compilation, tout le code contenu dans le fichier Winconst.WL sera automatiquement intégré au code de l'application.

Ensuite, les différents événements gérés sont déclarés :

```
// Événements sur le champ LISTE_Jour
// Pression d'une touche du clavier
Événement ("SourisOuClavier",LISTE_Jour..NomComplet,WM_KEYDOWN)
// Clic gauche de souris
Événement ("SourisOuClavier",LISTE_Jour..NomComplet,WM_LBUTTONDOWN)
```

La procédure `SourisOuClavier` est appelée dès qu'une utilisation du clavier sur la liste est effectuée (événement Windows correspondant : `WM_KEYDOWN`) ou dès que le clic gauche de la souris est utilisé (événement Windows correspondant : `WM_LBUTTONDOWN`).

- ▶ Le code de la procédure est très simple : la variable globale `bClavier` est mise à Vrai ou Faux selon l'outil utilisé (clavier ou souris).

```
PROCEDURE SOURISOUCLAVIER ()
//déterminer si la validation de l'élément a été effectuée
//avec le clavier ou avec la souris
SELON _EVE.wMessage
    CAS 256          // C'est le clavier
        bClavier=Vrai
    CAS 513         // C'est la souris (click down)
        bClavier=Faux
FIN
```

- ▶ Le message est ensuite affiché grâce au traitement associé au champ `LISTE_Jour`.

LEÇON 8.5. GESTION AUTOMATIQUE DES ERREURS

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce que la gestion automatique des erreurs ?
- Utilisation de la gestion automatique des erreurs



Durée estimée : 10mn

Présentation

WinDev peut gérer automatiquement les erreurs. Cette fonctionnalité permet à la fois de réduire sensiblement le code d'un projet tout en uniformisant et en centralisant la gestion des erreurs. L'utilisation de cette fonctionnalité rend également le code plus visible.

Fonctionnement

Lorsqu'une fonction du WLangage rencontre une erreur, deux opérations sont réalisées :

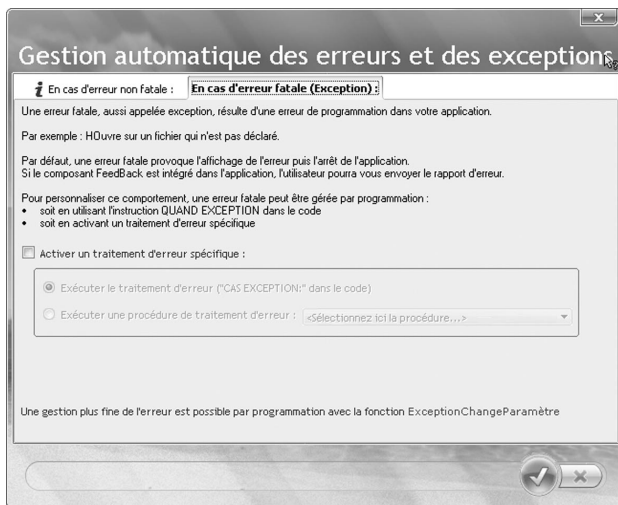
- une valeur de retour d'erreur est renvoyée par la fonction (par exemple, la fonction **fOuvre** renvoie la valeur "-1" si le fichier spécifié n'a pu être ouvert).
- l'erreur est détectée par le WLangage (la variable **ErreurDétectée** correspond à Vrai) et il est possible de récupérer des informations sur cette erreur grâce à la fonction **ErreurInfo**.

C'est cette seconde opération qui peut être gérée en automatique grâce à la gestion des erreurs de WinDev.

Mise en place

Le paramétrage de la gestion automatique des erreurs peut s'effectuer :

- soit directement dans l'éditeur de code :



- soit par programmation, avec la fonction **ErreurChangeParamètre**.

Types d'erreurs concernées


Le WLangage peut rencontrer deux types d'erreurs :

- les erreurs "non fatales" (appelées également erreurs d'exécution) : ces erreurs sont généralement gérées dans le code et ne provoquent pas l'arrêt de l'application. Par exemple, ouverture d'un fichier inaccessible ou d'une archive inexistante.
- les erreurs "fatales" (appelées également erreurs de programmation) : ces erreurs sont généralement liées à un problème matériel (plus d'espace disque, perte de connexion réseau...)

ou à des problèmes graves de conception (accès à un fichier non déclaré, utilisation de champs inexistantes...). Une erreur "fatale" peut également survenir suite à une erreur "non fatale" mal gérée. Dans ce cas, l'application va s'arrêter.

Le mécanisme de gestion des erreurs permet de gérer de façon bien distincte ces deux types d'erreurs afin de pouvoir spécifier des comportements adaptés aux erreurs rencontrées.

Utilisation de la gestion automatique des erreurs

- ▶ Ouvrez le projet "Découverte Gestion Erreurs.WDP" (option "Fichier .. Ouvrir un projet"). Ce projet se trouve dans le sous-répertoire "\Autoformation\Corrigés\Découverte Gestion Erreurs" du répertoire d'installation de WinDev.
- ▶ Ce projet présente la gestion :
 - d'**une erreur non fatale** (ouverture d'une archive inexistante).
Lors de l'exécution de la ligne de code générant l'erreur, un message d'erreur s'affiche et propose à l'utilisateur de re-essayer l'opération, d'annuler l'opération ou d'arrêter l'application.
 - d'**une erreur fatale** (division d'un entier par 0).
Lors de l'exécution de la ligne de code générant l'erreur, une procédure est automatiquement appelée. Cette procédure permet d'afficher le message d'erreur et d'arrêter le traitement en cours.
 - d'**une erreur sur plusieurs niveaux** (traitement appelant une procédure qui ouvre une archive inexistante).
Lors de l'exécution de la ligne de code générant l'erreur :
 - la procédure retourne "Faux" au traitement appelant.
 - le traitement appelant affiche un message d'erreur et arrête le traitement.
- ▶ Testez le projet en cliquant sur l'icone  .



EXEMPLE

Exemple

Pour plus de détails sur la gestion automatique des erreurs, consultez l'exemple "Detection erreurs", livré en standard avec WinDev. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

LEÇON 8.6. INDIRECTION

Ce que vous allez apprendre dans cette leçon ...

- Définition.
- Exemples d'utilisation.



Durée estimée : 20mn

Présentation

Le mécanisme "d'indirection" permet de construire le nom d'un champ, d'une rubrique de fichier ou le nom d'une variable à partir d'une expression de type chaîne.

Ceci permet par exemple de créer des traitements génériques indépendants des noms des champs, des variables, des rubriques de fichier ...

L'indirection est réalisée à l'aide des opérateurs { }.

Pour optimiser la vitesse de vos applications, il est conseillé de préciser lors de l'utilisation de la syntaxe d'indirection le type de l'élément manipulé.

Ainsi, si vous manipulez un champ, le type correspondant sera IndChamp (voir exemple ci-dessous).

Exemple

Exemple d'indirections simples :

```
{ "NOM", indChamp } = NomCli
//est équivalent à NOM=NomCli
{ "NOM", indChamp } = { "CL.NOMCLI" }
//est équivalent à NOM=CL.NOMCLI
{ "FENCLI.NOM", indChamp } = NomCli
//est équivalent à FENCLI.NOM=NomCli
{ "FENCLI"+" .NOM", indChamp } = NomCli
//est équivalent à FENCLI.NOM=NomCli
```

Exemple d'indirection avec une procédure :

```
NomChamp est une chaîne
NomChamp = "SAISIE1" //SAISIE1 est le nom du champ
//appel d'une procédure rendant un champ invisible
INVISIBLE(NomChamp)

PROCEDURE INVISIBLE(NChamp)
{NChamp, indChamp}..Visible = Faux
```



EXEMPLE

Exemple

Pour plus de détails sur l'utilisation des indirections, consultez l'exemple "Manipulation des indirections" (exemple unitaire), livré en standard avec WinDev. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants".

LEÇON 8.7. LES THREADS

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce que les threads ?



Durée estimée : 10mn

Définition

Les threads permettent d'exécuter du code (ou des traitements) en parallèle de l'application principale. Il est ainsi possible d'exécuter plusieurs traitements différents qui peuvent être très longs en tâche de fond, sans bloquer l'application principale.

Les threads remplacent certains types de timer.

Un thread efficace est un thread qui attend un événement, par exemple, une action de l'utilisateur, la réception d'un email, la réception d'un appel téléphonique, ...

Quelques exemples d'utilisation des threads :

- Récupération d'emails en tâche de fond, pendant la saisie d'un nouvel email par exemple.
- Application de communication : gestion des appels téléphoniques, communication par socket, ...

Avec WinDev, vous avez la possibilité de :

- gérer simplement des threads (fonctions **Thread***, **ThreadExécute**, **ThreadArrête ...**)
- utiliser les threads avec des "sémaphores", afin de limiter leur action en simultané à un instant donné (fonctions **Sémaphore***)
- gérer des "signaux" afin de synchroniser plusieurs threads (fonctions **Signal***, **SignalCrée**, **SignalDétruit ...**)

Exemple



EXEMPLE

Exemple

WinDev est livré en standard avec un exemple permettant de comprendre l'intérêt et l'utilisation des threads, "Pool de Threads" (exemple didactique, accessible depuis le volet "Assistants, Exemples et Composants" de WinDev).

Pour plus de détails sur la gestion du multithread avec WinDev, consultez l'aide en ligne (mot-clé : "Thread").

LEÇON 8.8. LES SOCKETS

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Application Serveur
- Application Cliente



Durée estimée : 20mn

Présentation

WinDev met à votre disposition des fonctions qui permettent de réaliser une gestion avancée des sockets.

Une socket est une ressource de communication utilisée par les applications pour communiquer d'une machine à une autre sans se soucier du type de réseau.

Ce mode de communication peut par exemple être utilisé pour établir une communication entre des postes reliés par Internet.

Une application WinDev peut gérer les sockets selon différents modes :

- Application WinDev/WebDev Cliente : l'application se connecte à un serveur quelconque et échange des données par une socket.
- Application WinDev "Serveur simplifié" : l'application WinDev est un serveur, échangeant des informations par une socket avec un seul poste client (socket WinDev conseillée sur le poste client mais non obligatoire)
- Application WinDev "Serveur standard" : l'application WinDev est un serveur, échangeant des informations par sockets avec plusieurs postes client quelconques.



EXEMPLE

Exemple

WinDev est livré en standard avec un exemple permettant de comprendre l'utilisation des sockets : "WD Utilisation des sockets" (exemple didactique, accessible depuis le volet "Assistants, Exemples et Composants" de WinDev).

Application Serveur (simplifié)

WinDev offre la possibilité de créer un serveur de socket simplifié. **Ce serveur permet de communiquer avec un seul poste client simultanément.** Ce type d'application est très utile lorsque deux applications distantes ont besoin de communiquer.

Les étapes pour créer un serveur simplifié sont les suivantes :

1. Création de la socket
2. Echange de données
3. Fermeture de la socket

Création de la socket

Pour créer la socket, le poste serveur utilise la fonction **SocketCrée**. Une socket est associée à un port spécifique. Pour simplifier la manipulation de la socket par programmation sur le poste serveur, précisez le nom de la socket.

Le poste client va venir se connecter sur cette socket pour échanger des données. La connexion entre les deux postes sera effectivement réalisée au premier échange de données entre les deux postes (c'est-à-dire à la première lecture d'informations par le poste serveur).

La connexion est établie au premier succès de la fonction **SocketLit** sur le serveur.

Echange de données

Lorsque deux postes utilisent la même socket, un flux de communication s'établit entre ces deux postes. Il est alors possible pour ces deux postes de lire et d'écrire des chaînes de caractères sur la socket.

Remarque : Pour ne pas bloquer les applications, la gestion de la réception des messages est souvent gérée par un thread spécifique.

Pour lire et écrire sur la socket, l'application serveur WinDev doit utiliser les fonctions **SocketLit** et **SocketEcrit**.

Attention : Pour réaliser une lecture, une écriture doit avoir été réalisée. Par exemple :

1. Le poste client écrit sur la socket : il envoie une requête au poste serveur.
2. Le poste serveur effectue une lecture sur la socket dans un thread. Dès qu'un message est reçu, le message est traité par le poste serveur.
3. Si une réponse au message est nécessaire, le poste serveur identifie le poste client (fonction **SocketClientInfo**), et lui envoie une réponse.

Fermeture de la socket

Pour terminer la communication, le serveur peut fermer la socket avec la fonction **SocketFerme**.

Remarque : la socket peut aussi être fermée par le poste client.

Application Cliente

Une application cliente d'un serveur de sockets se connecte à un serveur quelconque pour échanger des informations par l'intermédiaire d'une socket.

Exemple : Une application WinDev cliente peut se connecter à un serveur de news standard sur Internet.

Les étapes pour créer une application cliente sont les suivantes :

1. Connexion au serveur
2. Echange de données
3. Fin de communication

Connexion au serveur

Pour se connecter à une socket du serveur, il suffit d'utiliser la fonction **SocketConnecte**. Cette fonction permet d'effectuer une demande de connexion au serveur.

La socket est identifiée par son port et par son adresse.

Echange de données

Lorsque deux postes utilisent la même socket, un flux de communication s'établit entre ces deux postes. Il est alors possible pour ces deux postes de lire et d'écrire des chaînes de caractères sur la socket.

Remarque : Pour ne pas bloquer les applications, la gestion de la réception des messages est souvent gérée par un thread spécifique.

Pour lire et écrire sur la socket, l'application cliente WinDev doit utiliser les fonctions **SocketLit** et

SocketEcrit.

Fin de communication

Pour terminer la communication, il suffit de fermer la socket depuis le poste client avec la fonction **SocketFerme**.

Remarque : il est également possible de terminer la communication depuis le poste serveur.

LEÇON 8.9. LE FTP

Ce que vous allez apprendre dans cette leçon ...

- Prise de connaissance avec les fonctions FTP de WinDev.



Durée estimée : 20mn

Présentation

Le FTP (File Transfert Protocol) est un protocole standard permettant d'effectuer le transfert de fichiers d'une machine vers une autre machine. Une des deux machines doit être un serveur FTP.

Le WLangage possède les commandes de programmation nécessaires pour effectuer des transferts de fichiers en utilisant ce protocole avec un serveur. Ces fonctions de programmation commencent toutes par "FTP".

WinDev ne propose que des fonctions "clientes" pour le FTP. Il est nécessaire d'utiliser un serveur FTP standard.

Connexion à un serveur FTP

La fonction WLangage **FTPConnecte** permet de se connecter à un serveur FTP. Vous devez au préalable posséder un compte FTP (nom d'utilisateur et mot de passe) pour accéder à un serveur FTP.

Exemple de code :

```
NomUser est une chaîne = "Utilisateur"
Password est une chaîne = "MotDePasse"
AdresseFTP est une chaîne = "ftp.monsite.com"
Res est un entier
Res = FTPConnecte(AdresseFTP, NomUser, Password)
SI Res = -1 ALORS
    Erreur("Impossible de se connecter." + RC + ...
           ErreurInfo(errMessage))
SINON
    Info("Connexion réussie.")
FIN
```

Il est également possible de préciser le numéro de port de connexion du serveur FTP (par défaut "21") ainsi que le mode de connexion ("Vrai" pour une connexion dite "passive", "Faux" pour une connexion dite "active"). Pour plus de détails, consultez l'aide en ligne (mot-clé : "FTP, Fonctions").

Envoyer un fichier

Pour envoyer un fichier vers un serveur FTP, il suffit tout simplement d'utiliser la fonction **FTPEnvoie** :

```
// Lors de la connexion au serveur avec FTPConnecte, nous avons
// récupéré le numéro de connexion dans la variable Res
// Transfert du fichier "C:\MesDocuments\Fichier.DOC" vers
// le répertoire "Temp" qui se trouve sur le serveur.
Result est un booléen = FTPEnvoie(Res, ...
                                   "C:\MesDocuments\Fichier.DOC", "/Temp")
```




Attention !

Faites très attention à la casse (minuscules, majuscules) du nom des répertoires sur le serveur. En effet, certains serveurs FTP sont des machines fonctionnant sous UNIX et sont "case-sensitive", c'est-à-dire qu'ils sont sensibles à la casse du nom des fichiers et des répertoires.

Par exemple, si sur le serveur FTP se trouve un répertoire nommé "MonRepertoire" et que vous tentez d'accéder au répertoire "monrepertoire", le serveur FTP pourra vous renvoyer une erreur du type "Chemin inexistant".

Récupérer un fichier

Pour récupérer un fichier depuis un serveur FTP sur votre machine, il suffit tout simplement d'utiliser la fonction **FTPRecupère** :

```
// lors de la connexion au serveur avec FTPConnecte, nous avons
// récupéré le numéro de connexion dans la variable Res
// Récupération du fichier "/Document/Fichier.DOC" présent
// sur le serveur FTP vers le répertoire "D:\Temp" sur
// le poste en cours
Res est un booléen = FTPRecupère(Res, ...
                                "/Document/Fichier.DOC", "D:\Temp")
```

Déconnexion

Comme pour la fermeture d'une session de messagerie, il existe en WLangage une fonction pour se déconnecter d'un serveur FTP. Cette fonction se nomme **FTPDeconnecte** :

```
// lors de la connexion au serveur avec FTPConnecte, nous avons
// récupéré le numéro de connexion dans la variable Res
// Déconnexion
FTPDeconnecte(Res)
```

Il existe également d'autres fonctions FTP que nous ne détaillerons pas dans ce cours. Elles permettent notamment de :

- créer, supprimer, modifier des répertoires sur le serveur FTP,
- créer, supprimer, modifier des fichiers sur le serveur FTP,
- récupérer les informations d'un répertoire et/ou d'un fichier,
- ...

Pour plus de détails, consultez l'aide en ligne (mot-clé : "FTP, Fonctions").

LEÇON 8.10. LES EMAILS

Ce que vous allez apprendre dans cette leçon ...

- Prise de connaissance avec les fonctions Email de WinDev.



Durée estimée : 20mn

Présentation

Le WLangage possède toutes les fonctions de programmation nécessaires pour gérer la réception ou l'envoi de messages Internet (appelés "emails"). De plus, grâce à WinDev, vous pourrez connaître toutes les caractéristiques d'un email :

- expéditeur, destinataires
- date d'envoi, sujet, message
- pièces jointes ...

WinDev permet de gérer les emails grâce à deux méthodes :

1. Les protocoles **POP3 / SMTP** (méthode la plus courante). Ces protocoles sont des protocoles de gestion des emails reconnus par tous les fournisseurs de service. Ces protocoles vous permettent de dialoguer directement avec le serveur, disponible chez votre fournisseur d'accès.
2. L'API "**Simple Mail API**" (appelé aussi **SMAPI** ou **Simple MAPI**) : ce mode de gestion des emails est utilisé par la plupart des applications Microsoft et principalement Microsoft Exchange.

Nous évoquerons dans ce cours uniquement la méthode de gestion par les protocoles POP3 / SMTP. En effet, c'est le mode qui est actuellement le plus utilisé dans le monde.

Pour plus de détails sur Simple MAPI, consultez l'aide en ligne (mot-clé : "Simple MAPI").

Les fonctions de gestion des emails commencent toutes par "Email".

Les protocoles POP3 / SMTP

Le protocole POP3 permet de recevoir des emails.

Le protocole SMTP permet d'envoyer des emails

Nous ne détaillerons pas ici le principe de fonctionnement de ces protocoles.

Pour envoyer ou lire des messages grâce aux protocoles POP3/SMTP, il faut :

1. Se connecter au fournisseur d'accès (si nécessaire).
2. Ouvrir une session de messagerie avec la fonction **EmailOuvreSession**.
3. Envoyer et lire les messages.
4. Fermer la session de messagerie avec la fonction **EmailFermeSession**.
5. Se déconnecter (si nécessaire).



EXEMPLE

Exemple

Pour plus de détails sur le protocole POP3, consultez l'exemple "POP3" (exemple complet), livré en standard avec WinDev. Cet exemple est accessible depuis le volet "Assistants, Exemples et Composants" de WinDev.

Ouvrir une session de messagerie

- Pour ouvrir une session de messagerie, il faut utiliser la fonction **EmailOuvreSession** :

```

USER est une chaîne = "Utilisateur"
Password est une chaîne = "MotDePasse"
ServeurPOP est une chaîne = "pop.wanadoo.fr"
ServeurSMTP est une chaîne = "smtp.wanadoo.fr"
NomUser est une chaîne
SI EmailOuvreSession (USER, Password, ServeurPOP, ServeurSMTP) ALORS
    NomUser = USER
SINON
    NomUser = ""
    Erreur("Impossible d'établir la connexion")
FIN

```

Avec la fonction **EmailOuvreSession**, vous pourrez à la fois lire et envoyer des emails. Pour plus de détails, consultez l'aide en ligne (mot-clé : "Email, Fonctions Emails").

Envoyer un email

Pour envoyer un message avec WinDev, il suffit de renseigner les principales informations d'un email qui seront contenues dans la structure **Email**. Nous listons ici les éléments indispensables à l'envoi d'un email. Bien entendu, il existe d'autres informations que vous pouvez renseigner (fichiers attachés, destinataires en copie cachée, etc ...); consultez l'aide en ligne pour plus de détails (mot-clé : "Email, Structure email").

Email.AdresseExpéditeur	Chaîne de caractères obligatoire contenant l'adresse de l'expéditeur.
Email.Expéditeur	Chaîne de caractères contenant l'adresse à laquelle la réponse au message doit être envoyée.
Email.NbDestinataire	Entier contenant le nombre de destinataires du message. Il est conseillé de limiter ce nombre à 20, en effet certains serveurs n'acceptent pas plus de 20 destinataires par message.
Email.Destinataire	Tableau de chaînes de caractères contenant les adresses des destinataires principaux.
Email.Sujet	Chaîne de caractères contenant le sujet du message
Email.HTML	Chaîne de caractères contenant le message avec une mise en forme HTML. Limité à 2 Go
Email.Message	Chaîne de caractères contenant le texte du message (avec une mise en forme texte simple). Limité à 2 Go



Astuce

Dans le cas d'un email avec un contenu de type "texte brut", utilisez la variable **Email.Message** plutôt que la variable **Email.HTML**.

Le code suivant illustre le remplissage des principales variables à renseigner :

```
// Connexion au serveur effectuée avec EmailOuvreSession
// Remplissage des variables Email
Email.AdresseExpediteur = "pcsoft@pcsoft.fr"
Email.Expediteur = "pcsoft@pcsoft.fr"
Email.NbDestinataire = 1
Email.Destinataire[1] = "utilisateur@mail.com"
Email.Sujet = "Envoyer des emails avec WinDev !"
Email.Message = "Bonjour, ceci est un test"
```

Pour mettre en forme le message, on aurait pu écrire à la place de **Email.Message** :

```
Email.HTML = "<HTML><BODY><P ALIGN='CENTER'>Bonjour, ceci" ...
+ " est un test</P></BODY></HTML>"
```

Il est possible d'utiliser les deux formes ensemble (texte et HTML). Ainsi les destinataires lisant leur email avec un outil incapable d'afficher du HTML verront le texte de **Email.Message**.

- Pour envoyer le message que nous venons de décrire, il suffit tout simplement d'utiliser la fonction **EmailEnvoieMessage** :

```
// Lors de la connexion nous avons récupéré le nom
// de l'utilisateur dans la variable NomUser
// Envoie le mail
EmailEnvoieMessage(NomUser)
SI ErreurDétectée ALORS
    Erreur("EmailEnvoieMessage a échoué '" + ...
        ErreurInfo(errMessage) + "'")
SINON
    Info("EmailEnvoieMessage a réussi")
FIN
```

- Pour ré-initialiser toutes les variables Emails (par exemple lors de l'envoi de plusieurs emails les uns à la suite des autres), il suffit d'utiliser la fonction **EmailRAZ**.

Lire un email

Maintenant que nous sommes connectés à notre serveur et que nous avons envoyé un message, nous allons voir comment lire un email.

Sur le même principe qu'un parcours de fichier avec les fonctions **HLitPremier**, **HLitSuivant** ... il est possible de lire les emails reçus avec les fonctions **EmailLitPremier**, **EmailLitSuivant** ... Pour plus de détails sur ces fonctions, consultez l'aide en ligne (mot-clé : "Email, Lire un email").

Pour chaque email lu, la structure **Email** est initialisée. Les variables de cette structure correspondent à l'ensemble des caractéristiques de l'email en cours.

Exemple de code :

```
// Connexion au serveur effectuée avec EmailOuvreSession
// Lors de la connexion nous avons récupéré le nom
// de l'utilisateur dans la variable NomUser
// Lecture du premier email
sMessage est une chaîne
EmailLitPremier(NomUser)
SI PAS Email.EnDehors ALORS
  SI Email.HTML = "" ALORS
    sMessage = Email.Message
  SINON
    sMessage = "Email au format HTML:" + RC + Email.HTML
  FIN
Info("Expéditeur: " + Email.Expediteur + RC + ...
     "Date d'envoi: " + Email.DateReception + RC + ...
     "Sujet: " + Email.Sujet + RC + "Message: " + sMessage)
FIN
```

Comment récupérer un document attaché ?

Utilisez la variable **Email.Attache** et la fonction **EmailSauveFichierAttaché**. Exemple de code :

```
Repert_Temp est une chaîne = "C:\temp\"
CID est une chaîne
I est un entier
// Pour chaque fichier attaché
POUR I = 1 à Email.NbAttache
  // Copier le fichier dans un répertoire temporaire
  EmailSauveFichierAttaché(Email.Attache[I], Repert_Temp ...
    + Email.Attache[I])
  // Récupérer l'identifiant du fichier attaché dans le mail
  CID = "cid:" + Email.IdentifiantAttache[I]
  // Remplacer les références au fichier attaché par
  // le nom réel du fichier
  Email.HTML = Remplace(Email.HTML, CID, "file:" + ...
    Repert_Temp + Email.Attache[I])
FIN
```

Déconnexion

Pour vous déconnecter, il suffit tout simplement d'utiliser la fonction **EmailFermeSession** :

```
SI NomUser <> "" ALORS
    EmailFermeSession(NomUser)
    NomUser = ""
FIN
```

Autres possibilités

Il est également possible de :

- envoyer des fichiers attachés dans vos emails. Pour plus de détails, consultez l'exemple "POP3" (exemple complet) fourni avec WinDev.
- effectuer des mailings par emails. Pour plus de détails, consultez l'exemple "Mailing par email" (exemple didactique).
- manipuler les données Outlook (grâce aux fonctions email du WLangage).
- manipuler les données Lotus Notes (grâce aux fonctions Notesxxx du WLangage).

Pour plus de détails sur la gestion des emails avec WinDev, consultez l'aide en ligne (mot-clé : "Email, Gestion des emails").

LEÇON 8.11. LE GRAPHEUR

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Le champ graphe
- Les fonctions graphe



Durée estimée : 20mn

Présentation

Les graphes sont un outil de présentation de statistiques bien souvent plus parlant qu'un simple tableau. Dans toute application de gestion vient un moment où l'utilisation de graphes devient vite nécessaire.

WinDev met à votre disposition plusieurs outils pour intégrer des graphes dans vos applications :

- le champ Graphe intégré dans WinDev (dans l'éditeur de fenêtres comme dans l'éditeur d'états
- les fonctions WLangage de gestion des graphes

WinDev propose également de créer un graphe automatiquement à partir du menu contextuel des tables : il suffit que la table contienne au moins une colonne de type numérique pour que l'option "Graphique" soit disponible.

Dans cette leçon, nous allons voir les possibilités du champ Graphe et des fonctions WLangage.

Le champ Graphe

Pour découvrir les possibilités du champ Graphe, nous avons préparé une fenêtre regroupant les différents modes de remplissage d'un champ Graphe.

- ▶ Ouvrez le projet "Grapheur". Ce projet est disponible dans le sous-répertoire "Autoformation\Corrigés\Grapheur" de WinDev.
- ▶ Ouvrez la fenêtre "Champ_Graphe" et testez-la. Cette fenêtre présente différents modes de remplissage d'un champ graphe :
 - Table (fichier ou mémoire)
 - Données fournies par programmation
 - Liste de valeurs

Dans chaque cas, il est possible de modifier directement les informations affichées dans le graphe. Ces modifications sont immédiatement prises en compte.



Astuce

Le menu contextuel du champ graphe permet de modifier de nombreuses options d'affichage du graphe.

- ▶ Revenez sous l'éditeur de WinDev. Nous allons créer une fenêtre contenant un champ graphe basé sur une requête. Cette requête (REQ_CA_par_année) permet d'obtenir le Chiffre d'affaires par année.
 1. Ouvrez la requête REQ_CA_par_année sous l'éditeur.

2. Testez la requête.

Aperçu de REQ_CA_par_annee

6 enregistrements correspondent à votre requête.

Année	la_somme_TotalVente
2001	55 325,00
2002	59 524,00
2003	33 046,00
2004	23 248,00
2005	55 995,00
2006	24 085,00


Utiliser le clic droit pour imprimer, exporter... vos enregistrements.

Exporter vers un fichier Hyper File...

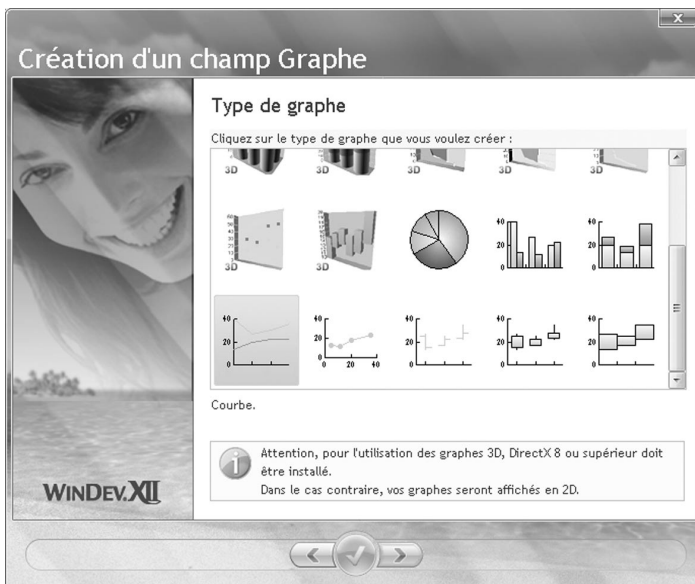
Fermer

► Nous allons maintenant créer le champ Graphe permettant d'afficher les données de la requête.

1. Créer une nouvelle fenêtre.

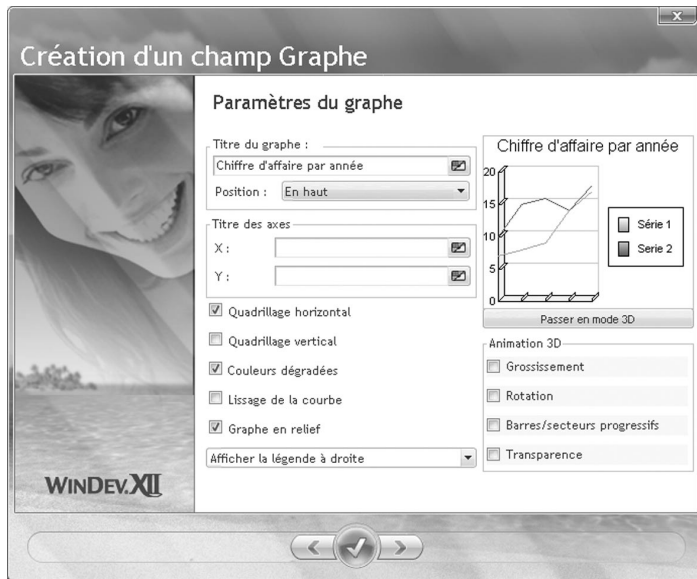
2. Dans cette fenêtre, créez un champ Graphe (cliquez sur le bouton ). L'assistant de création de graphe se lance.

3. Sélectionnez le type du graphe à créer (par exemple courbe)



4. Donnez les caractéristiques du graphe :

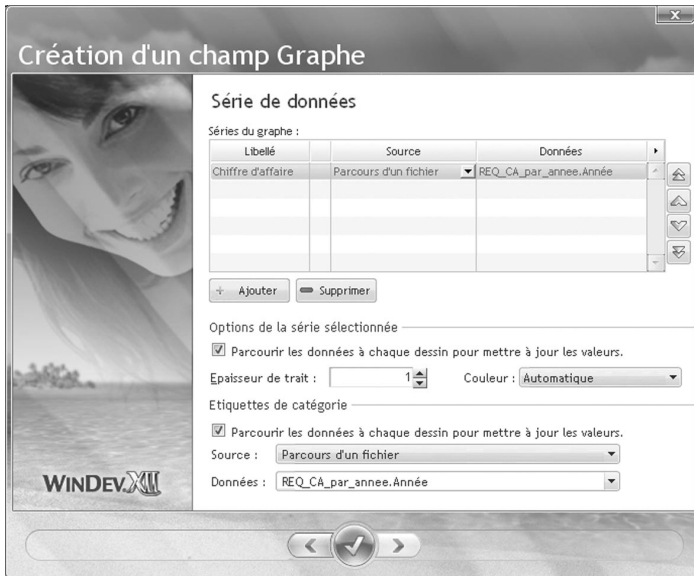
- Titre du graphe : Chiffre d'affaires par année
- Quadrillage horizontal
- Couleurs dégradées
- Afficher la légende à droite



5. Indiquez les séries du graphe. Dans notre cas, nous utiliserons une seule série.

- Cliquez sur la première ligne du tableau.
- Indiquez le libellé : "Chiffre d'affaires".
- Indiquez la source : sélectionnez l'option "Parcours d'un fichier". Cette option permet de sélectionner le fichier ou la requête qui sera la base de la série.
- Indiquez les données : cliquez dans la colonne "Données". Sélectionnez la requête "REQ_CA_par_Année", puis la rubrique : la somme totale vente. Dans la rubrique de parcours, sélectionnez "Année".
- Indiquez l'étiquette de catégorie à utiliser : source correspond à "Parcours d'un fichier" et

données correspond à la rubrique Année de la requête REQ_CA_Par_Année. I



6. Donnez le nom du graphe et validez.
7. Redimensionnez le champ graphe dans votre fenêtre et testez.

Les fonctions graphes

WinDev propose un ensemble de fonctions pour créer et configurer un graphe. Ces fonctions commencent toutes par les lettres "gr". Ces fonctions peuvent être utilisées :

- dans un champ image. Le graphe est entièrement dessiné à l'aide des fonctions grXXX.
- dans un champ graphe. Les fonctions permettent alors de modifier par programmation certaines caractéristiques définies dans le champ graphe.

Comme pour le champ Graphe, nous avons préparé une fenêtre permettant de dessiner un graphe entièrement par programmation.

- ▶ Ouvrez le projet "Grapheur". Ce projet est disponible dans le sous-répertoire "Autoformation\Corrigés\Grapheur" de WinDev.
- ▶ Ouvrez la fenêtre "Grapheur" et testez-la. Cette fenêtre présente les différents graphes, avec les options correspondantes.
- ▶ Le code du bouton "Graphe" contient toutes les fonctions nécessaires au dessin du graphe. Pour plus de détails sur ces fonctions, nous vous conseillons de consulter l'aide en ligne (mot-clé : "Graphe").

LEÇON 8.12. LA POO

Ce que vous allez apprendre dans cette leçon ...

- Concepts de la programmation orientée objet.
- Exemples de déclaration d'objet.



Durée estimée : 30mn

Concepts

La Programmation Orientée Objet (P.O.O.) a pour but de permettre une meilleure réutilisabilité du code. Les programmes développés en POO sont structurés : ils sont décomposés en modules gérant chacun une fonctionnalité du logiciel. Les modules développés peuvent être facilement réutilisés dans d'autres logiciels. Ils vont regrouper un ensemble de procédures et encapsuler la structure de données sur laquelle les procédures vont agir.

Pour programmer "en objet", il faut déclarer des classes, les membres et les méthodes associés.

Les classes

Une **classe** rassemble la description d'une structure de données (les membres) et les procédures (méthodes) qui manipulent les membres.

Une classe définit donc un type de données et son comportement.

Les objets

Une classe permet de créer des **objets**. Chaque objet créé possède des membres décrits dans sa classe et peut être manipulé par l'intermédiaire des méthodes de sa classe. On dit qu'un objet est une instance de la classe.

Lorsque la classe est déclarée, il suffit d'associer un objet à une classe pour que l'objet puisse être manipulé par toutes les méthodes de cette classe.

Les membres

Un **membre** est une donnée (ou paramètre) de l'objet.

Les méthodes

Une **méthode** permet d'agir sur l'objet, pour modifier ses membres par exemple.

Une méthode est une procédure. Son fonctionnement est identique à celui des procédures classiques du WLangage.

Notion d'héritage

L'**héritage** permet d'inclure les caractéristiques d'une classe existante (*classe de base*) dans une nouvelle classe (*classe dérivée*). L'héritage permet de créer un nouveau type de données à partir d'un type connu, dans le but de lui ajouter des fonctionnalités, ou d'en modifier le comportement. La classe de base ne sera donc pas modifiée. Une classe peut hériter d'une classe dont elle devient une sous-classe.

Une *classe dérivée* hérite des membres et des méthodes de sa *classe mère* (qui peut, elle-même, être une sous-classe d'une autre classe mère), en plus de ses propres membres et méthodes (et également des membres et méthodes de la première classe mère, s'il s'agit d'un héritage multiple). Il n'est pas nécessaire de dupliquer les membres et méthodes de la ou des classe(s) mère(s).

Constructeur et Destructeur

La notion de **Constructeur** et **Destructeur** est importante puisqu'elle permet un appel automatique de méthodes d'initialisation lors de la création d'un objet et lors de sa destruction.

La méthode Constructeur associée à une classe est automatiquement appelée lors de la déclaration d'un objet de la classe.

La méthode Destructeur associée à une classe est automatiquement appelée lors de la suppression de l'objet (sortie de procédure dans laquelle l'objet a été déclaré).

Encapsulation de données

L'encapsulation des données permet de garantir que les données membres de l'objet ne seront pas modifiées à tort par des fonctions (méthodes) extérieures à l'objet.

Il est ainsi possible d'interdire à l'utilisateur d'un objet l'accès à certain ou à tous ses membres. Les membres dont l'accès est interdit sont appelés membres privés.

Il n'est possible d'accéder à ces membres privés qu'à partir des méthodes prévues à cet effet dans la classe.

Créer un programme orienté objet

Pour créer un programme en objet en WLangage, il faut :

1. Décrire la classe et les membres de la classe
2. Donner toutes les méthodes de la classe
3. Déclarer les objets en leur associant une classe ("instancier une classe")

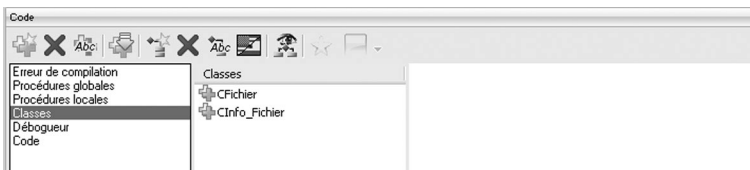
- ▶ Ouvrez le projet "Programmation.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\Programmation" de WinDev 12.
- ▶ Ouvrez la fenêtre "FEN_Poo1.WDW".

Nous ne détaillons pas ici la syntaxe de la P.O.O., nous présenterons uniquement un exemple simple de programme orienté objet. Pour avoir plus de détail sur la syntaxe de la P.O.O., consultez l'aide en ligne et le manuel de programmation de WinDev (mot-clé : "POO, Classe").

Déclarer une classe

Pour créer une classe :

1. Affichez le volet "Code".
2. Choisissez "Classe". Dans le cadre qui apparaît, faites un clic droit puis sélectionnez l'option "Nouvelle classe".
3. Saisissez un nom et validez par [Entrée].



Une classe est de la forme :

```
CFichier est une classe
  NomLong est une chaîne
  NomCourt est une chaîne
  Extension est une chaîne
  CheminCourt est une chaîne
  CheminLong est une chaîne
FIN
```

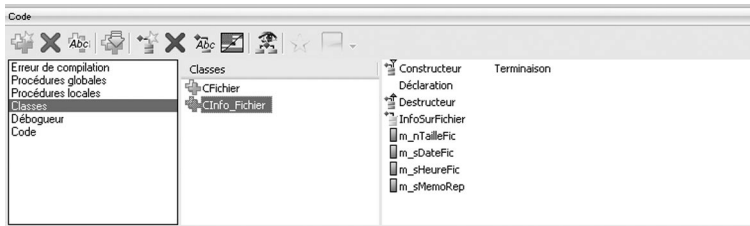
"Fichier" est le nom de la classe.

"NomLong", "NomCourt", "Extension"... sont des membres de la classe.

Décrire les méthodes

Pour créer une méthode :

1. Cliquez avec le bouton droit de la souris sur votre classe présente dans le volet "Code".
2. Choisissez "Nouvelle méthode", en bas du menu qui apparaît.
3. Saisissez un nom et validez par [Entrée].



Une méthode est de la forme :

```
//Méthode SelectionFichier
//Cette méthode renvoie un résultat
PROCEDURE CFichier::SelectionFichier()
Res est une chaîne
Res = fSélecteur(fRepEnCours(), "*.BMP", ...
    "Sélection des images BMP", "Images" + TAB + ...
    "*.BMP", "BMP")
:NomLong = fNomLong(Res)
:NomCourt = fNomCourt(Res)
:Extension = ExtraitChaîne(fNomLong(Res), 2, ".")
//Chemin long
Pos est un entier
Pos = Position(Majuscule(Res), Majuscule(:NomLong))
:CheminLong = Res[[1 à Pos-2]]
//Chemin Court
Ch1 est une chaîne = ""
Ch2 est une chaîne = ""
//Extraire le lecteur
Pos = Position(Res, "\\")
Ch1 = Res[[1 à Pos-1]]
Ch2 = Ch1 + "\\"
//Extraire chaque sous-répertoire
BOUCLE
    Pos = Position(Res, "\\", Pos+1)
    Ch1 = Res[[1 à Pos-1]]
    SI Pos = 0 ALORS SORTIR
    Ch2 = Ch2 + fNomCourt(Ch1) + "\\"
FIN
:Chemincourt=Ch2[[1 à Taille(Ch2)-1]]
RENOYER Res
```

Le nom de la méthode est "SelectionFichier".

"Fichier::SelectionFichier" signifie que la méthode "SelectionFichier" appartient à la classe "Fichier".

Pour faire référence à un membre de l'objet "Fichier", il faut utiliser la syntaxe

```
//Depuis la classe Fichier
:<Nom du membre>
//Depuis le programme appelant
<NomObjet>:<nom du membre>
```

Déclarer et manipuler les objets

Dans les traitements de la fenêtre, un objet est déclaré en même temps que les autres variables :

```
GLOBAL
    Fic1 est un objet CFichier
```

L'objet est manipulé de la façon suivante :

```
//appel de la méthode
Ch = Fic1:SelectionFichier()
SI Ch = "" ALORS RETOUR
    Infocic..Etat = Grisé
    NOML = Fic1:NomLong
    NOMC = Fic1:NomCourt
    EXTENS = Fic1:Extension
    CHEMINL = Fic1:CheminLong
    CHEMINC = Fic1:CheminCourt
    TailleF = Fic1:TailleFichier()
    DateF = Fic1:DateFichier()
    HeureF = Fic1:HeureFichier()
```

- ▶ Ouvrez la fenêtre "FEN_Poo2.WDW" et testez cette fenêtre.
Cette fenêtre présente un exemple de programme objet manipulant les héritages et les méthodes Constructeur.

Nous ne nous attarderons pas plus sur la présentation de la POO dans ce cours.

DÉVELOPPEZ 10 FOIS PLUS VITE

PARTIE 9

Gestion de projet avancée

XIII

Express

LEÇON 9.1. TABLEAU DE BORD

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Mode de visualisation
- Options du tableau de bord



Durée estimée : 20mn

Présentation

Le tableau de bord du projet est un élément indispensable à la gestion de projets WinDev. Le tableau de bord permet d'avoir une vision globale et synthétique de l'état d'avancement d'un projet.

Nous avons déjà vu dans la partie 2, quelques fonctionnalités du Tableau de bord.

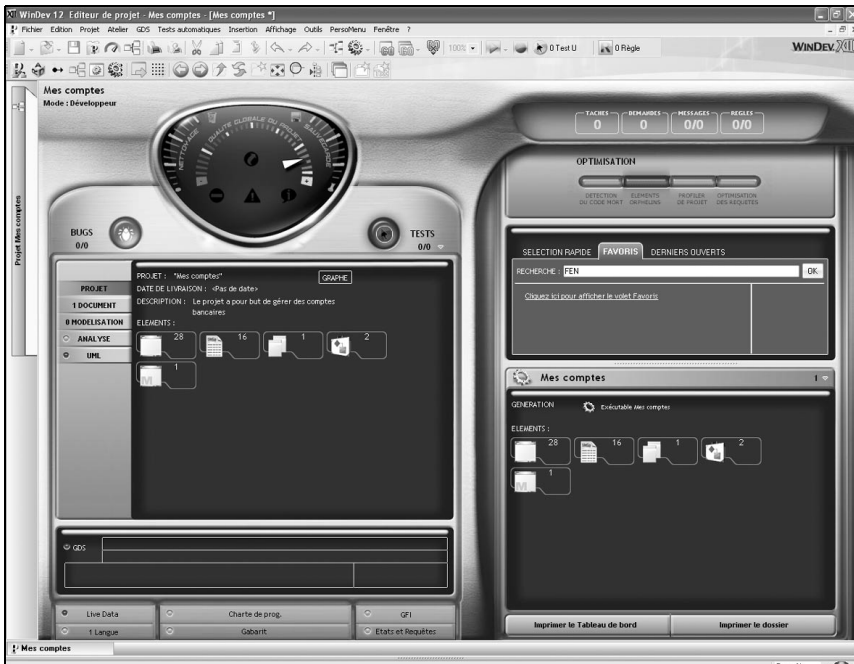
Nous allons maintenant l'étudier plus en détail, et voir ses interactions avec les centres de contrôles.

Pour cela, nous allons reprendre le projet "Mes Comptes". Si vous n'avez pas créé cette application, une version corrigée est disponible dans le sous-répertoire "\\Autoformation\Corrigés\Mes Comptes" de WinDev 12.

Les différents éléments du tableau de bord

Le tableau de bord du projet est composé :

- de voyants : Si les voyants sont verts, tout va bien. Les voyants rouges signalent un problème potentiel.
- de listes d'éléments, permettant un accès rapide aux principales options du projet.
- de compteurs, permettant de gérer les nouveautés, les demandes, ...



Les listes

Le tableau de bord est composé de différentes listes. Ces listes permettent d'optimiser l'accès aux éléments du projet.

Par exemple, la liste "Sélection rapide" permet de rechercher facilement un élément du projet. Il suffit de saisir quelques lettres composant le nom de l'élément recherché. La sélection est automatiquement réalisée et un aperçu permet de choisir l'élément voulu.



Pour rechercher la fenêtre FEN_PersonneX, rien de plus simple : saisissez les lettres "FEN_Per" et faites votre choix. Un double-clic sur le nom de l'élément permet de l'ouvrir immédiatement sous l'éditeur associé.



Astuce

Vous voulez rechercher un élément sans être dans le tableau de bord ? Utilisez la combinaison de touches [CTRL] + [E] ... et vous retrouverez les mêmes possibilités.

Autre liste intéressante : les favoris. Qui dans son application n'a pas une fenêtre ou un code qui sont utilisés fréquemment ? Une fenêtre générale regroupant par exemple les principales fonctionnalités ? Pour ne pas perdre de temps à rechercher cet objet, il suffit de le mettre dans les favoris. Par exemple, nous allons mettre la fenêtre "FEN_PersonneX" de notre application en favoris :

1. Dans le tableau de bord, sélectionnez l'onglet "Favoris", et cliquez sur le lien pour afficher le volet des favoris.



Note

Le volet des favoris peut également être affiché par l'option "Affichage .. Barre d'outils .. Favoris".

2. Affichez la fenêtre "FEN_PersonneX" sous l'éditeur (par exemple en double-cliquant sur son nom dans le volet WinDev).

3. Dans le volet "Favoris", cliquez sur le bouton "+". Vous pouvez saisir le nom du favori (par

exemple MaFenêtre).



Validez. Le favori apparaît



Note

Si plusieurs développeurs travaillent sur le même projet, il est possible de visualiser "Tous les favoris".

4. Dans le tableau de bord, vous retrouvez également votre favori.

Les voyants

Dans la partie 2, nous avons vu le voyant concernant les tests. Regardons les autres voyants.

Les informations concernant la **qualité globale du projet** sont regroupées dans un compteur spécifique.



Dans le compteur, vous trouvez :

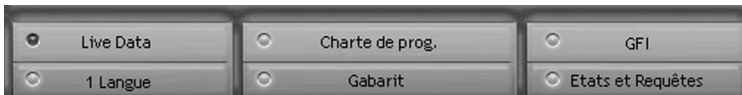
- Un indicateur de qualité globale du projet,
- Une jauge de sauvegarde. Il suffit de cliquer sur la jauge pour lancer une sauvegarde du projet.
- Une jauge de nettoyage. Il suffit de cliquer sur la jauge pour lancer le nettoyage du projet.
- Des voyants d'avertissement : composant pas à jour, erreurs de compilations détectées, informations détectées, warnings détectés.

Le tableau de bord présente également des voyants d'optimisation :

- **détection de code mort** : Si ce voyant est rouge, cela signifie que des procédures de votre application sont inutilisées.
- **éléments orphelins** : Si ce voyant est rouge, cela signifie que des éléments de votre projet ne sont pas utilisés (par exemple, fenêtre utilisée uniquement pour test lors du développement de l'application).

- **profil de projet** : Si ce voyant est rouge, cela signifie que vous n'avez jamais lancé l'analyseur de performances sur votre application. Cliquez sur ce voyant : l'analyseur de performances est automatiquement lancé. Utilisez normalement votre application. À la fermeture de votre application, un rapport complet des actions effectuées sera affiché. Vous trouverez peut-être des traitements à optimiser. Nous détaillerons l'analyseur de performances dans le chapitre "L'analyseur de performances", page 482.
- **optimisation de requêtes** : Si ce voyant est rouge, cela signifie que votre projet utilise des requêtes qui peuvent être optimisées par une simple modification de l'analyse (ajout d'une clé composée par exemple). N'hésitez pas à utiliser ce bouton : il ne peut qu'améliorer les performances de votre application.

Par le biais du tableau de bord, il est également possible de connaître et de modifier certaines caractéristiques importantes du projet. Ces éléments sont également présentés sous forme de voyants : le voyant est vert ? Cela signifie que la fonctionnalité est activée dans votre projet. Dans le cas contraire, un simple clic sur le voyant permet d'activer ou de configurer la fonctionnalité. Passons rapidement en revue les éléments disponibles :



- **GDS** : Permet de savoir si le projet est partagé via le Gestionnaire de Sources. Nous verrons plus en détail cette fonctionnalité dans la leçon suivante ("GDS", page 436).
WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.
- **Live data** : Permet d'utiliser le contenu des fichiers de données du poste de développement dans les fenêtres, les états, ... manipulés sous l'éditeur. Cette fonctionnalité est très intéressante par exemple pour définir la taille des champs présents dans une fenêtre.
- **Charte de programmation** : Nous l'avons vu dans la partie 1, la charte de programmation permet d'unifier les noms des éléments utilisés dans un projet. Comme notre projet utilise une charte de programmation, ce voyant est vert.
- **GFI (Grisage automatique des fenêtres inaccessibles)** : Cette option indique si les fenêtres inaccessibles de votre application seront automatiquement grisées ou non. Cette fonctionnalité est souvent très appréciée des utilisateurs. Lorsque plusieurs fenêtres sont superposées, les fenêtres inaccessibles sont automatiquement grisées.
- **Langues** : vous trouvez ici, le nombre de langues gérées par votre projet. Dans notre cas, une seule langue est gérée. Un clic sur ce voyant vous permet d'ajouter simplement de nouvelles langues à votre projet. Nous étudierons la gestion des projets multi-langues un peu plus loin dans ce cours ("Multilingue", page 507)
- **Gabarit** : Vous le savez, les gabarits permettent de définir simplement la charte graphique de votre application. Un clic sur ce voyant vous permet d'accéder à l'écran de configuration de gabarit de votre application.
- **Etats et requêtes** : Nous l'avons déjà vu, le logiciel "Etats et Requêtes" permet à l'utilisateur final de composer ses propres états et ses propres requêtes. Pour l'intégrer dans votre application : un voyant suffit. Pour plus de détails, consultez "Diffusez "Etats et Requêtes" avec vos applications", page 366.
WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

Les compteurs en relation avec les centres de contrôle

WinDev est livré avec plusieurs Centres de Contrôle. Les Centres de Contrôle, comme leur nom l'indique, permettent de contrôler un ensemble de fonctionnalités. Nous avons déjà vu le Centre de Contrôle Hyper File, qui permet notamment de contrôler les applications et les fichiers de données des applications Hyper File Client / Serveur.

WinDev propose également le Centre de Suivi de projets. Ce centre permet de gérer des projets (informatiques ou non), de leur conception à leur livraison et même leur maintenance.

Nous ne détaillerons ici que les éléments en relation avec le tableau de bord. Pour plus de détails sur les Centres de Contrôle et leur configuration, consultez le chapitre "Centres de contrôles", page 447.

Le compteur de bugs

Le compteur de bugs représente le nombre de bugs actuellement présents sur le projet. Ces bugs sont référencés dans le Centre de Suivi Qualité et Evolutions.

Ces bugs ont été fournis directement par les utilisateurs de l'application, au moyen de l'option "? .. Envoyer une suggestion, ..." de leur application. Cette option est proposée en standard si vous intégrez le menu automatique "?" dans votre application.

WinDev 12 version "Express" : L'envoi de suggestions n'est pas disponible dans cette version d'évaluation.

Le compteur de tâches

Le compteur de tâches représente le nombre de tâches actuellement en cours sur le projet. Ces tâches sont référencées dans le Centre de Suivi de Projets.

Ces tâches correspondent par exemple à :

- des fonctionnalités à ajouter au projet
- des développements futurs, ...

Le compteur de demandes

Le compteur de demandes représente le nombre de demandes actuellement présentes sur le projet. Ces demandes sont référencées dans le Centre de Suivi Qualité et Evolutions.

Ces demandes ont été fournies directement par les utilisateurs de l'application, au moyen de l'option "? .. Envoyer une suggestion, ..." de leur application. Cette option est proposée en standard si vous intégrez le menu automatique "?" dans votre application.

WinDev 12 version "Express" : L'envoi de suggestions n'est pas disponible dans cette version d'évaluation.

Le compteur de messages

Le compteur de messages représente le nombre de messages associés au projet. Ces messages sont visibles dans la messagerie de WinDev. Ces messages sont par exemple automatiquement envoyés lors du travail avec le gestionnaire de sources. Vous êtes ainsi automatiquement averti d'une mise à jour d'un élément dans le GDS.

Le compteur de règles

Le compteur de règles permet de connaître le nombre de règles métier concernant le projet. Nous avons déjà créé des règles métier dans le début de ce cours. Les règles métier peuvent être saisies :

- par vous, développeur de l'application
- par les utilisateurs, au moyen de l'option "? .. Envoyer une suggestion, ..." de leur application. Cette option est proposée en standard si vous intégrez le menu automatique "?" dans votre application.

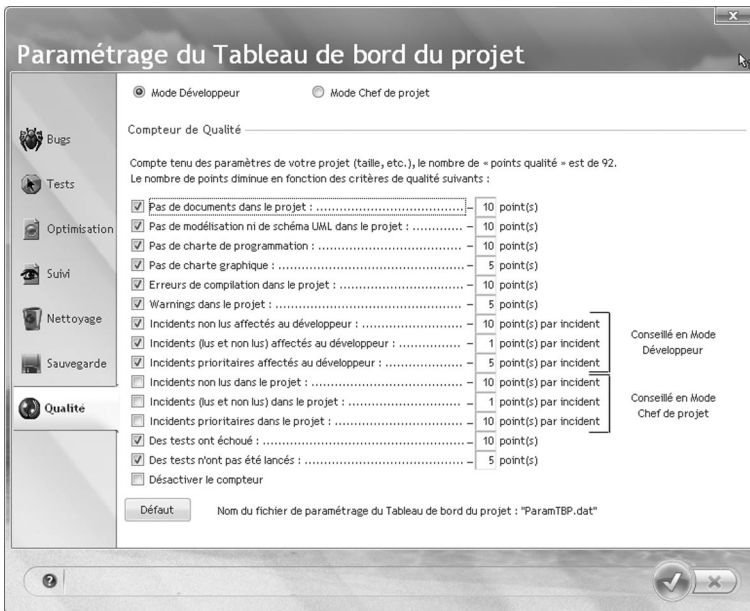
WinDev 12 version "Express" : L'envoi de suggestions n'est pas disponible dans cette version d'évaluation.

Configuration du tableau de bord

Nous avons vu les grandes catégories d'informations affichées dans le tableau de bord. WinDev propose deux modes de visualisation de ces informations : mode développeur, et mode chef de projet. Pour changer de mode, il suffit de cliquer sur "Mode" en haut du tableau de bord.

Pour chaque mode, vous pouvez configurer de manière précise les niveaux d'alerte du tableau de bord.

Pour configurer le tableau de bord, sélectionnez l'option "Option du tableau de bord" du menu contextuel du tableau de bord.



Il est possible de configurer précisément pour les deux modes :

- les voyants concernant les bugs
- les voyants concernant les tests
- les voyants concernant l'optimisation
- les compteurs de tâches et de demandes
- le nettoyage
- la sauvegarde
- la qualité du projet.



Note

Le look de votre tableau de bord vous semble démodé ? Essayez un nouveau look grâce à l'option "Apparence" du menu contextuel du tableau de bord.

LEÇON 9.2. GDS

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Le gestionnaire de sources
- Utilisation du gestionnaire de sources



Durée estimée : 30mn

WinDev 12 version "Express" :

Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

Introduction

Un développement informatique important nécessite la participation de plusieurs développeurs. Ces développeurs doivent travailler sur un projet WinDev unique, en partageant les différentes ressources (fenêtres, classes, ...) manipulées.

WinDev gère en standard :

- un gestionnaire de sources nommé "GDS" permettant de partager les sources de différents projets entre développeurs et de connaître l'historique complet des modifications réalisées (dans le code, l'interface, ...).
- un mode de développement nommé "Groupware développeur".

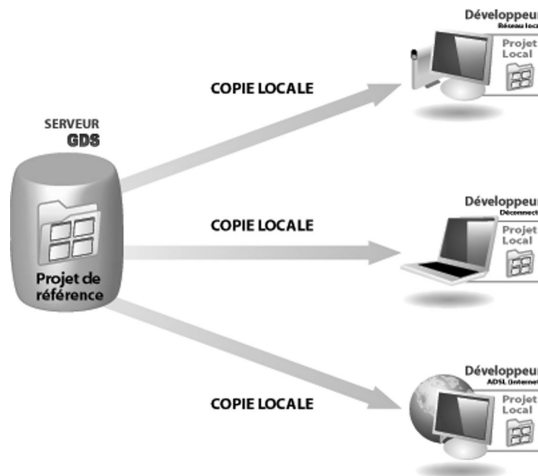
GDS

Principe du GDS

Le gestionnaire de sources permet de stocker et de partager des projets et tous leurs éléments.

Le principe est le suivant :

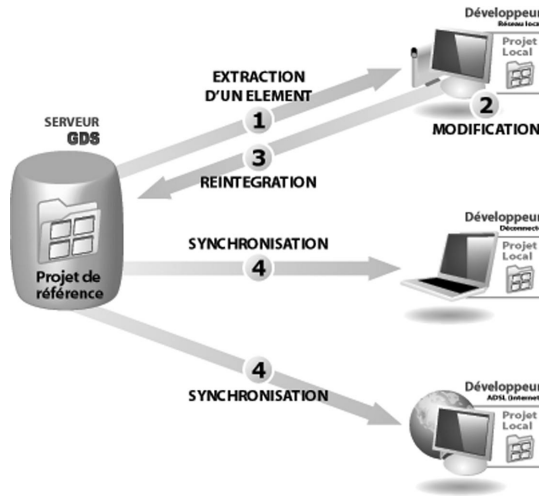
- Une version de référence de chacun de vos projets est présente sur un serveur. L'ensemble de ces versions est appelé "Base de sources".
- Chaque développeur a sur son poste une copie locale des différents projets sur laquelle il travaille.



- Lorsqu'un développeur veut modifier un élément d'un projet (une fenêtre, un état, une requête, ...), il indique au GDS qu'il s'approprie temporairement cet élément. Pour cela, le développeur va extraire cet élément de la base de sources.
- Ce développeur obtient alors les droits exclusifs sur cet élément : il peut faire toutes les modifications voulues sur cet élément.
- Les autres développeurs continuent de travailler avec la copie de la version de référence de cet élément (présente sur la base de sources).
- Dès que le développeur a terminé ses modifications, il réintègre l'élément extrait dans la

base de sources.

- Les autres développeurs sont automatiquement avertis de cette réintégration. Ils peuvent alors mettre à jour leur copie locale.



Le GDS gère donc le travail collaboratif et permet de connaître l'historique de toutes les modifications réalisées.

Le GDS permet également d'administrer et de régler les éléments partagés entre plusieurs projets.

Création de la base de sources

Pour partager un projet grâce au gestionnaire de sources, il est nécessaire de créer une base de sources. Cette base de sources doit être créée **une seule fois** sur un poste serveur.

Cette base de sources peut être créée à différents moments :

- lors de l'installation de WinDev.
- lors de la création d'un projet utilisant le GDS.
- lors de l'importation d'un projet dans le gestionnaire de sources.
- directement dans l'administrateur du GDS.

- ▶ Nous allons créer notre base de sources lors de l'importation d'un projet dans le GDS, à la prochaine étape.



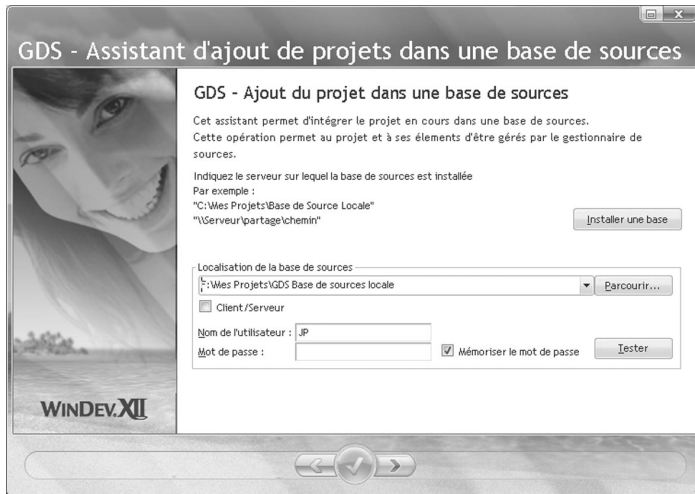
Notes

Il est conseillé de faire régulièrement des sauvegardes de la base de sources du GDS. Pour cela, il suffit de se connecter en tant qu'administrateur à l'outil d'administration du GDS, et de sélectionner l'option "Outils .. Administration .. Sauvegarde complète de la base".

Intégration d'un projet dans le GDS

Pour utiliser un projet existant avec le gestionnaire de sources, il suffit d'intégrer ce projet dans la base de sources du GDS.

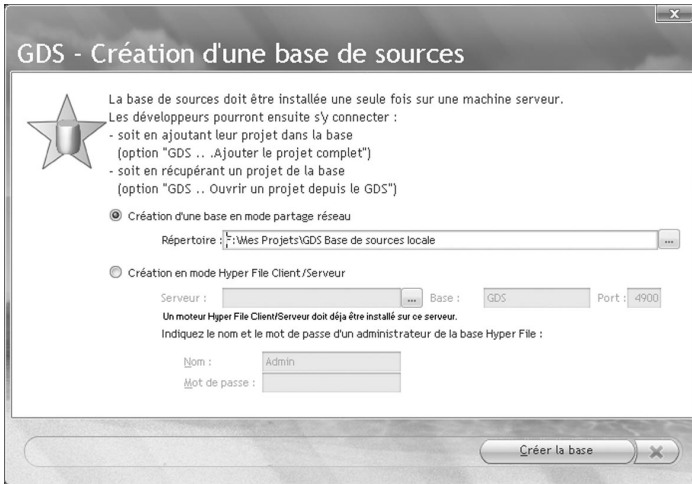
- ▶ Nous allons intégrer le projet "Fenêtres et Champs.WDP" dans la base de sources du GDS :
 1. Ouvrez le projet "Fenêtres et Champs.WDP". Ce projet est présent dans le sous-répertoire "Autoformation\Corrigés\Fenêtres et champs" de WinDev.
 2. Sélectionnez l'option "GDS .. Ajouter le projet complet". L'assistant d'ajout de projet dans le GDS se lance :



Nous n'avons pas précédemment créé de base de sources. Nous allons donc en créer une. Remarque : Nous allons créer une base de sources "locale" (sur le poste de développement). Le fonctionnement serait similaire pour une base de sources réseau.

3. Cliquez sur le bouton "Installer une base". La base de sources peut être au format Hyper File Classic (local ou réseau) ou au format Hyper File Client / Serveur. Nous allons créer une base

de sources au format Hyper File Classic.



Notes

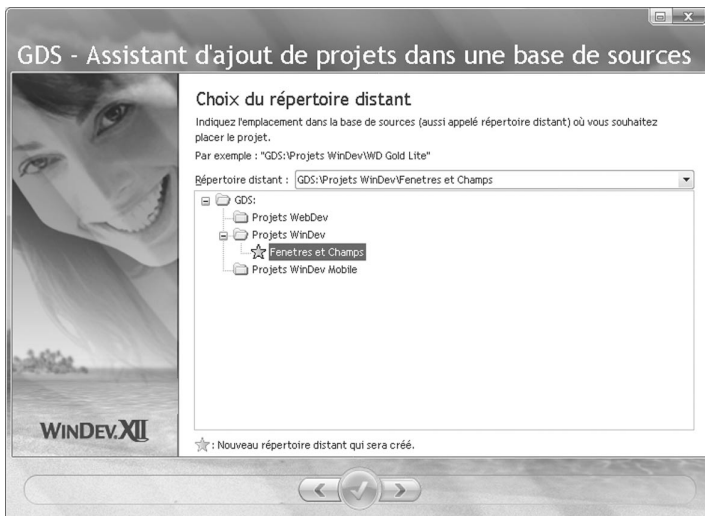
L'utilisation d'une base de sources au format Hyper File Client / Serveur permet d'utiliser cette base de sources à distance.

4. Conservez l'option "Création d'une base en mode partage réseau" sélectionnée et indiquez le répertoire de cette base de sources ("C:\Mes Projets\Ma base de sources" par exemple). Validez la création de la base de sources (bouton "Créer").

La base de sources est à présent créée. Nous allons pouvoir intégrer notre projet dans cette base de sources.

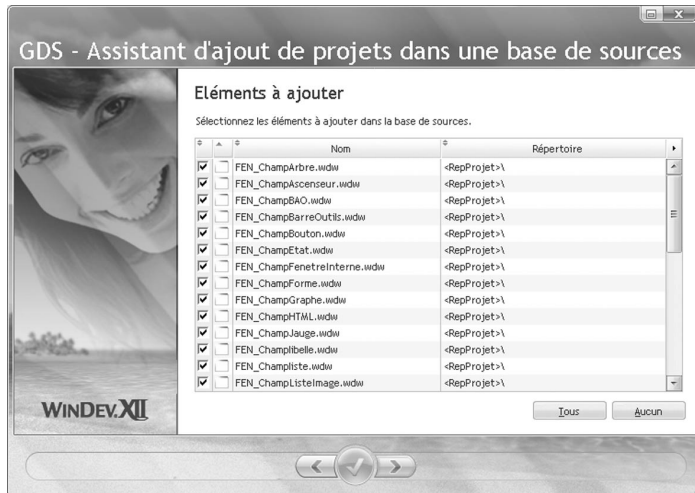
5. Passez à l'écran suivant.

6. L'assistant nous propose de placer le projet dans le sous-répertoire "Projets WinDev" de la base de sources.



Nous allons accepter cet emplacement. Passez à l'écran suivant.

7. L'assistant nous demande de sélectionner les différents éléments du projet à ajouter dans la base de sources.



Nous voulons que tous les éléments du projet soient ajoutés. Passez à l'écran suivant.

8. L'assistant nous demande de sélectionner les différentes dépendances du projet à ajouter dans la base de sources. Ces dépendances correspondent à tous les éléments externes nécessaires au projet (images, feuilles de styles, ...).

Nous voulons que toutes les dépendances du projet soient ajoutées. Passez à l'écran suivant.

9. Validez l'intégration du projet dans le GDS. Notre projet et tous ses éléments sont désormais ajoutés dans notre base de sources.

Une fenêtre d'aide sur la barre d'outils d'utilisation du GDS s'affiche. Lisez et validez cette fenêtre.

Vous avez ajouté le projet dans le gestionnaire de sources, vous devenez donc le maître du projet.



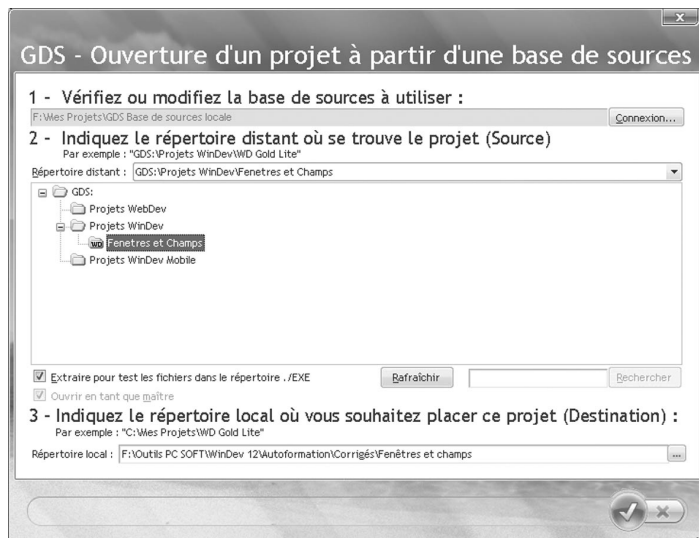
Notes

Partage d'éléments du projet

Lors de l'intégration dans le GDS de projets partageant les mêmes ressources (la même analyse, les mêmes fenêtres, ...), les éléments concernés peuvent être partagés entre les différents projets. Ainsi, le même élément n'est intégré qu'une seule fois dans le GDS.

Ouverture du projet depuis le GDS

Pour travailler sur un projet présent dans le gestionnaire de sources, il faut récupérer, en local, une copie de ce projet. Pour cela, il suffit d'ouvrir le projet depuis le gestionnaire de sources (option "GDS .. Ouvrir un projet depuis le GDS").



Cette opération doit être effectuée une seule fois par chaque développeur utilisant le projet.

Le développeur qui a ajouté le projet dans le gestionnaire de sources (dans ce cas, c'est vous !) n'a aucune manipulation à effectuer.



Notes

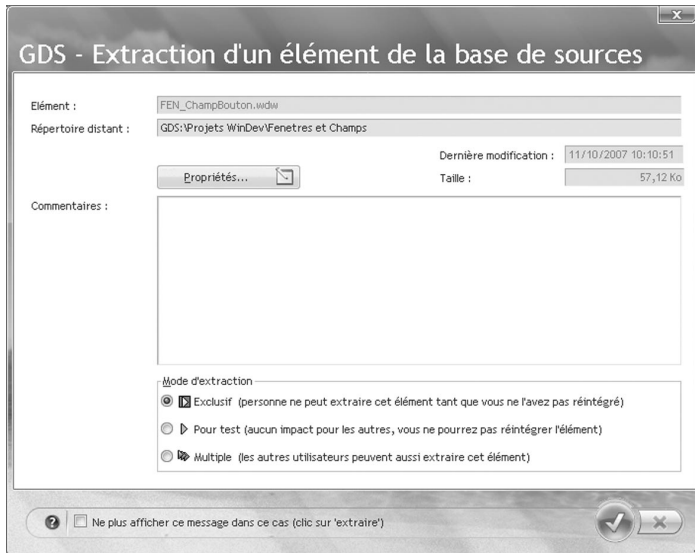
Les prochaines ouvertures d'un projet géré par le GDS sont identiques aux ouvertures d'un projet non géré par le GDS : il suffit d'ouvrir le projet (fichier ".WDP") correspondant à la copie locale.

Extraction d'un élément

Pour modifier un élément du projet, il est nécessaire de l'extraire.

- ▶ Nous allons extraire la fenêtre "FEN_ChampBouton" :
 1. Sélectionnez la fenêtre "FEN_ChampBouton" dans le graphe du projet.
 2. Sélectionnez l'option "Extraire" du menu contextuel.
 3. Saisissez un commentaire ("Modification des libellés des champs" par exemple). Ce commentaire sera utile pour les autres développeurs.
 4. Le GDS propose trois modes d'extraction :
 - **Exclusif** (mode conseillé) : personne ne pourra extraire cet élément jusqu'à sa réintégration. L'élément pourra uniquement être extrait pour test.
 - **Pour test** : l'élément pourra être modifié mais les modifications ne pourront pas être réintégrées.

- **Multiple** : l'élément pourra également être extrait par d'autres utilisateurs. Dans ce cas, lors de la réintégration, il sera possible de visualiser les différences entre les différentes versions de l'élément. Ce mode est réservé à des cas d'utilisation spécifiques et à des développeurs avancés.



5. Nous allons extraire la fenêtre en mode exclusif. Conservez l'option "Exclusif" cochée et validez l'extraction.

La fenêtre est extraite. Elle est entourée d'un trait rouge sous l'éditeur de projet.

Modification de l'élément extrait

La modification d'un élément extrait (IHM, code, ...) s'effectue comme sur un projet non géré par le GDS.

Cependant, toutes les modifications effectuées sur un élément extrait ne sont pas visibles par les autres développeurs.

Si un autre développeur exécute l'élément extrait, l'élément qui est actuellement dans la base de sources sera utilisé.

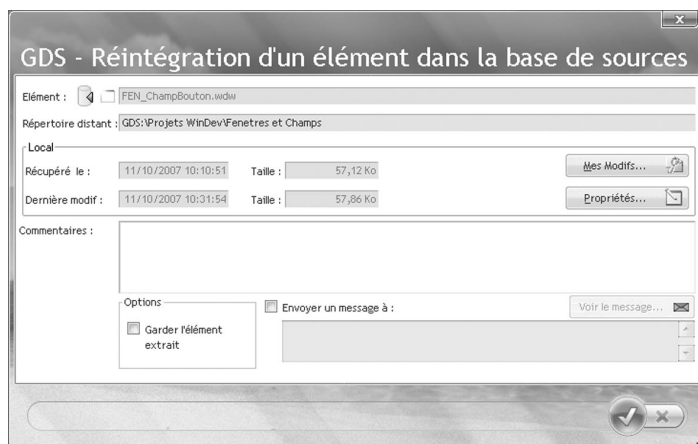
Il est ainsi possible de faire évoluer une application tout en gardant une version stable dans la base de sources.

- ▶ Modifiez la fenêtre extraite. Par exemple, changez l'emplacement des champs, modifiez les libellés, ajoutez des champs ou du code, ...
- ▶ Testez vos modifications.

Réintégration de l'élément extrait

Maintenant que les modifications sont terminées et testées, nous allons réintégrer la fenêtre dans la base de sources. Vos modifications seront alors accessibles aux autres développeurs.

- Sélectionnez l'option "GDS .. Réintégrer l'élément". L'écran suivant s'affiche :



Cet écran permet de :

- connaître les modifications effectuées en comparant l'élément de la base de sources avec l'élément extrait (bouton "Mes Modifs").
- accéder à l'historique de l'élément dans la base de sources (bouton "Propriétés").
- saisir un commentaire sur les modifications réalisées. Par défaut, WinDev propose le commentaire saisi lors de l'extraction.
- envoyer un message aux autres développeurs.
- réintégrer les modifications effectuées dans l'élément, tout en conservant l'élément extrait (option "Garder le fichier extrait").

- Validez la réintégration.

Synchronisation du projet

Plusieurs options permettent de configurer un projet manipulé avec le GDS. Ces options sont regroupées dans l'onglet "GDS/Groupe" de la description du projet (option "Projet .. Description du projet").

Ces options sont les suivantes :

- *Proposer la récupération de la dernière version des éléments à l'ouverture du projet.*
Cette option permet lors de l'ouverture d'un projet présent dans le GDS, de proposer la récupération de la dernière version des différents éléments du projet.
Par défaut, la dernière version des éléments est automatiquement récupérée.
- *Proposer la réintégration des éléments à la fermeture du projet.*
Cette option permet d'afficher lors de la fermeture du projet la liste de tous les éléments

actuellement extraits, afin de réintégrer certains ou tous ces éléments.

Par défaut, à la fermeture du projet, les éléments extraits ne sont pas réintégrés.

- *Proposer la réintégration et la récupération de la dernière version des éléments lors de la génération d'exécutables, de bibliothèques, de composants, ...*

Cette option permet d'afficher lors de la création d'un exécutable, d'une bibliothèque, ... la liste des éléments extraits, afin de réintégrer et de récupérer la dernière version des éléments.

L'exécutable, le composant ou la bibliothèque peut ainsi contenir les derniers éléments les plus à jour.

Par défaut, la génération de l'exécutable, de la bibliothèque, ... se fait avec les éléments du projet actuellement présents sur le poste local.



Notes

Fusion de code

Vous pouvez comparer un élément avec une de ces versions précédentes. Il est alors possible de comparer le code pour récupérer un code "perdu" ou supprimé par erreur par un autre développeur par exemple.

Mode déconnecté (ou nomade)

Le GDS permet très simplement de travailler en mode déconnecté ou nomade.

Ce mode permet par exemple à un développeur utilisant un portable de continuer à travailler sur un projet présent dans la base de sources tout en étant déconnecté.

Le principe est simple :

- avant la déconnexion, il suffit d'utiliser l'option "GDS .. Travail à distance .. Se déconnecter pour une utilisation nomade".
- à la reconnexion, il suffit d'utiliser l'option "GDS .. Travail à distance .. Se reconnecter et synchroniser". Il suffit alors de réintégrer les éléments modifiés.

En mode nomade, deux solutions s'offrent à vous concernant l'extraction d'éléments :

- Vous ne faites aucune extraction d'élément du GDS. Ainsi, pendant toute votre période nomade, d'autres développeurs pourront travailler sur le ou les mêmes éléments que vous. À votre retour et à votre reconnexion au GDS, il sera nécessaire de faire des fusions entre vos propres modifications sur l'élément et celles effectuées par d'autres développeurs.
- Vous faites une extraction exclusive sur le ou les éléments que vous allez modifier. Vous vous réservez ainsi l'élément pendant toute votre période nomade.

Maître et invité

Le GDS considère deux types d'utilisateurs :

- le maître : le maître est l'utilisateur qui a intégré le projet dans le gestionnaire de sources. Généralement, le maître est responsable de la génération de l'exécutable et du programme d'installation.
- les invités : les invités sont les développeurs qui manipulent le projet depuis le gestionnaire de sources.

Il n'est pas obligatoire d'être connecté en mode maître sur un projet. Le mode "Maître" est nécessaire uniquement pour :

- modifier les caractéristiques du projet et réintégrer ces modifications dans la base de sources.
- réintégrer tous les éléments pour faire l'exécutable et le programme d'installation de l'application.

Pour passer du mode maître au mode invité, sélectionnez l'option "GDS .. Gestion du projet .. Devenir invité sur le projet (et tout réintégrer)". Dans ce cas, le gestionnaire de sources propose la réintégration de tous les éléments du projet (y compris le fichier .WDP).

Pour passer du mode invité au mode maître, sélectionnez l'option "GDS .. Gestionnaire du projet .. Devenir maître sur le projet".

Administrateur du GDS

L'administrateur de GDS permet de manipuler directement les différents projets inclus dans le gestionnaire de sources.

Il permet par exemple de :

- gérer les bases de sources (création, connexion à une base de sources).
- gérer les fichiers et les répertoires présents dans un projet de la base de sources (ajouter, supprimer, renommer, ... des fichiers et des répertoires).
- gérer les différents fichiers de la base de sources (extraction, réintégration, partage, ...).
- lancer certains outils (options, maintenances, ...).
- visualiser l'historique d'un élément.
- visualiser l'état des éléments.
- réaliser des sauvegardes.
- donner des droits aux différents utilisateurs du GDS

- ▶ Lancez l'administrateur du GDS (option "GDS .. Administrateur du Gestionnaire de Sources"). Tous les éléments du projet y sont listés.

Pour plus de renseignements sur le GDS, consultez l'aide en ligne de WinDev (mot-clé : "GDS").

Groupware développeur

Lors du développement d'un projet à plusieurs, il est également possible d'utiliser le groupware développeur.

Nous ne détaillerons pas le groupware développeur dans ce guide d'auto-formation.

Pour plus de renseignements sur le groupware développeur, consultez l'aide en ligne de WinDev (mot-clé : "Groupware développeur").

LEÇON 9.3. CENTRES DE CONTRÔLES

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Le centre de Suivi de projets
- Suivre les retours client avec centres de contrôle
- Les autres centres de contrôle



Durée estimée : 20mn

Présentation

WinDev propose un ensemble d'outils pour suivre le développement d'un projet, de sa conception à sa maintenance. Ces outils s'appellent les Centres de Contrôle.

Dans un chapitre précédent, nous avons déjà évoqué le Centre de Contrôle Hyper File. Ce Centre de Contrôle (redistribuable) permet notamment de gérer les applications Hyper File Client / Serveur déployées.

Dans cette partie, nous allons nous intéresser aux Centres de Contrôles liés au développement d'une application :

- le centre de suivi de projets
- le centre de suivi de la qualité et des évolutions

Ces deux centres utilisent une base de données spécifique. Cette base de données peut être :

- soit une base de données Hyper File Classic : au premier lancement de WinDev, vous avez donné le chemin de cette base de données. Par défaut cette base est installée dans un sous-répertoire de WinDev 12.
- soit une base de données Hyper File Client / Serveur.

Ces informations sont modifiables à tout moment depuis les options de WinDev (option "Outils .. Options de WinDev", bouton "Paramètres des Centres de Contrôle").

Le Centre de Suivi de projets

Le Centre de Suivi de projets est le cœur de l'organisation. Le Centre de Suivi de projets permet de gérer le planning de réalisation d'un projet, de sa création à sa livraison finale. Il est ainsi possible de connaître l'ensemble des tâches à réaliser sur un projet, les développeurs concernés, les durées nécessaires, l'état d'avancement du projet, ...

Pour tester les centres de contrôles, nous allons utiliser notre application "Mes Comptes".

- ▶ Lancez le centre de suivi de projets (option "Outils .. Centre de suivi de projets"). La première chose à faire est de définir le "Projet" de travail. Dans notre cas, ce projet correspond à un projet WinDev, mais il pourrait s'agir de n'importe quel autre projet.



Notes

Le Centre de Suivi de projets peut vous assister pour vous aider à gérer, à planifier n'importe quel projet ou tâche (de développement, commercial, ou autre).

1. Créez un nouveau projet (option "Gestion .. Gestion des projets", bouton "Nouveau projet").
2. Saisissez :
 - le nom du projet (par exemple "Autoformation")
 - sa description (par exemple "Mes comptes")
 - le type d'application : Windows
 - un groupe
 - le chemin du projet (celui de votre application)
 - le spool associé (vous pouvez en créer un nouveau). Le spool permettra de récupérer les demandes réalisées par les utilisateurs.
3. L'écran permettant de sélectionner les intervenants de votre projet est automatiquement affiché. Laissez vous-même, et validez.

4. Fermez la fenêtre de gestion des projets.

► Maintenant, il est possible de définir des tâches sur le projet. Nous allons créer une tâche pour passer notre projet en multilingue :

1. Sélectionnez l'option "Tâches .. Créer une nouvelle tâche".

2. Donnez les caractéristiques suivantes :

3. Validez. La tâche apparaît dans la liste des tâches.

Vous pouvez créer autant de tâches que vous voulez. Ces tâches représentent votre planning. Ces tâches peuvent également être des tâches fixes (par exemple une réunion à un jour et une heure donnée) ou périodiques (par exemple, une réunion tous les vendredis).

Pour activer une tâche, il suffit d'utiliser le menu contextuel de la liste des tâches, et sélectionner l'option "Démarrer cette tâche".

L'utilisation du Centre de Suivi de Projets est très simple : dès que vous effectuez une tâche, vous sélectionnez cette tâche dans le centre de suivi de projet, et vous indiquez que vous "démarrerez cette tâche" (option du menu contextuel de la liste des tâches). Si votre projet est lié à un projet WinDev, dès que le projet est ouvert sous WinDev, la tâche correspondante est activée.

Gérer son temps

La gestion du temps est primordiale. Où passe le temps de chaque journée ? Comment le savoir sans contrainte supplémentaire, sans astreindre les membres de l'équipe à une saisie fastidieuse, et sans non plus donner l'impression désagréable d'être espionné ?

La saisie du temps passé sur les tâches est une fonctionnalité intéressante de WinDev. Cette fonctionnalité est liée à l'utilisation du Centre de Suivi de Projets.

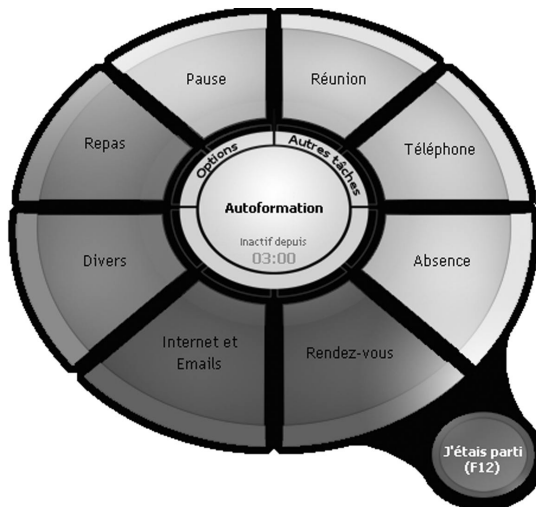
Le principe est simple : la liste des tâches est saisie dans le centre de suivi de projets.

- ▶ Pour activer l'aide à la gestion du temps :
 1. Dans le centre de Suivi de projets, sélectionnez "Paramètres .. options du Centre de Suivi de projets".
 2. Dans l'onglet "Aide à la gestion du temps", sélectionnez "Activer la gestion du temps".
 3. Validez.

La tâche en cours est rappelée à intervalle de temps régulier.



L'utilisateur peut valider la tâche en cours, ou si nécessaire, changer sa tâche en cours. De plus, si le poste de travail est inutilisé pendant une période déterminée, une "roue des tâches" s'affiche. Cette roue des tâches permet de sélectionner la tâche d'interruption correspondant au temps écoulé pendant l'affichage de la roue.



Configuration

Il est possible d'activer / désactiver la gestion du temps dans la fenêtre de paramétrage de la gestion du temps. Cette fenêtre peut être affichée :

- dans le centre de suivi de projets (option "Paramètres .. Options du centre de suivi de projets")
- dans WinDev : option "Projet .. Gestion du temps .. Options de la gestion du temps".

Suivre les retours client avec centres de contrôle

Nous avons vu dans la partie 2 de ce cours, l'intérêt du menu automatique. Une des options de ce menu automatique est l'envoi de suggestions, dysfonctionnements ou règles métier au développeur de l'application.

La gestion des retours clients dépend de l'utilisation en mode connecté ou non de l'application.

Si l'application est en mode connecté (la base de données des centres de contrôle est accessible aux utilisateurs), les demandes sont directement enregistrées dans la base de données des centres des contrôles :

- les suggestions et les dysfonctionnements sont enregistrés dans le spool de l'application.
- les règles métier sont automatiquement ajoutées aux règles métier du projet.

Si l'application est en mode déconnecté (la base de données des centres de contrôle n'est pas accessible aux utilisateurs), les demandes se font par emails. Le centre de Suivi de projets doit être configuré pour recevoir les emails (paramétrage réalisé dans les options administrateur du Centre de Suivi de projet) :

- les suggestions et les dysfonctionnements sont enregistrés dans le spool de l'application.
- les règles métier sont automatiquement ajoutées aux règles métier du projet.



Notes

Pour plus de détails sur la gestion des retours Client, il est conseillé de consulter l'aide en ligne (mot-clé : "Retour Client").

WinDev 12 version "Express" : Cette fonctionnalité n'est pas disponible dans cette version d'évaluation.

Les autres centres de contrôle

WinDev dispose également d'autres centres de contrôle, réservés aux développeurs WinDev :

- Centre de Réutilisabilité : Ce centre permet de centraliser les ressources (composants, bibliothèques, classes, ...) de vos différents projets afin de les mettre à disposition de votre équipe de développement.
- Centre de Gestion des documents : Ce centre permet de centraliser les aides et documents générés pour vos projets.
- Centre Topologique : Ce centre permet de réaliser un schéma topologique d'un site d'exploitation. Il est possible de représenter les serveurs, les postes clients, les périphériques mobiles, ...

LEÇON 9.4. WEB SERVICES

Ce que vous allez apprendre dans cette leçon ...

- Prise de connaissance avec les services Web.
- Etape par étape : utiliser un service Web dans une application.



Durée estimée : 40mn

Introduction

.NET, J2EE, SOAP, WSDL, XML ... que de termes ! Bienvenue dans le monde merveilleux des services Web XML !

Les plates-formes serveurs Microsoft .NET et Sun J2EE permettent d'exporter leurs composants sous forme de services Web XML.

Un service Web est généralement défini comme une application accessible via des protocoles Internet standard. Les services Web permettent notamment l'interactivité entre plusieurs ordinateurs connectés sur le réseau Internet.

Via les services Web, il est possible de faire exécuter des procédures et des traitements sur un serveur Web (.NET ou J2EE) à distance depuis un poste client.

Aspect théorique

Avec WinDev, ces services WEB sont utilisables en tant que client, via le protocole SOAP sur HTTP (le protocole standard d'Internet pour le transfert des pages HTML), grâce aux fonctions **SOAPxx**, **DotNetxx** et **J2EExx**.

Quelle que soit la plate-forme du serveur Web (.NET, J2EE ...), un service Web est accessible via l'utilisation de SOAP.



Notes

Avec WinDev, il n'est pas nécessaire de maîtriser ces informations. Un assistant se charge de ("presque") tout !

Le standard SOAP

Le SOAP (Simple Object Access Protocol) est un protocole de communication qui permet d'exécuter des procédures sur un serveur distant, d'envoyer des paramètres et d'en récupérer. Ce protocole s'appuie principalement sur le protocole HTTP et le langage XML. Notez toutefois que ce protocole peut également s'appuyer sur le protocole SMTP, protocole que nous avons évoqué dans la leçon sur la gestion des emails (page 410).

Les données sont transmises entre le poste local et le serveur SOAP sous forme de texte structuré au format XML (Extensible Markup Language).

Le principal avantage du protocole SOAP vient du fait qu'il repose sur deux protocoles standards :

- XML : pour la structure des messages.
- HTTP : pour le transport des données.

De ce fait, le protocole SOAP est indépendant des systèmes d'exploitation et des langages de programmation. Le protocole SOAP permet "l'interopérabilité".

De plus, l'utilisation du protocole HTTP (pour le transfert des données) permet au protocole SOAP de franchir les "firewalls" (solutions matérielles ou logicielles garantissant la sécurité d'un réseau vis-à-vis d'Internet, appelés également "barrières de sécurité") sans problème.

Il est vrai que le protocole SOAP est relativement complexe à manipuler. C'est pourquoi il existe un standard de description de service, le WSDL, créé pour simplifier l'intégration des services Web XML dans les différents outils qui voudraient les utiliser.

WinDev est capable d'importer un service Web XML grâce à ce standard de description. En bref, avec WinDev :

- Vous sélectionnez le fichier WSDL attaché à un service.
- WinDev génère une classe ou une collection de procédures qui intègre le service. Et voilà, le service Web XML s'utilisera ensuite comme une classe ou une collection de procédures WinDev!

Etape par étape, l'importation d'un service Web

Les paragraphes suivants détaillent l'importation d'un service WEB dans une application WinDev.

Etape 1 : Création du projet

Au lieu de demander à notre grenouille nationale favorite, nous allons créer un projet très simple permettant de connaître la météo à tel ou tel endroit du globe. N'est-ce pas une façon plus moderne ?

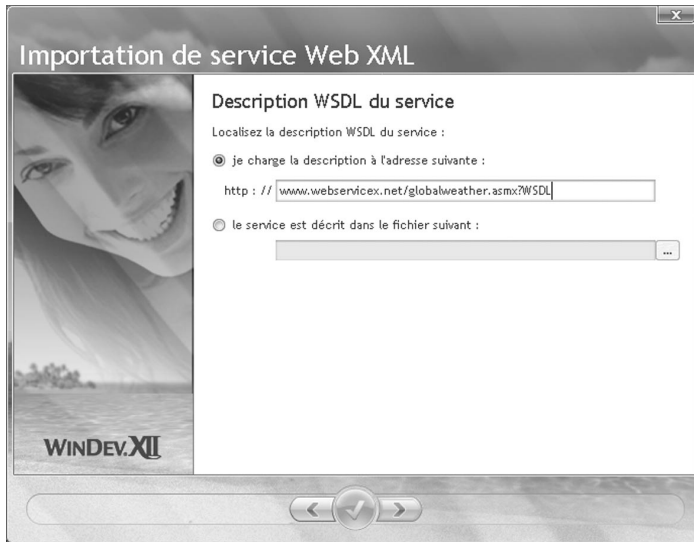
- ▶ Créez un nouveau projet.
Le nom de ce projet sera "METEO". Ce projet n'utilisera pas d'analyse. Conservez toutes les autres options par défaut.
Remarque : le projet entièrement réalisé est disponible dans le sous-répertoire "Autoformation\Corrigés\WEBSservice" du répertoire d'installation de WinDev 12 (projet "METEO.WDP")

Avant de continuer, assurez-vous que vous possédez une connexion active à Internet.

Etape 2 : Importation du service Web

Nous allons commencer par importer la structure du service Web à interroger.

- ▶ Dans le menu de WinDev, sélectionnez l'option "Atelier .. Service Web XML (SOAP, .Net, J2EE) .. Importer et utiliser un service dans ce projet".
L'assistant d'importation de services Web se lance. Passez à l'écran suivant..



- Sélectionnez l'option "je charge la description à l'adresse suivante", et saisissez l'adresse suivante dans le champ de saisie situé au dessous : "www.webservicex.net/globalweather.asmx?WSDL".
En effet, nous allons charger la description du service directement depuis Internet.
Passez à l'écran suivant.



Notes

Le serveur où se trouve le service Web que nous allons utiliser est un serveur de test américain. **Ce serveur peut ne plus exister à l'heure où vous lisez ces lignes.** Nous vous demandons de nous excuser par avance.

Dans le cadre d'une utilisation professionnelle de ce service Web, vous **devez** l'installer sur votre serveur Web.

Pour plus de détails sur ce service, consultez les conditions d'utilisation à l'adresse suivante : <http://www.webservicex.net>

- Sélectionnez le type du service Web : "SOAP générique". Passez à l'écran suivant.



- Vous pouvez générer au choix une collection de procédures ou une classe pour utiliser le service Web dans votre application. Pour cet exemple, choisissez "une collection de procédures" :

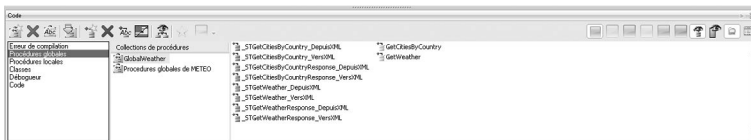


- L'assistant va générer une collection de procédures globales. Une procédure sera créée pour

chaque fonction du service Web :



- ▶ Ne modifiez rien, cliquez sur "Terminer".
WinDev génère la collection correspondante et votre service Web est intégré ! Vous pouvez retrouver la collection de procédures dans le volet "Code".



Prenez quelques minutes pour observer le contenu des procédures générées. Vous retrouvez les fonctions "XML" du WLangage.

Etape 3 : utilisation de la collection de procédures

La collection de procédures générée contient différentes procédures (globales) permettant de dialoguer avec le service Web. Nous allons manipuler quelques-unes de ces procédures.

- ▶ Créez une fenêtre vierge.
 1. Nommez la fenêtre "FEN_InfoMeteo".
 2. Saisissez le titre "Quel temps fait-il ?".
 3. Enregistrez la fenêtre.

- ▶ Ajoutez un champ de saisie. Le nom de ce champ est "SAI_NOMSTATION". Le libellé de ce champ est "Nom de la station :".
- ▶ Ajoutez un champ de saisie. Le nom de ce champ est "SAI_PAYS". Le libellé de ce champ est "Pays".
- ▶ Ajoutez un champ de saisie. Le nom de ce champ est "SAI_TEMPERATURE". Le libellé de ce champ est "Température". Ce champ est de type numérique.
- ▶ Ajoutez un champ de saisie. Le nom de ce champ est "SAI_POINTROSEE". Le libellé de ce champ est "Rosée". Ce champ est de type numérique.
- ▶ Ajoutez un champ de saisie. Le nom de ce champ est "SAI_HUMIDITE". Le libellé de ce champ est "Humidité". Ce champ est de type numérique.
- ▶ Ajoutez un champ de saisie. Le nom de ce champ est "SAI_PRESSION". Le libellé de ce champ est "Pression". Ce champ est de type numérique.

- ▶ Ajoutez un bouton. Le nom de ce bouton est "BTN_METEO". Le libellé de ce bouton est "Quel temps ?".



Notes

Dans notre exemple, le service Web utilise le format XML pour renvoyer les informations. Ces informations seront traitées grâce aux fonctions XML du WLanguage.

Exploitation du service Web

Les mesures sont référencées suivant un code *ICAO* (d'après la documentation du service).

Le code *ICAO* correspond à l'identifiant des villes (des stations), par exemple pour Montpellier c'est le code "LFMT".

Comme personne ne connaît ces codes, le code *ICAO* sera recherché grâce au nom de la station (avec la procédure **GetWeather**).

- ▶ Saisissez le code suivant dans le traitement de clic du bouton "BTN_METEO" :

```

ResultatXML est STGetWeatherResponse

lameteo est STGetWeather
lameteo:m_CityName = SAI_Station
lameteo:m_CountryName = SAI_pays
ResultatXML = GetWeather(SAI_Station, SAI_pays)
ResultatXML:m_GetWeatherResult = ...
                XMLVersTexte (ResultatXML:m_GetWeatherResult)

SI ResultatXML:m_GetWeatherResult = "Data Not Found" ALORS
Erreur("La ville "+SAI_Station+" pour le pays "+SAI_Pays+...
        " n'a pas été trouvée");
    RepriseSaisie(SAI_Station)
SINON

```

La fonction **XMLExtraitChaîne** permet de récupérer la chaîne contenue dans une balise XML. Pour plus de détails sur cette fonction, consultez l'aide en ligne (mot-clé : "XMLExtraitChaîne").

Après exécution de ces lignes, la variable `ResultatXML:m_GetWeatherResult` contient un code XML manipulable facilement grâce à la fonction **XMLExtraitChaîne**.

Après une courte analyse de ce code XML, nous savons que :

- la température est contenue dans la balise "Temperature"
- le point de rosée est contenu dans la balise "DewPoint"
- le taux d'humidité est contenu dans la balise "RelativeHumidity"
- la pression atmosphérique est contenue dans la balise "Pressure"

► Nous allons ensuite extraire les différentes données.

Saisissez le code suivant (toujours dans le code de clic du bouton "BTN_METEO") :

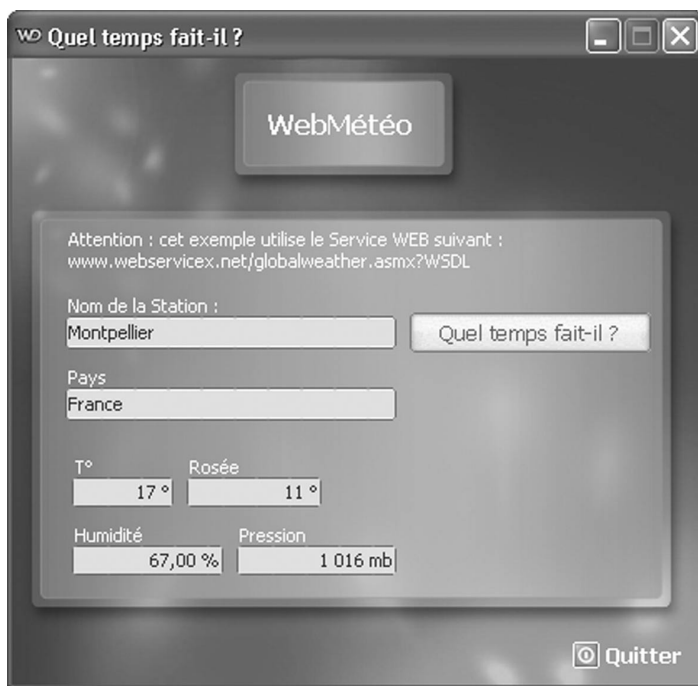
```
SAI_TEMPERATURE = XMLExtraitChaîne(...
                    ResultatXML:m_GetWeatherResult, "Temperature")
SAI_TEMPERATURE = Conversion(SAI_TEMPERATURE, "°F", "°C")
SAI_POINTROSEE = XMLExtraitChaîne(...
                    ResultatXML:m_GetWeatherResult, "DewPoint", 1, ...
                    XMLIgnoreLaCasse)
SAI_POINTROSEE = Conversion(SAI_POINTROSEE, "°F", "°C")
SAI_HUMIDITE = Val(XMLExtraitChaîne(...
                    ResultatXML:m_GetWeatherResult, "RelativeHumidity", ...
                    1, XMLIgnoreLaCasse))
SAI_PRESSION = XMLExtraitChaîne(...
                    ResultatXML:m_GetWeatherResult, "Pressure", 1, ...
                    XMLIgnoreLaCasse)
SAI_PRESSION = Conversion(SAI_PRESSION, "inHg", "mbar")

FIN
```

Et voilà, c'est terminé.

► Enregistrez puis testez la fenêtre.

Saisissez par exemple "Montpellier" dans le champ "Station" et "France" dans le champ "Pays", puis cliquez sur "Quel temps ?".



En quelques secondes, le service Web a été interrogé, le résultat exploité dans l'application et les données affichées. Et tout ça, en ne saisissant que très peu de code ! Une vraie petite grenouille électronique !

Pour plus de détails sur l'utilisation des services Web, consultez l'aide en ligne (mot-clé : "Services Web")

Créer des services Web XML

WinDev permet également de créer des services Web à partir de vos applications. Ces services Web devront être installés sur serveur Web supportant SOAP.

Pour générer un service Web XML :

1. Créez un projet WinDev. Ce projet contiendra uniquement des procédures globales (pas de fenêtres, pas d'états, ...). Chaque procédure globale pourra être exécutée par le serveur SOAP WinDev.
Saisissez si nécessaire le code d'initialisation et de fermeture du projet. Ces codes seront respectivement exécutés au chargement et au déchargement de la bibliothèque sur le serveur SOAP WinDev.
2. Sélectionnez l'option "Atelier .. Service Web XML .. Générer le service à partir de ce projet". L'assistant de génération de service Web XML se lance. Répondez aux questions posées par l'assistant.
3. Installez le service Web XML généré sur le serveur Web.

Nous ne détaillerons pas la création d'un service Web dans ce cours d'auto-formation. Pour plus de détails, consultez l'aide en ligne (mot-clé : "Service Web XML").

LEÇON 9.5. LES COMPOSANTS

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce qu'un composant ?
- Etape par étape, la création d'un composant.
- Diffusion d'un composant.

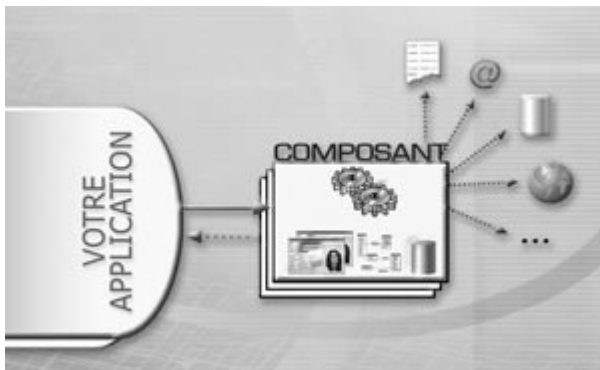


Durée estimée : 30mn

Présentation

Un composant WinDev est une brique applicative qui peut être réutilisée dans un nombre illimité de projets (et donc d'exécutables).

Un composant permet d'avoir un traitement identique avec un code source unique pour réaliser une opération donnée même si ce traitement doit être réalisé par plusieurs projets.



Les projets qui utilisent un composant ont accès sous l'éditeur WinDev au nom des objets, procédures ou méthodes rendues visibles par le concepteur du composant. Les projets ne peuvent pas voir ou modifier le code source. La confidentialité et la stabilité du code sont garanties.

La réalisation d'un composant est **très simple**.

Comment faut-il faire ? Faites comme d'habitude, créez vos fenêtres, procédures, classes. Puis lorsque vous aurez terminé, au lieu de choisir l'option de création de l'exécutable, prenez l'option de création d'un composant, c'est tout !

Un composant peut contenir du code, des fenêtres, une analyse, des fichiers de données, etc ... !



Notes

Pour partager du "code", il y a quatre méthodes principales avec WinDev :

1. Les collections de procédures
2. Les classes
3. Les superchamps (à travers l'utilisation d'un dictionnaire)
4. Les composants

Voici plusieurs cas dans lesquels les composants peuvent être intéressants.

Travail en équipe

Un développeur crée et maintient un composant qu'il met à la disposition des autres développeurs. Il ne risque pas d'y avoir des modifications faites "par erreur" !

Les projets volumineux

Les composants permettent d'avoir plusieurs projets distincts de tailles réduites, et d'avoir un projet central dans lequel les éléments des autres projets sont visibles. L'utilisabilité des composants est bien meilleure que les bibliothèques (fichiers WDL).

Les bases de données accédées par plusieurs projets

Lorsqu'une même base de données est accédée par plusieurs projets, il est fréquent d'avoir des incohérences dans la base de données dues à des codes sources modifiés ou trop vieux. En regroupant les opérations d'accès (en écriture au moins) à la base de données dans un composant, cela permet d'avoir un seul code source à vérifier/maintenir, et cela permet donc de limiter le risque d'incohérences dans la base de données.

Les traitements utilisés dans plusieurs projets

Il est fréquent d'avoir des traitements plus ou moins complexes utilisés dans plusieurs projets. Il est possible de réutiliser ces traitements par l'intermédiaire de "collections de procédures" ou de "classes". Dans ce cas il existe un risque de modifications non reportées entre les différents projets, qui rendent ensuite ces éléments incompatibles entre eux.

L'utilisation des composants permet d'éviter des modifications intempestives et permet de conserver des projets compatibles auxquels il est simple d'apporter une modification commune.

De plus avec l'utilisation possible du mot-clé PRIVE, vous pourrez assurer la confidentialité de votre code à tous les niveaux du composant. Lors de la réutilisation de votre composant dans un autre projet WinDev, l'édition du code ne sera pas permise mais la documentation concernant l'utilisation des fonctions (par exemple des procédures) sera affichée !

La possibilité de diffuser une fonctionnalité ou un ensemble de fonctionnalités

Les composants permettent de réaliser une fonctionnalité ou un ensemble de fonctionnalités. D'autres développeurs WinDev pourront intégrer ces fonctionnalités dans leurs propres projets. Les développeurs qui utilisent un composant peuvent voir les éléments du composant rendus visibles. Cependant il ne sera pas possible de voir ou de modifier le code source.

Vous pouvez diffuser gratuitement ou même commercialiser des composants !

Composant multiproduits

Un composant peut être prévu pour fonctionner dans :

- une application WinDev
- une application WebDev
- une application WinDev Mobile
- ou dans les trois types d'applications.

Dans ce dernier cas, WinDev permet :

- d'intégrer les éléments provenant des différents produits (WebDev et / ou WinDev Mobile) dans un même composant.
- d'indiquer pour chaque plateforme d'exécution le code WLangage correspondant (par exemple, l'affichage d'une fenêtre se fait par la fonction **Ouvre** en WinDev, et l'affichage d'une page se fait par la fonction **PageAffiche** en WebDev).

Etape par étape

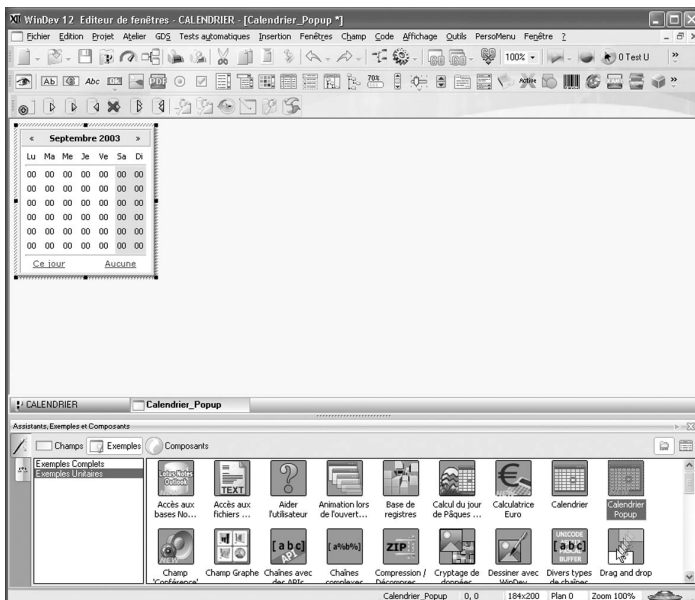
Etape 1 : Création d'un composant

Nous allons créer un composant proposant un calendrier. Lors de l'appel au composant, vous pourrez :

- Passer une date en paramètre pour se positionner dans le calendrier, à l'ouverture.
 - Récupérer la date sélectionnée dans le calendrier à la fermeture.
- ▶ Créez un nouveau projet nommé "CALENDRIER".
Ce projet n'a pas d'analyse.

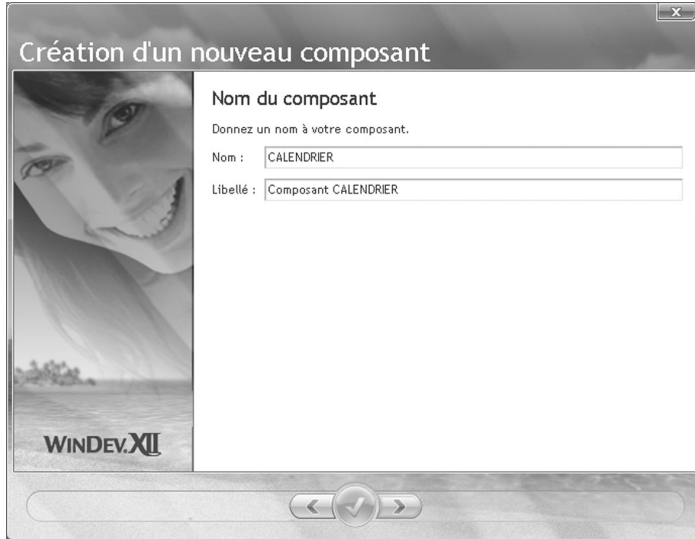
Le composant est constitué d'une fenêtre contenant un calendrier.

- ▶ Dans l'éditeur de WinDev, affichez le volet "Assistants, Exemples et Composants" si ce n'est pas déjà fait (option "Affichage .. Barres d'outils .. Assistants, Exemples et Composants").
- ▶ Affichez les exemples dans l'onglet "Assistants, Exemples et Composants" (bouton "Exemples"). Sélectionnez les exemples unitaires.
- ▶ Réalisez un "Drag and Drop" de l'exemple "Calendrier popup" vers l'éditeur de WinDev. La fenêtre apparaît sous l'éditeur de fenêtres.
- ▶ Sauvegardez la fenêtre.

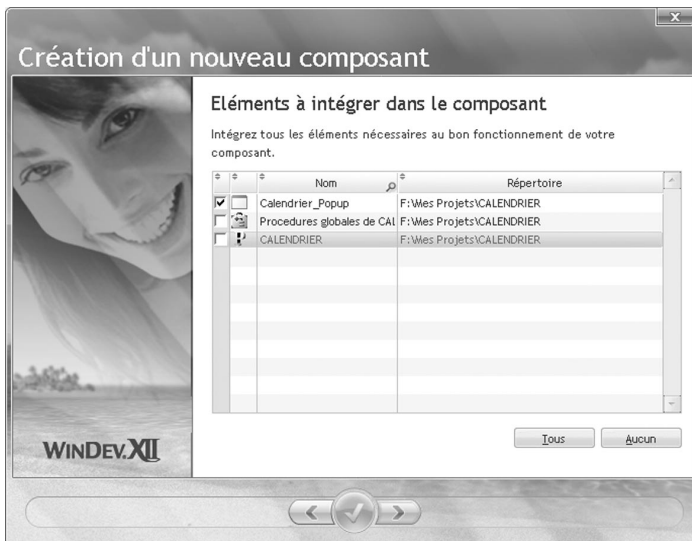


- ▶ Pour créer un composant, dans le menu de WinDev, sélectionnez l'option "Atelier .. Composant .. Définir un nouveau composant à partir de ce projet". WinDev propose de créer une nouvelle configuration de projet : cliquez sur "Non".
- ▶ L'assistant de création du composant démarre. Passez à l'écran suivant.

- ▶ Identifiez votre composant :
 1. Saisissez le nom "Calendrier".
 2. Saisissez comme libellé : "Composant calendrier".

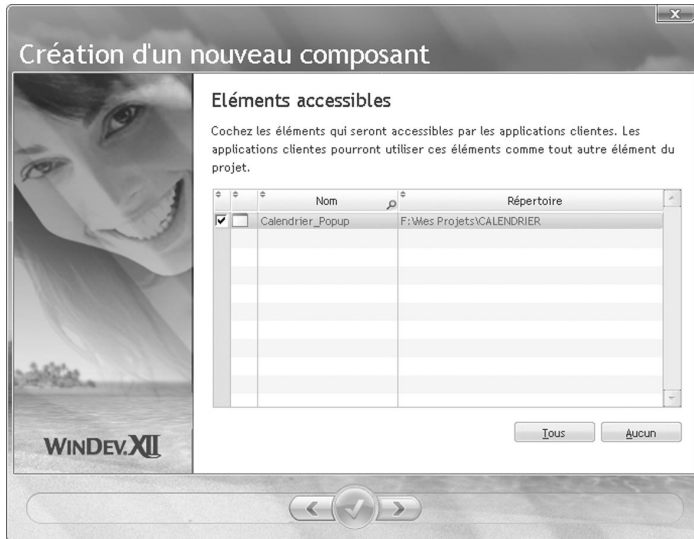


- ▶ Passez à l'écran suivant.
Sélectionnez les éléments qui constituent le composant. Cochez uniquement la fenêtre "Popup_Calendrier".



- ▶ Passez à l'écran suivant.

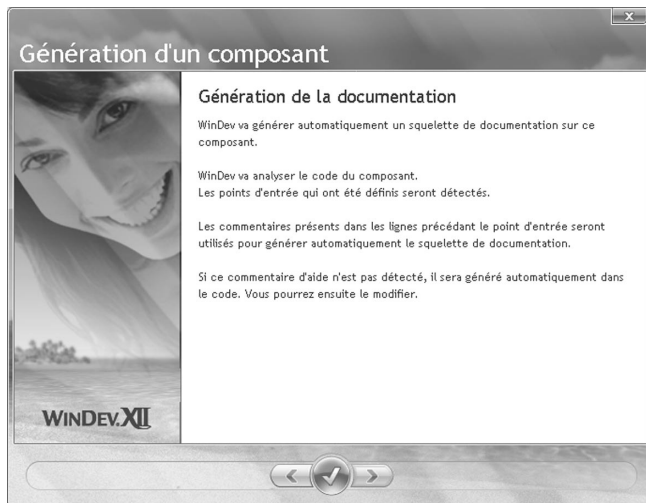
L'assistant vous demande de sélectionner les éléments du composant qui seront accessibles depuis l'application cliente. Dans notre exemple, seule la fenêtre "Calendrier_Popup" sera utilisée :



- ▶ Passez à l'écran suivant. WinDev propose de générer le composant.
- ▶ Dans l'assistant qui apparaît, choisissez les langues du composant. L'assistant propose ensuite de gérer les différentes versions du composant. Dans notre exemple, le composant vient d'être créé. Conservez les options par défaut. Passez à l'écran suivant.
- ▶ Dans cette fenêtre, saisissez les différentes informations concernant le composant :
 - Propriétaire,
 - Libellé, Copyright, ...
- ▶ Passez à l'écran suivant.
- ▶ Vous pouvez associer une image à votre composant. Les utilisateurs du composant pourront l'identifier rapidement dans le volet "Catalogue d'exemples".
- ▶ Passez à l'écran suivant.

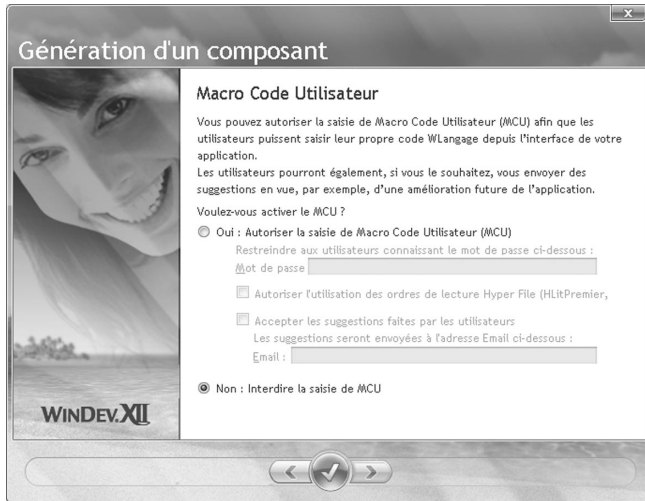


- ▶ Spécifiez la présentation générale du composant.
- ▶ Passez à l'écran suivant. L'assistant va maintenant effectuer la génération automatique de la documentation sur le composant. Cette documentation peut être basée sur les commentaires que vous aurez insérés dans votre code source.



- ▶ Passez à l'écran suivant.. Vous avez la possibilité de modifier l'aide générée. Nous n'allons pas y toucher. Passez à l'écran suivant.
- ▶ Vous avez la possibilité de créer un fichier d'aide associé à votre composant (format CHM). Ce fichier d'aide sera livré avec le composant. Les clients pourront y accéder en faisant F1 à partir du code du composant. Nous n'allons pas générer de fichier d'aide. Décochez si nécessaire l'option. Passez à l'écran suivant.

- ▶ Vous avez la possibilité d'autoriser les utilisateurs finals du composant à saisir leur propre code WLangage pour personnaliser vos traitements. Les suggestions peuvent directement vous être envoyées. Dans cet exemple, nous n'allons pas autoriser la saisie de "Macro Code Utilisateur". Passez à l'écran suivant.



- ▶ Nous n'allons pas enregistrer le composant dans le centre de ré-utilisabilité. Il ne reste plus alors qu'à donner un nom physique à notre composant.

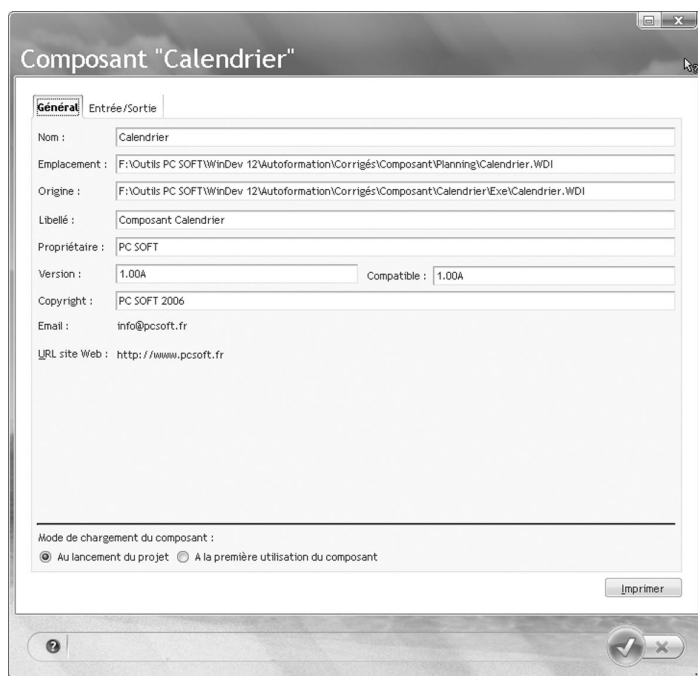


- ▶ Cliquez sur "Terminer". Voilà, votre premier composant est maintenant créé, il est prêt à l'emploi !
Le composant apparaît alors dans le volet "Assistants, Exemples et Composants" (bouton "Composants", rubrique "Composants personnels").

Etape 2 : Utilisation du composant

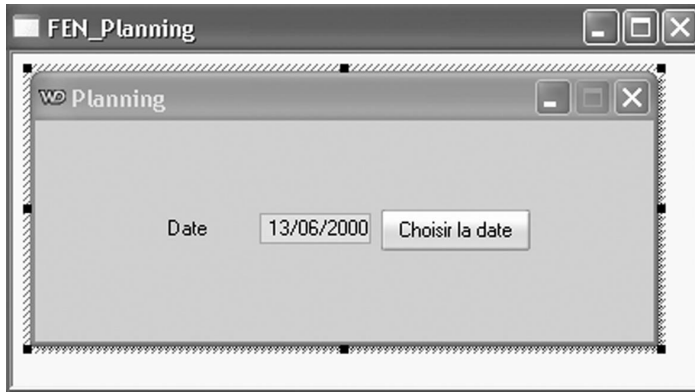
Une fois créé, votre composant est alors utilisable dans n'importe quel autre projet WinDev. Nous allons maintenant voir comment réutiliser ce composant.

- ▶ Créez un nouveau projet. Ce projet s'appelle "Planning" et n'a pas d'analyse.
- ▶ Dans l'éditeur de projet, sélectionnez l'option "Atelier .. Composant .. Importer un composant dans le projet .. À partir d'un fichier".
Remarque : Si vous avez enregistré le composant dans le centre de ré-utilisabilité, vous pouvez l'importer directement depuis le centre de ré-utilisabilité.
- ▶ Dans le répertoire du projet CALENDRIER, sélectionnez le sous-répertoire EXE, puis le fichier "CALENDRIER.WDI".
- ▶ Après avoir cliqué sur "Ouvrir", la description de notre composant s'est affichée :



- ▶ Validez.
Vous allez maintenant créer une nouvelle fenêtre vierge dans laquelle vous allez ajouter :
 - un champ de saisie de type "Date"
 - un bouton.
- ▶ Cliquez sur "Fichier .. Nouveau .. Fenêtre". Sélectionnez une fenêtre vierge.
- ▶ Nommez votre fenêtre "FEN_Planning" et donnez-lui le titre "Planning".
- ▶ Créez un champ de saisie que vous nommerez "SAI_LaDate". Dans la fenêtre de description de ce champ, indiquez qu'il s'agit d'un champ "Date".

- ▶ Créez ensuite un bouton que vous nommerez "BTN_ChoisirDate" avec comme libellé "Choisir la date".



- ▶ Editez maintenant le code de clic du bouton "BTN_ChoisirDate" et saisissez le code suivant :

```
SAI_LaDate = Ouvre (PopCalendrier)
```

- ▶ Sauvegardez votre fenêtre, puis testez en cliquant sur le bouton "GO".



Et c'est terminé ! Simple n'est-ce pas ?

Vous savez désormais créer un composant et le réutiliser dans vos applications. Mais vous pourriez également décider de gérer des procédures d'installation de vos composants, pour, par exemple, les redistribuer séparément de vos applications.

Distribuer un composant

Vous avez deux choix possibles pour distribuer un composant :

1. Fournir les fichiers nécessaires "manuellement", c'est une distribution "simple".
2. Créer une distribution plus "professionnelle", grâce à l'éditeur d'installation de WinDev, WDIInst.

Distribution simple

Vous devez dans ce cas fournir tous les fichiers nécessaires au bon fonctionnement de votre composant. Ces fichiers sont créés à la génération du composant (fichiers WDI, WDK, WDO, images, autres documents, ...). Ces fichiers seront copiés manuellement de leur répertoire d'origine vers le répertoire de destination. Les projets WinDev qui utiliseront ce composant, iront chercher les fichiers dépendants dans ce répertoire de destination.

Liste des fichiers à fournir pour une distribution simple :

- les fichiers générés automatiquement par WinDev (WDK, WDI, ...).
- les fichiers de dépendances.
- si le composant utilise des fichiers de dépendances, il faudra fournir le fichier WDO. Ce fichier contient les références aux différents fichiers externes utilisés dans le composant.

Distribution professionnelle

La diffusion des composants à l'aide d'une procédure d'installation consiste à fournir aux utilisateurs du composant WinDev, un programme d'installation. Ce programme installe tous les fichiers nécessaires à l'utilisation du composant dans le répertoire spécifié par l'utilisateur.

Ce mode d'installation permet de gérer automatiquement :

- le fichier WDO et l'installation des fichiers supplémentaires utilisés par le composant.
- l'installation automatique d'outils nécessaires (MDAC, driver ODBC pour Hyper File, ...).
- la mise à jour automatique des fichiers de données utilisés par le composant (si nécessaire).
- la désinstallation du composant.

- ▶ Revenez sur le projet "Calendrier". Pour créer l'installation, sélectionnez l'option "Atelier .. Composant .. Créer la procédure d'installation d'un composant". L'assistant de création d'installation du composant se lance.

Nous ne détaillerons pas les différents modes d'installation d'un composant. Suivez les instructions de l'assistant. Si vous souhaitez obtenir plus de détails concernant l'installation de composants, consultez l'aide en ligne (mot-clé : "Composant WinDev, Distribuer un composant").

Quelques composants livrés en standard avec WinDev 12 :

- Affiche image
- Agent
- Ballon Tip
- CodePostaux France
- Envoi Fax
- Gestion Login
- JourFérié
- Moteur de recherche
- Paiement sécurisé
- Sélecteur
- Sélecteur Couleurs
- SOAPGMT
- Zone mémoire visuelle
- ComposantHFInternet

LEÇON 9.6. MULTI-CONFIGURATIONS

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Utiliser un même projet pour différentes configurations



Durée estimée : 20mn

Présentation

Vous avez créé une application, et vous voulez créer un composant à partir de certaines fenêtres du projet ? Votre projet contient des procédures qui pourraient être utilisées dans un service Web ? Une partie de votre application peut être utilisée en Java ?

Comment ne pas multiplier les projets ? Un seul projet et plusieurs destinations possibles, voilà le défi résolu grâce aux configurations de projet WinDev.

Les configurations de projet permettent de créer à partir d'un même projet plusieurs "cibles" différentes. À tout moment, vous pouvez choisir la configuration sur laquelle vous travaillez, et générer en une seule opération tous les éléments pour toutes les configurations d'un projet.

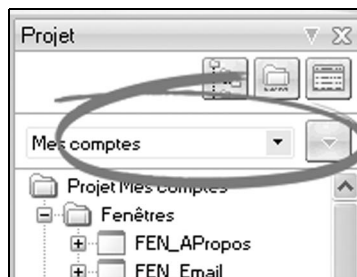
Un exemple pratique ? Nous allons tester les configurations de projet sur notre application complète.

Création d'une configuration de projet

Nous allons créer une configuration de projet afin de créer un composant à partir du projet "Mes Comptes".

Pour créer la configuration de projet :

1. Ouvrez si nécessaire le projet "Mes Comptes".
1. Dans le volet "Projet", sélectionnez la flèche vers le bas :



2. Sélectionnez "Nouveau". L'assistant de création d'une configuration de projet se lance.
3. Indiquez le nom, la description, et le type de génération (Composant test, test, composant pour notre exemple).



4. Indiquez pour quels produits le composant pourra être utilisé (WinDev).
5. Indiquez les éléments à intégrer dans la configuration. Pour notre exemple, intégrez deux fenêtres.
6. Validez la création de la configuration.
7. L'assistant de génération de l'objet de la configuration (dans notre cas, le composant) se lance automatiquement.
8. Poursuivez l'assistant pour créer le composant.
9. A la fin de l'assistant, la configuration "Composants tests" est automatiquement sélectionnée dans le volet "Projet". Sélectionnez la configuration "Mes comptes".

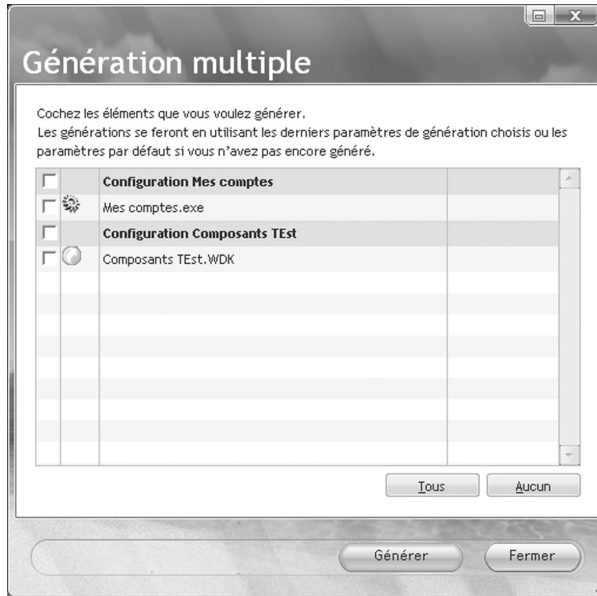
Gestion des configurations de projet

L'option "Projet .. Configurations de projet .. Gestion des configurations de projet" permet de définir toutes les options des configurations de projet. Il est possible de :

- Générer une configuration
- Supprimer
- Paramétrer le mode test
- Obtenir la description de l'élément principal de la configuration
- paramétrer des actions avant et après génération (comme par exemple la copie de l'élément dans un répertoire spécifique).
- Créer une nouvelle configuration

Génération multiple

Nous avons vu comment générer une configuration, mais bien souvent, après une modification, il est nécessaire de générer les éléments correspondants à toutes les configurations définies sur le projet. Pour cela, il suffit d'utiliser l'option "Atelier .. Génération multiple".



Il suffit de sélectionner les éléments à générer. Cette génération sera effectuée en une seule opération.

LEÇON 9.7. LE DÉBOGUEUR

Ce que vous allez apprendre dans cette leçon ...

- Présentation
- Déboguer un élément du projet
- Déboguer un projet
- Options avancées



Durée estimée : 20mn

Présentation

Lors de la réalisation de l'application complète (projet MesComptes de la partie 2), nous avons effectué plusieurs tests : tests de fenêtres, tests de requêtes, tests d'états, ...

Une erreur apparaît et vous n'arrivez pas à trouver la cause ? WinDev permet d'aller plus loin dans les tests, en permettant de suivre pas à pas l'exécution d'un programme : vous visualisez et vous validez toutes les lignes de code qui sont exécutées grâce au débogueur.

Le débogueur vous permet pendant l'exécution de votre application, de connaître les valeurs des différentes variables utilisées, d'exécuter certains traitements, de ré-exécuter des lignes de code. Il est même possible de modifier des lignes de code tout en débogueant.

Le débogueur peut être utilisé en développement (pour trouver la cause d'un problème pendant la phase de développement), ou en exécution : il est possible de déboguer un exécutable lancé sur le poste ou sur un autre poste, un composant, ...

Ces fonctionnalités vous intéressent ? Passons aux choses sérieuses et testons directement ces fonctionnalités.

Déboguer une fenêtre

Lancer le débogueur

Nous avons vu que pour lancer le test d'une fenêtre, il suffit d'utiliser l'icône GO (le raccourci [F9] ou l'option "Code .. Tester la fenêtre").

Pour déboguer une fenêtre, plusieurs méthodes sont à votre disposition.

1ère méthode : Vous voulez déclencher le débogueur pendant le test de votre application, à partir d'une action spécifique : il suffit d'utiliser la combinaison de touches [CTRL] + [PAUSE]

2ème méthode : Vous connaissez le code qui cause l'erreur dans votre fenêtre, vous voulez lancer le débogueur à partir d'une ligne de code spécifique : vous pouvez poser un point d'arrêt ou utiliser l'instruction STOP.

Par exemple, nous allons lancer le débogueur sur une des fenêtres de notre application développée dans la partie 2

1. Ouvrez le projet "MesComptes". Si vous n'avez pas réalisé l'application, le corrigé est disponible dans le sous-répertoire "\\Autoformation\Corrigés\MesComptes".
2. Ouvrez la fenêtre "FEN_PersonneX".
3. Affichez le code du bouton de recherche à l'identique (BTN_Identique).
4. Cliquez avec la souris dans la colonne jaune précédant la première ligne de code. Un point rouge apparaît : c'est un point d'arrêt.

```


● HLitRecherchePremier(PERSONNE, IDPERSONNE, COMBO_PERSONNE)
  □ SI HTrouve(PERSONNE) ALORS
    |   FichierVersEcran()
    |
    | FIN
  
```

En test, le débogueur sera activé dès l'exécution de la (ou les) ligne(s) contenant un point d'arrêt.



Notes

Le mot-clé STOP a le même effet que le point d'arrêt. Comme le point d'arrêt, il n'est pas pris en compte dans l'exécutable.

5. Le point d'arrêt étant posé, lançons le test de notre fenêtre (icone .
6. Sélectionnez un élément dans la combo.
7. Cliquez sur le bouton "Recherche identique".
8. L'éditeur de code apparaît en mode débogage.

Nous allons maintenant détailler les informations affichées sous le débogueur.

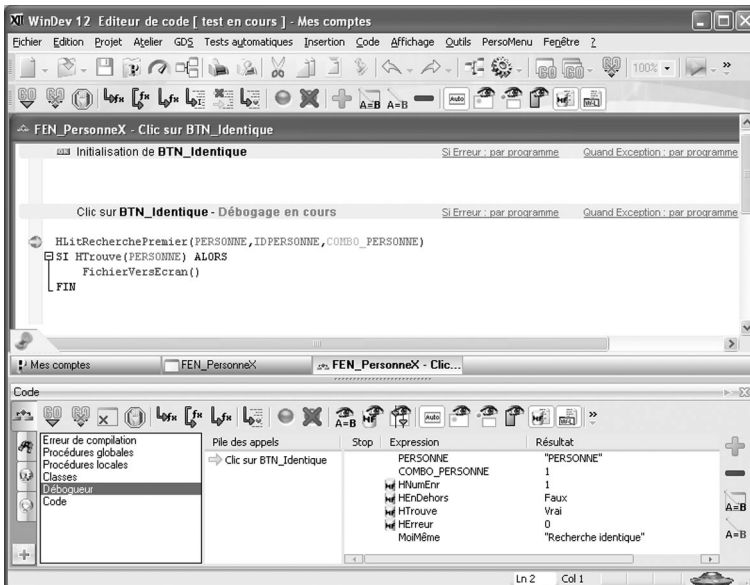
Visualiser les informations sous le débogueur

En mode débogage, l'éditeur de code est un petit peu modifié. Une barre d'outils spécifique au débogage est affichée, et le volet Code se place automatiquement en mode "Débogueur".



L'éditeur de code va permettre de suivre ligne par ligne l'exécution du code.

Le volet "Code" en mode "Débogueur" va permettre de suivre les valeurs prises par les différentes variables manipulées par le code.



Suivons les premières lignes de débogage :

1. Dans l'éditeur de code, la flèche jaune indique la ligne qui va être exécutée.
2. Utilisez la touche [F7]. La ligne est exécutée. La flèche jaune passe à la ligne suivante.

3. Positionnez le curseur de souris sur la ligne "FIN".
4. Utilisez la touche [F6]. Toutes les lignes sont exécutées jusqu'au curseur.

Nous allons maintenant voir les valeurs des variables utilisées par la procédure :

- Les variables manipulées dans le code sont automatiquement visualisées dans le volet "Code", option "Débogueur". Leur valeur est automatiquement affichée.
- Pour connaître la valeur d'une variable présente dans le code, il suffit de la survoler avec le curseur de la souris : la valeur apparaît dans une bulle d'aide.
- Pour ajouter une expression dans le volet "Code", option "Débogueur", il suffit de sélectionner l'expression dans l'éditeur de code, et de sélectionner l'option "Ajouter l'expression dans le débogueur" du menu contextuel de l'éditeur de code.

Lorsque l'utilisation du débogueur n'est plus nécessaire, vous pouvez :

- utiliser la touche [F5] pour continuer l'application sans utiliser le débogueur.
- utiliser la touche [SHIFT + F5] pour arrêter l'application. Dans ce cas, l'éditeur de code sera ré-affiché en mode normal sur le code en cours d'exécution.

Nous avons vu ici les points les plus importants du débogueur. De nombreuses options sont également disponibles, comme les expressions auto-stop, la modification de code pendant le débogage, ... Nous vous conseillons de consulter l'aide en ligne pour plus de détails (mot-clé : "Débogueur").

Fenêtre de trace du débogueur

Pour déboguer un projet, il est également possible d'utiliser la fonction **WLanguage Trace**. Cette fonction permet d'afficher dans une "fenêtre de trace" toutes les informations que vous voulez. Par exemple, il est possible d'afficher la valeur d'une variable, un message pour savoir par où passe l'application, ...

Lors du GO (du projet, de la fenêtre, ...), lorsque la fonction **Trace** est rencontrée, la fenêtre de trace s'affiche automatiquement (cette fenêtre ne s'affichera pas en clientèle). Lorsque le test se termine, cette fenêtre disparaît. Cependant, vous pouvez retrouver les informations affichées dans la fenêtre de trace grâce au volet "Trace du débogueur".

Cette fenêtre affiche toutes les informations affichées pendant le test avec la fonction "Trace". Un clic sur une des lignes de la fenêtre de trace permet d'accéder automatiquement à la ligne de code correspondante.



Notes

Une erreur survient pendant le test de votre application ? Vous n'avez pas le temps de noter les informations affichées ? Pas de problème, la fenêtre de trace a mémorisé ces informations. Un double-clic sur le message d'erreur permet d'obtenir toutes les informations détaillées.

Déboguer un projet

Nous venons de voir comment déboguer une fenêtre, mais il est également possible de déboguer le projet dans son ensemble. Dans ce cas, les paramètres spécifiés dans l'option "Projet .. Mode Test .. Paramétrage du mode test" sont pris automatiquement en compte.

Options avancées du débogueur

Le débogueur de WinDev est un outil puissant, permettant de déboguer de nombreuses applications, dans les cas les plus divers. Il est ainsi possible de déboguer :

- un composant, par exemple depuis un projet qui utilise le composant.
- un exécutable (sur le poste en cours ou sur une machine distante)
- un exécutable déjà lancé (sur le poste en cours ou sur une machine distante)
- ...

LEÇON 9.8. L'ANALYSEUR DE PERFORMANCES

Ce que vous allez apprendre dans cette leçon ...

- Présentation de l'analyseur de performances.
- Utilisation de l'analyseur de performances.



Durée estimée : 20mn

Présentation

L'analyseur de performances est un outil permettant de vérifier et d'optimiser le temps d'exécution de votre application.

Il est conseillé d'utiliser l'analyseur de performances pour optimiser votre application (avant sa diffusion par exemple).

Son principe est simple :

Vous testez votre application. Pendant ce test, l'analyseur de performances répertorie toutes les actions et traitements exécutés.

À la fin du test, l'analyseur de performances vous présente :

- les 10 traitements qui ont pris le plus de temps
- toutes les actions effectuées dans l'application testée, triées par durée (de l'action la plus longue à l'action la moins longue).

Il est possible de sélectionner en particulier un traitement afin d'analyser les causes d'un temps de traitement trop important.

Utiliser l'analyseur de performances

L'analyseur de performances peut être lancé :

- **directement depuis l'éditeur de WinDev** (option "Projet .. Performances et améliorations .. Analyser les performances").

Dans ce cas, le projet est alors automatiquement exécuté en mode test. Vous pouvez manipuler votre application et lancer les traitements de votre choix.

Pour revenir sous l'éditeur WinDev, il suffit de quitter votre application.

L'analyseur de performances affiche alors le résultat de l'analyse. Ce résultat est enregistré sous forme d'un fichier WPF.

- **depuis un de vos traitements en WLangage**, grâce aux fonctions suivantes :

AnalyseurDébut	Lance la "collecte d'informations" pour l'analyseur de performances.
AnalyseurFin	Arrête la "collecte d'informations" pour l'analyseur de performances.

Dans ce cas, seul le code présent entre les fonctions **AnalyseurDébut** et **AnalyseurFin** est analysé. Le fichier WPF correspondant peut être ouvert dans l'analyseur de performances (option "Projet .. Performances et améliorations .. Rapport de performances .. Ouvrir un rapport de performances").

Par défaut, l'analyseur de performances enregistre les statistiques réalisées sur le code de l'application dans un fichier <Nom du Projet>.WPF.

Pour changer ce nom, cliquez sur le bouton "Enregistrer sous ..." dans l'analyseur de performances.

Pour ouvrir un fichier de statistiques spécifique (par exemple un fichier créé par programmation) :

1. Sélectionnez l'option "Projet .. Performances et améliorations .. Rapport de performances .. Ouvrir un rapport de performances".
2. Indiquez le chemin et le nom du fichier de statistiques.

La liste des derniers fichiers de statistiques ouverts est présente dans le menu "Projet .. Performances et améliorations .. Rapport de performances".

Optimiser un traitement avec l'analyseur de performances

Lire le résultat de l'analyseur de performances

L'analyseur de performances présente le résultat de l'analyse dans deux onglets :

- l'onglet "Synthèse" présente les 20 traitements qui ont pris le plus de temps.
- l'onglet "Détail" présente tous les traitements lancés lors du test de l'application (classés du plus long au plus rapide).

Pour chaque traitement, les informations suivantes sont affichées :

Function	Temps total	Nb appels	Temps 1 Appel	% code	Parent
Exécution complète	5 s 421 ms	1	5 s 421 ms	0,1 %	
Total Fenêtre "FEN_MenuChamp"	5 s 417 ms	1	5 s 417 ms	0,5 %	
sélection du menu de _Menu.Champ.OPT_Graphe	3 s 862 ms	1	3 s 862 ms	0,0 %	FEN_MenuChamp
Total Fenêtre "FEN_ChampGrappe"	3 s 862 ms	1	3 s 862 ms	5,0 %	
Ouvre("FEN_ChampGrappe")	3 s 518 ms	1	3 s 518 ms	0,0 %	FEN_MenuChamp
initialisation de DONNEE_1	3 s 16 ms	1	3 s 16 ms	0,0 %	FEN_ChampGrappe
grDesigne()	3 s 13 ms	1	3 s 13 ms		FEN_ChampGrappe
sélection du menu de _Menu.Champ.OPT_Libellé	803 ms	1	803 ms	0,0 %	FEN_MenuChamp
Total Fenêtre "FEN_ChampLibellé"	803 ms	1	803 ms	6,3 %	
Ouvre("FEN_ChampLibellé")	650 ms	1	650 ms	0,0 %	FEN_MenuChamp
sélection du menu de _Menu.Champ.OPT_Ascenseur	454 ms	1	454 ms	0,0 %	FEN_MenuChamp
Total Fenêtre "FEN_ChampAscenseur"	454 ms	1	454 ms	0,0 %	

Les éléments affichés sont les suivants :

1. Fonction : Fonction, traitement ou procédure exécutée.
2. Temps Total : Temps d'exécution de la fonction.
3. Nb appels : Nombre d'appels effectués à la fonction (procédure ou traitement)
4. Temps 1 appel : Temps d'exécution d'un appel à la fonction (procédure ou traitement)
5. % Code : Pourcentage du temps passé dans le traitement de la fonction ou de la procédure (code du développeur pouvant être optimisé)
6. Parent : Élément qui contient le traitement.

Choisir un traitement à optimiser

Le choix du traitement à optimiser se fait en fonction de plusieurs critères :

- le temps d'exécution du traitement. Les traitements les plus longs doivent bien entendu être optimisés.
- le pourcentage de temps passé dans le traitement de la fonction ou procédure. Plus ce pourcentage est important, plus le code peut contenir des traitements pouvant être optimisés.

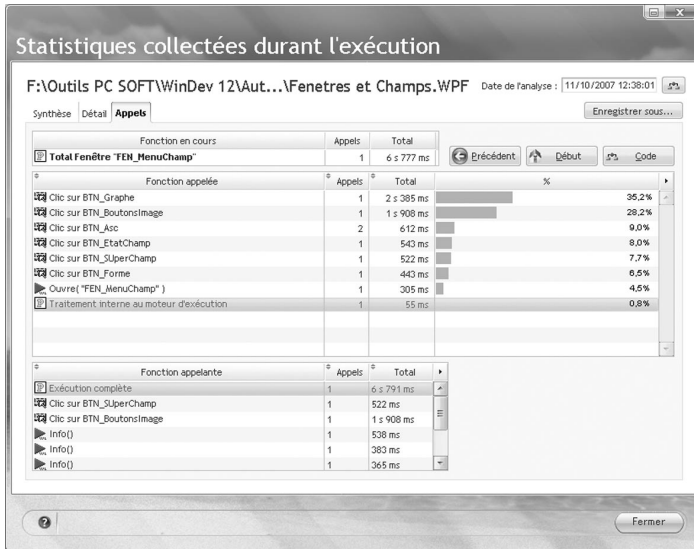
Optimiser un traitement

Lorsque le traitement à optimiser est choisi, l'analyseur de performances vous permet de connaître le détail des opérations effectuées.

Pour afficher le détail des opérations effectuées dans un traitement :

1. Sélectionnez la fonction à optimiser dans l'onglet "Détail".
2. Affichez le détail de cette fonction :
 - soit en double-cliquant sur la fonction.
 - soit en cliquant sur le bouton "Appels".

L'onglet suivant apparaît :



Cet onglet est divisé en trois parties :

- La fonction sélectionnée (avec le nombre d'appels à cette fonction et le temps total de traitement).
- La liste des fonctions appelées par la fonction sélectionnée.
- La liste des fonctions appelant le traitement sélectionné.

La liste des fonctions appelées permet d'affiner la recherche du traitement à optimiser. En effet, il est possible de visualiser les différentes fonctions appelées et le temps de traitement de chaque fonction.

Tous les traitements contenant du code WLangage (nommé "Code du développeur") peuvent être optimisés.

Un double-clic sur une des fonctions de cette liste (par exemple : "Clic sur Valider") permet de visualiser le détail des traitements appelés par cette fonction.

Remarques :

- Le libellé "Traitement interne au moteur d'exécution" correspond au temps d'exécution de la fonction ou de la procédure (cas d'une fonction WLangage par exemple). Ce temps est incompressible et ne peut pas être optimisé.
 - Le libellé "Code du développeur" correspond au temps d'exécution du code même de la fonction ou de la procédure (hors appels de sous-fonctions). Ce temps est compressible et peut être optimisé.
 - Pour visualiser rapidement le code du traitement en cours, il suffit de cliquer sur le bouton "Code". L'analyseur reste ouvert et vous pouvez modifier le code en cours.
 - Le bouton "Précédent" permet de revenir à la fonction immédiatement appelante.
- Le bouton "Début" permet de revenir au traitement du début de l'application.

LEÇON 9.9. IMPORT / EXPORT

Ce que vous allez apprendre dans cette leçon ...

- Importer des éléments d'un projet dans un autre.
- Exporter les éléments de votre projet.
- Imports spécifiques (Projet WebDev, Access, Visual Basic, fenêtres non WinDev, ...).



Durée estimée : 10mn

Importer des éléments

Vous avez la possibilité d'importer des éléments WinDev existants dans votre projet. Vous pouvez importer tous les types d'éléments WinDev :

- des fenêtres
- des états
- des classes
- des composants
- des procédures
- ...

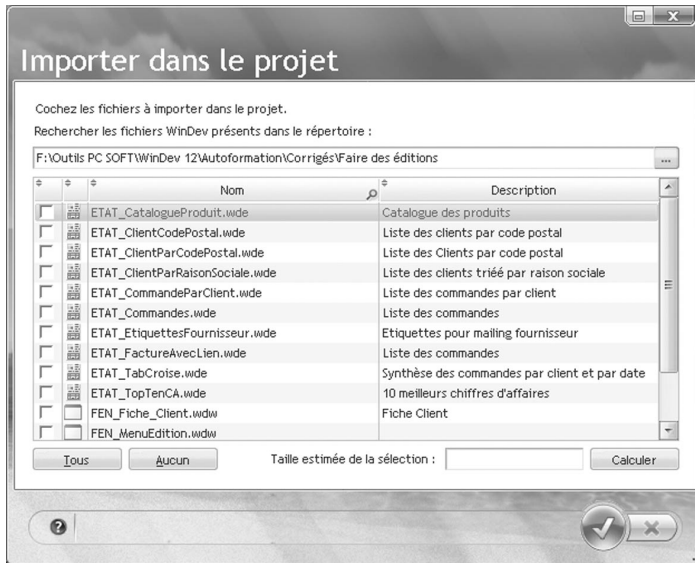
► Pour importer des éléments existants dans le projet en cours :

1. Sélectionnez l'option "Fichier .. Importer .. Des éléments WinDev et leurs dépendances...".



2. Cliquez sur "...", et sélectionnez le répertoire où se trouvent les éléments à importer (le répertoire doit comporter des éléments WinDev).

3. Validez. WinDev liste alors les différents éléments que vous pouvez importer présents dans le répertoire spécifié (les sous-répertoires ne sont pas pris en compte).



4. Sélectionnez les éléments à importer, puis validez par "OK".

Les éléments (et tous les fichiers utilisés par ces éléments : images, ...) font désormais partie du projet.



Notes

Le bouton "Calculer" (présent dans la fenêtre d'importation) permet de calculer la taille des éléments sélectionnés avec toutes leurs dépendances.

Exporter des éléments

Vous pouvez également exporter des éléments de votre projet vers un autre répertoire par exemple. Ces éléments pourront alors être réutilisés dans d'autres projets.



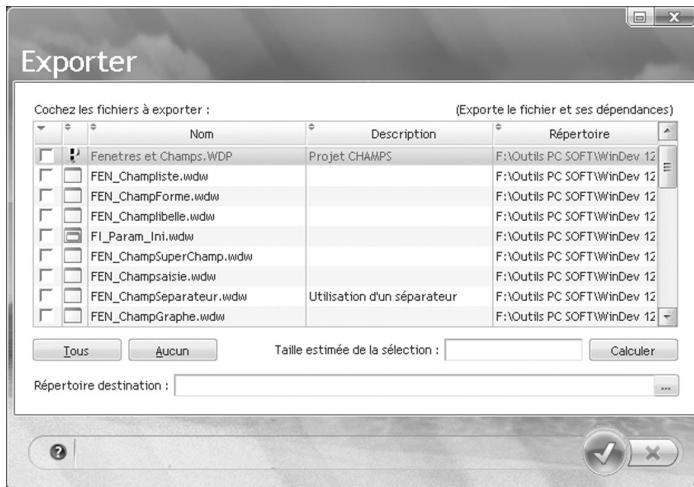
Attention !

L'exportation d'éléments est plus qu'une simple copie d'éléments. En effet, c'est le seul moyen sûr pour transmettre un élément d'un projet avec **toutes** ses dépendances (images, icônes ...). Une solution pratique pour transmettre des fenêtres avec leurs dépendances par e-mail, par exemple.

► Pour exporter des éléments de votre projet :

1. Cliquez sur "Fichier .. Exporter .. Vers un répertoire..."

Dans la fenêtre qui s'affiche, sélectionnez les éléments de votre projet que vous souhaitez exporter.



2. Indiquez le nom du répertoire de destination (ou sélectionnez-le avec le bouton "...").

3. Validez.

Les éléments sont alors exportés vers le répertoire indiqué. Ces éléments sont toujours disponibles dans votre projet.



Notes

Le bouton "Calculer" (présent dans la fenêtre d'importation) permet de calculer la taille des éléments sélectionnés avec toutes leurs dépendances.

Importations spécifiques

Importation d'un projet WebDev

De plus en plus de sites WebDev doivent être transformés en applications WinDev. Désormais, cette fonctionnalité est automatique.

WinDev permet d'importer une page ou un projet WebDev entier dans une application WinDev.

Lors de cette importation :

- Les pages sont transformées en fenêtres.
- Les codes "Serveur" et "Navigateur" sont regroupés.
- L'import d'éléments sans équivalent en WinDev (code Javascript par exemple) est effectué sous forme de commentaires ou d'éléments séparés.

Pour importer un projet WebDev dans un projet WinDev :

1. Sélectionnez l'option "Fichier .. Importer .. Un projet WebDev". L'assistant d'import de projet WebDev se lance.
2. Sélectionnez le projet WebDev à importer. Ce projet ne sera pas modifié et un nouveau projet WinDev sera créé.
3. Donnez le nom et l'emplacement du projet WinDev à créer.
4. Si une analyse est liée au projet, indiquez son nom, son emplacement, et si elle doit être utilisée par le projet WinDev.
5. Indiquez les éléments à partager entre le projet WinDev et le projet WebDev (éléments communs du type Etats, Classes, ...). Si les éléments sont partagés, ils ne seront pas copiés dans le projet WinDev.
6. Cliquez sur "Terminer". Le projet WebDev est converti en projet WinDev.

Pour importer des éléments WebDev dans un projet WinDev :

1. Ouvrez le projet WinDev dans lequel l'élément doit être importé.
2. Sélectionnez l'option "Fichier .. Importer .. Des éléments WebDev". La liste des éléments pouvant être importés s'affiche.
3. Sélectionnez les éléments à importer. Passez à l'écran suivant.
4. Indiquez les éléments à partager entre les projets WinDev et WebDev. Ces éléments ne seront pas copiés dans le projet WinDev.
5. Validez (bouton "Terminer"). Les éléments spécifiés sont automatiquement importés dans le projet WinDev en cours.

Importation d'une fenêtre

Une fenêtre d'une application non WinDev vous intéresse ? Vous souhaitez récupérer son interface ? Rien de plus simple.

WinDev permet d'importer une fenêtre ouverte dans l'environnement en cours.

La fenêtre est importée dans le projet en cours. Chaque élément de la fenêtre est automatiquement transformé en élément WinDev (libellé, champ de saisie, bouton, ...).

Remarque : Tous les éléments de la fenêtre sont importés mais il est souvent nécessaire de modifier la fenêtre pour obtenir l'interface voulue.

Pour importer une fenêtre dans un projet WinDev :

1. Ouvrez le projet WinDev dans lequel la fenêtre doit être importée.
 2. Sélectionnez l'option "Fichier .. Nouveau .. Fenêtre".
 3. Affichez l'onglet "Import". Les fenêtres actuellement affichées sur le poste en cours sont automatiquement listées. Ces fenêtres sont identifiées par leur titre (affiché dans la barre de titre).
- Remarque :** L'option "Générer des images pour les champs non reconnus" permet de rendre l'interface finale plus proche de l'interface de la fenêtre à importer.
4. Validez (bouton "OK"). L'aperçu de la fenêtre à importer s'affiche.
 5. Modifiez si nécessaire le type de champ créé lors de l'importation de la fenêtre (bouton "Détails").
 6. Validez (bouton "OK"). La fenêtre est automatiquement intégrée au projet WinDev en cours.

Limites : Ne sont pas importés :

- le code des éléments.
- les champs différents des champs suivants :
 - champ standard de Windows.
 - champs Delphi.
 - champs Visual Basic.
- les images et les images de fond (sauf si l'option "Générer des images pour les champs non reconnus" est cochée).
- la couleur des éléments.

Importation Access

WinDev offre la possibilité d'importer des éléments Access : Analyse, Formulaire, Requête ou projet complet.

Pour importer des éléments d'une application Access :

1. Sélectionnez l'option "Fichier .. Importer .. Un projet ou des éléments Access".
2. Sélectionnez le fichier ".mdb" correspondant au projet Access.
3. Sélectionnez les éléments à importer. Indiquez si nécessaire si la base de données doit être migrée au format Hyper File ou conservée au format Access.
4. Validez.

Attention : L'importation d'éléments d'une application Access peut être relativement longue.

Importation Visual Basic

WinDev offre la possibilité d'importer des projets Visual Basic et des formulaires Visual Basic.

Pour importer des éléments d'une application Visual Basic :

1. Sélectionnez l'option "Fichier .. Importer .. Un projet ou des éléments Visual Basic".
2. Sélectionnez les fichiers correspondants aux éléments à importer.
3. Validez.

Attention : L'importation d'éléments d'une application Visual Basic peut être relativement longue.

LEÇON 9.10. INSTALLATION D'UNE APPLI-CATION

Ce que vous allez apprendre dans cette leçon ...

- Editeur d'installation
- Update réseau, update Web
- Questions / Réponses



Durée estimée : 20mn

Présentation

Nous avons effectué précédemment le déploiement d'une application simple.

Cette leçon revient sur l'installation d'une application en présentant plus particulièrement

- les installations avec mise à jour automatique
- l'éditeur d'installation qui permet de créer des installations entièrement personnalisées.

Update réseau / update Web

WinDev propose par défaut plusieurs modes d'installation pour une application WinDev :

- **Installation autonome :**

Ce type d'installation permet de créer un programme d'installation unique. Ce programme d'installation sera exécuté par l'utilisateur final, directement sur son poste.

Pour mettre à jour l'application, il suffira de recréer une installation de l'application. L'utilisateur final devra installer ce nouveau programme.

Nous avons utilisé ce type d'installation avec l'application "MesComptes" dans la partie 2 de ce cours.

- **Installation avec mise à jour automatique :**

Ce type d'installation permet de détecter automatiquement les mises à jour lors du lancement de l'application. Si une mise à jour est disponible, l'utilisateur peut immédiatement se mettre à jour.

Ce type d'installation est disponible aussi bien via réseau ou via le Web.

Détaillons un peu plus le fonctionnement d'une installation avec mise à jour.

Installation avec mise à jour réseau

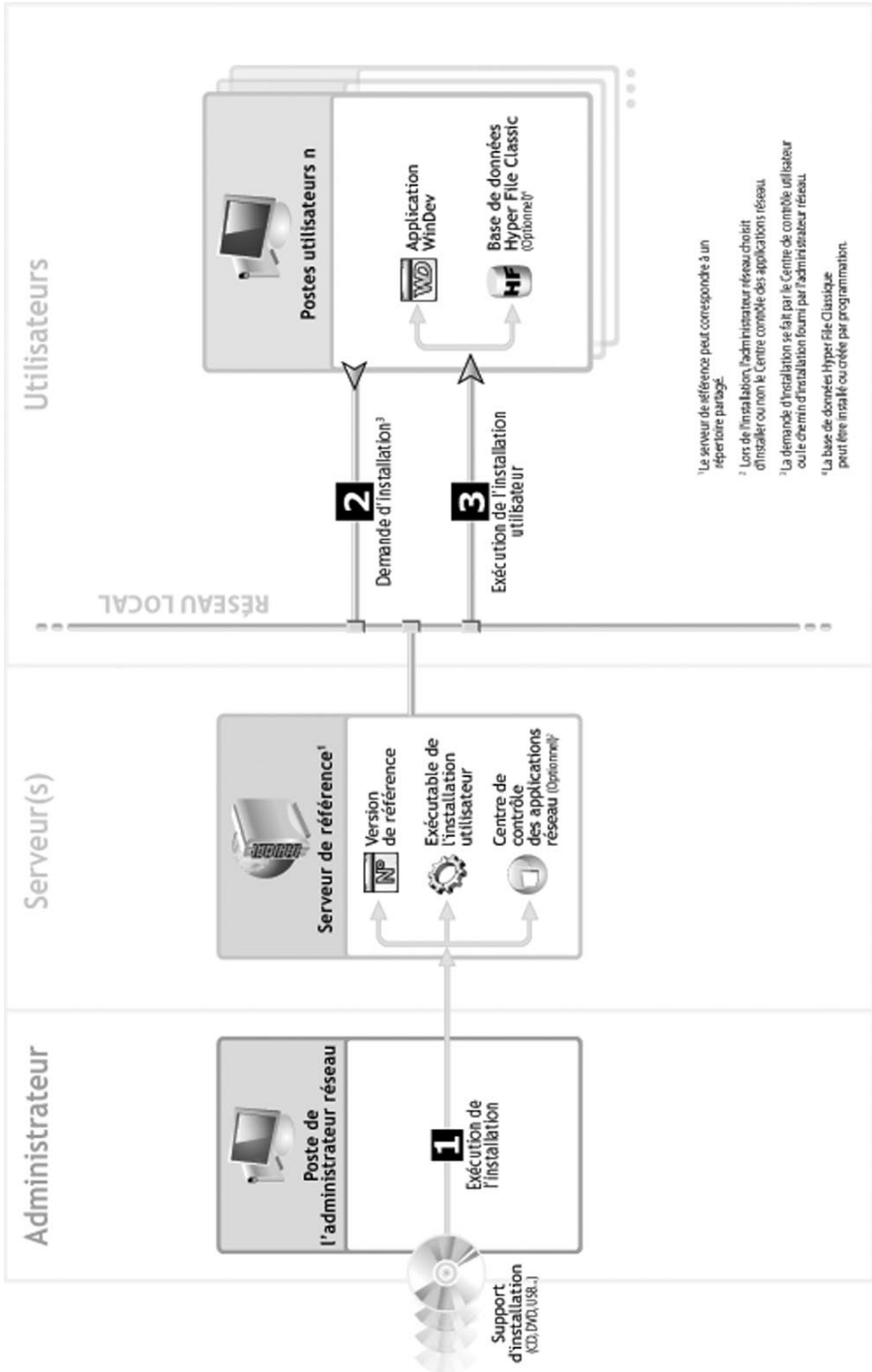
La création du programme d'installation est réalisée grâce à l'assistant de création de programme d'installation (option "Atelier .. Créer la procédure d'installation").

Le programme d'installation obtenu à la fin de l'assistant est en fait composé de deux installations :

- l'installation de l'application de référence. L'application de référence doit être installée sur le réseau (par exemple dans un répertoire partagé ou sur un serveur), accessible par tous les utilisateurs finals de l'application.
- l'installation de l'application elle-même. Cette installation est englobée dans l'application de référence. Elle n'est accessible qu'une fois l'application de référence installée.

Le schéma suivant présente le mode d'installation d'une application utilisant une base Hyper File Classic en local, avec une mise à jour réseau.

A chaque lancement de l'application par l'utilisateur final, la vérification de la version de l'application de référence est automatiquement réalisée. Si cette version a été modifiée (par exemple si une mise à jour de la version de référence a été effectuée), une mise à jour de l'application finale est automatiquement proposée.



Installation avec mise à jour Internet

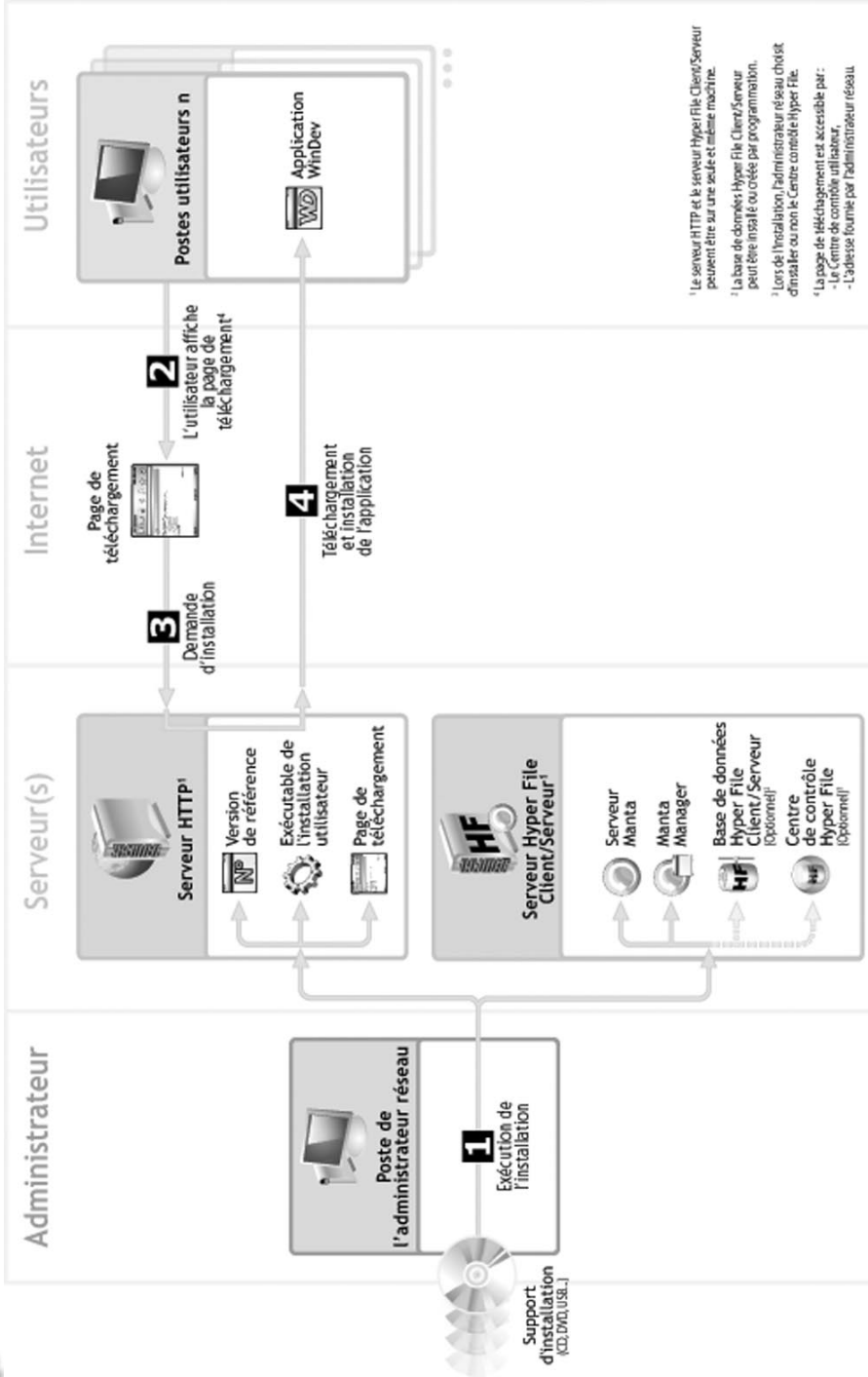
L'installation avec mise à jour par Internet utilise le même principe.

La création du programme d'installation est réalisée grâce à l'assistant de création de programme d'installation (option "Atelier .. Créer la procédure d'installation").

Le programme d'installation obtenu à la fin de l'assistant est en fait composé de :

- l'installation de l'application de référence et la page Web permettant de télécharger l'installation cliente. Ces éléments doivent être installés sur un serveur HTTP.
- l'installation de l'application elle-même. Cette installation est englobée dans l'application de référence. Elle n'est accessible qu'une fois l'application de référence installée, par la page Web de téléchargement.

Le schéma suivant présente le mode d'installation d'une application utilisant une base Hyper File Client / Serveur, avec une mise à jour par Internet.



Editeur d'installation

Nous avons réalisé jusqu'à présent des programmes d'installation avec l'assistant d'installation. Mais pour aller plus loin dans le paramétrage de vos installations, WinDev propose un éditeur d'installation regroupant toutes les options disponibles.

Pour lancer l'éditeur d'installation, sélectionnez l'option de menu "Outils .. WDIInst - Editeur d'installation". L'écran suivant apparaît :

Groupes affichés : <tous les groupes> <RepBase> : C:\Documents and Settings\VP.PCSOFT\Bureau\GAF

Fichier	Répertoire d'origine	Répertoire destination	Date / Heure	Version	Composant
Aide_Ma_première_Bas	<RepBase>\	<RepInst>\	11/10/2007 16		<aucun composant>
banques.txt	<RepBase>\	<RepInst>\	10/09/2007 15		<aucun composant>
banques.wdv	<RepBase>\	<RepInst>\	29/06/2007 12		<aucun composant>
banques.xls	<RepBase>\	<RepInst>\	29/06/2007 12		<aucun composant>
banques.xml	<RepBase>\	<RepInst>\	10/09/2007 15		<aucun composant>
CCMenu.WDK	<RepBase>\	<RepInst>\	04/06/2007 10		<aucun composant>
Ma première Base 12.e	<RepBase>\	<RepInst>\	11/10/2007 16	1.00Af	<aucun composant>

Chemin | Raccourci | Version | Option | Framework

Origine : <RepBase>\Aide_Ma_première_Base_12005.chm

Destination

Répertoire : <RepInst>\

Fichier : <même nom>

Le répertoire peut être décrit en utilisant le paramétrage suivant:

- <RepInst> chemin d'installation (ex: c:\Program Files\WonAppli)
- <LectInst> lecteur d'installation (ex: c:)
- <RepWin> chemin de Windows (ex: c:\Windows)
- <FicInst> répertoire d'installation des fichiers de données
- <srProgramFiles> répertoire "Program Files"

Liste des composants

N°	Français	Anglais

Nouveau Propriétés Supprimer

Cochez le ou les composants auxquels appartient les fichiers sélectionnés. Si un fichier n'appartient à aucun composant, il est toujours installé.

Questions / Réponses

Question Comment désinstaller une application réalisée avec WinDev ?

Lors de la création du programme d'installation, il est possible de prévoir la désinstallation de l'application.

Si cette option a été choisie, le programme de désinstallation a automatiquement été créé. Votre application a donc été enregistrée auprès de Windows pour pouvoir la désinstaller ultérieurement.

Pour désinstaller une application :

1. Cliquez dans le menu "Démarrer".
2. Sélectionnez le choix "Panneau de configuration".
3. Sélectionnez "Ajout/Suppression de programmes".
4. Sélectionnez l'application et cliquez sur "Désinstaller".

Question Comment créer une installation par CD-ROM ?

Lors de la création de l'installation (option "Atelier .. Créer la procédure d'installation"), vous avez la possibilité de choisir différents supports d'installation. En sélectionnant l'option "CD-ROM", WinDev créera un dossier dans lequel tous les fichiers nécessaires à une installation par CD-ROM seront créés (fichier AUTORUN, fichiers d'installation, ...).

Vous n'avez plus alors qu'à graver le contenu de ce dossier complet sur un CD-ROM vierge pour distribuer votre application !

Question Comment installer une application ?

Lorsque l'exécutable a été réalisé (option "Atelier .. Générer l'exécutable (32 ou 64 bits)"), le répertoire EXE présent dans le répertoire de votre projet contient tous les éléments permettant de faire fonctionner votre application.

Pour préparer une installation de votre application :

1. Sélectionnez l'option "Atelier .. Créer la procédure d'installation". L'assistant de préparation d'une installation démarre.
2. Suivez les instructions à l'écran.

LEÇON 9.11. UML ET 3-TIERS

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce que le UML ?
- Les différents types de diagrammes gérés par WinDev.



Durée estimée : 10mn

Qu'est-ce que UML ?

UML (Unified Modeling Language) est une norme de modélisation permettant de "modéliser" des traitements.

Les diagrammes gérés par WinDev

Diagramme de classes

Le diagramme de classes permet de modéliser la structure d'un système et des relations entre les différents éléments composant ce système. Il est principalement utilisé en POO.

Avec WinDev, grâce au dessin d'un diagramme de classe UML, vous pourrez automatiquement générer les classes et les méthodes qui en découlent.

Utilisable en rétro-analyse ("reverse engineering"), vous pourrez alors avoir un aperçu visuel des relations entre les différentes classes que vous aurez créées.

Diagramme de cas d'utilisation

Utilisable une fois le cahier des charges établi (en phase de pré-projet), le diagramme de cas d'utilisation permet d'établir de façon claire et visuelle le comportement de votre projet (correspondance entre les implémentations des demandes des utilisateurs par les développeurs, ...). On nomme parfois ce type de diagramme, "un diagramme de séquence" (sans chronologie particulière).

Il n'est pas possible de générer une application à partir de ce type de diagramme. Ce type de diagramme sert uniquement lors de la mise en place de la gestion de projet.

Diagramme d'objets

Un diagramme d'objets représente un ensemble d'objets et leurs relations à un moment donné.

Un diagramme d'objets est utilisé pour montrer un contexte (avant ou après une interaction entre objets par exemple).

Diagramme de composants

Un diagramme de composants décrit l'architecture physique et statique d'une application informatique. Par exemple : fichiers sources, bibliothèques, exécutables, ...

Diagramme d'activité

Un diagramme d'activité représente le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Diagramme de séquence

Un diagramme de séquence représente l'ordre chronologique des messages envoyés et reçus par un ensemble d'objets.

Diagramme de collaboration

Un diagramme de collaboration présente l'organisation structurelle des objets qui envoient et reçoivent des messages.

Diagramme d'états-transitions

Un diagramme d'états-transitions présente un automate à états finis. Il permet ainsi de décrire les changements d'états d'un objet ou d'un composant.

Un état se caractérise par sa durée et sa stabilité.

Une transition représente le passage instantané d'un état vers un autre.

Une transition est déclenchée :

- soit par un événement.
- soit automatiquement lorsqu'aucun événement déclencheur est spécifié.

Diagramme de déploiement

Un diagramme de déploiement montre la répartition physique des matériels (les noeuds) utilisés dans un système et l'association des programmes exécutables à ces matériels.

Nous ne détaillerons pas plus l'utilisation du langage UML avec WinDev. Consultez l'aide en ligne pour plus d'informations (mot-clé : "UML").

3-Tiers

Le 3 tiers, c'est quoi ?

Le développement en architecture "3-tiers" est maintenant simplifié avec WinDev.

Le but du 3 tiers est de séparer les 3 couches habituelles d'une application : IHM, Traitements et données.

Une application sera composée de 3 couches indépendantes :

- couche présentation
- couche Traitements
- couche d'accès aux données.

Ces 3 couches communiqueront entre elles à l'aide de fonctions spécifiques (des API dans les langages traditionnels, des fonctions avancées en WLangage).

Cette séparation a pour but de rendre indépendantes chacune des couches afin de simplifier la maintenance et les évolutions futures de l'application (par exemple, changement de système de base de données, portage d'un environnement graphique à un autre, ...).

Elle assure une sécurité plus importante car l'accès à la base de données n'est autorisé que par la couche Traitements.

Elle a également l'avantage d'optimiser le travail en équipe et le développement multi-cibles.

Comment mettre en oeuvre le 3 tiers ?

Le dialogue entre les différentes couches s'effectue par classes ou par structures (à votre choix) qui sont générées automatiquement par WinDev.

1. Vous choisissez les sources de données (fichiers ou requêtes que vous désirez manipuler en 3-Tiers).

Vous créez sous l'éditeur d'analyses et l'éditeur de requêtes les différents fichiers et requêtes nécessaires.

2. WinDev construit automatiquement le diagramme de classes qui correspond aux classes et structures qui vont être générées.

Pour cela, sous l'éditeur d'analyses, sélectionnez l'option "Analyse .. Générer le diagramme UML correspondant à l'analyse".

Il est également possible, dans l'éditeur d'UML de créer un diagramme de classes correspondant à l'analyse (Option "Fichier .. Nouveau .. UML .. Diagramme de classe .. Construire le diagramme de classe correspondant à l'analyse et aux requêtes").

Choisissez le type de code à générer (procédural ou objet)

3. Les classes et structures sont générées en WLangage.

Fermez le modèle UML et sélectionnez l'option "Atelier .. Modélisation UML .. Générer le code". Les collections de procédures et/ou les classes sont automatiquement générées.

4. Vous pouvez apporter une "valeur ajoutée" en développant vos propres méthodes dans ces classes.

La synchronisation entre les classes et le diagramme est effectuée en temps réel.

Il est également possible de modifier l'analyse, et de synchroniser le diagramme UML avec l'analyse.

5. Il ne vous reste plus qu'à utiliser ces classes ou structures pour faire circuler les informations entre la couche "Traitement" et la couche "Présentation".

Manipulation des différentes couches

- Pour la couche "Accès aux données" :

Vous manipulez les ordres d'accès aux données habituels : lecture, écriture, requêtes, transactions, ...

- Pour la couche "Traitements" :

Des fonctions WLangage spécifiques peuvent être utilisées : **FichierVersMémoire** et **MémoireVersFichier**. Ces fonctions permettent de remplir les membres de la classe à partir des rubriques des fichiers correspondants, et inversement.

- Pour la couche "Présentation" :

Vous manipulez les classes générées, sans vous soucier de la structure de la base de données.

Pour la communication entre les couches :

Suivant le protocole de communication entre les 2 couches (composant, service Web, HTTP, Site WebDev, ...), les classes peuvent être sérialisées en XML ou dans un format binaire (Fonctions **Sérialise** et **Désérialise**).

LEÇON 9.12. MODÉLISATION SOUPLE

Ce que vous allez apprendre dans cette leçon ...

- Principe
- Manipulation



Durée estimée : 20mn


Principe

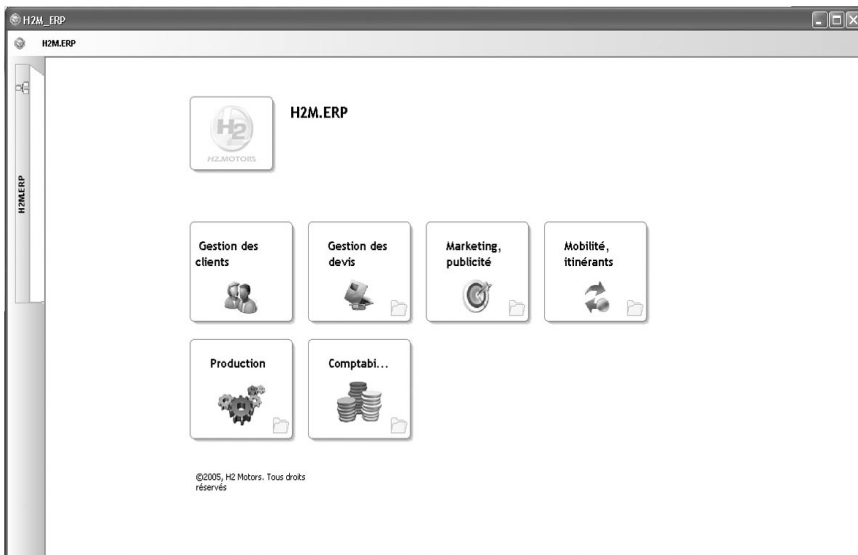
L'éditeur de modélisation permet de représenter toutes les organisations, existantes ou à venir. L'éditeur de modélisation peut par exemple vous permettre de représenter l'organisation de la gestion des mails dans votre entreprise, l'organisation des contacts clients, ...

Pour mieux connaître les possibilités de l'éditeur de modélisation, nous vous avons préparé un exemple de modélisation. Cette modélisation représente une partie de l'ERP (Entreprise Ressource Planning) de l'entreprise fictive H2M.

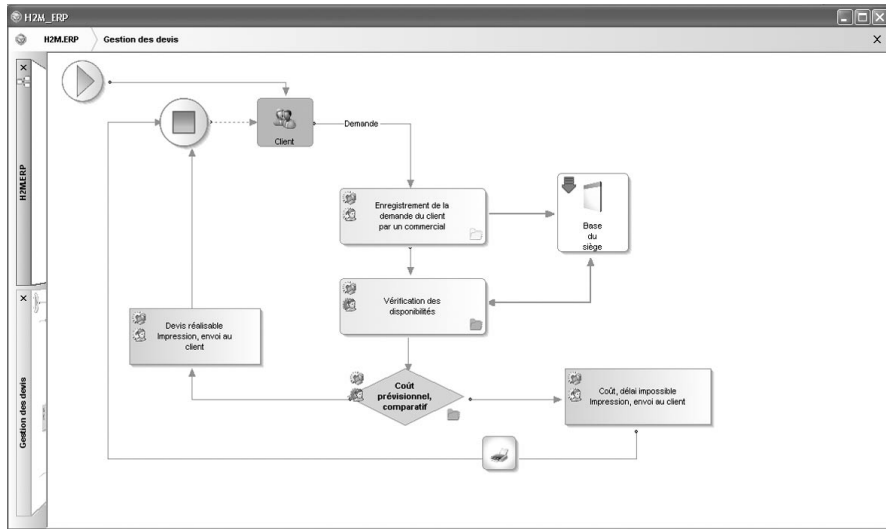
Manipulations

- ▶ Ouvrez le projet "Modelisation.wdp". Ce projet est présent dans le sous-répertoire "Auformation\Corrigés\Modelisation" du répertoire d'installation de WinDev 12.
- ▶ Ouvrez le fichier "H2M.ERP.wem" (option "Fichier .. Ouvrir" ou en double-cliquant sur la modélisation dans le volet "Projet").
Dans cette modélisation, vous constatez que plusieurs grands domaines sont présents : Gestion des clients, des devis, marketing, ...

L'icone  signale qu'une sous-modélisation est présente. En double-cliquant sur cet icone, vous accédez à cette sous-modélisation.

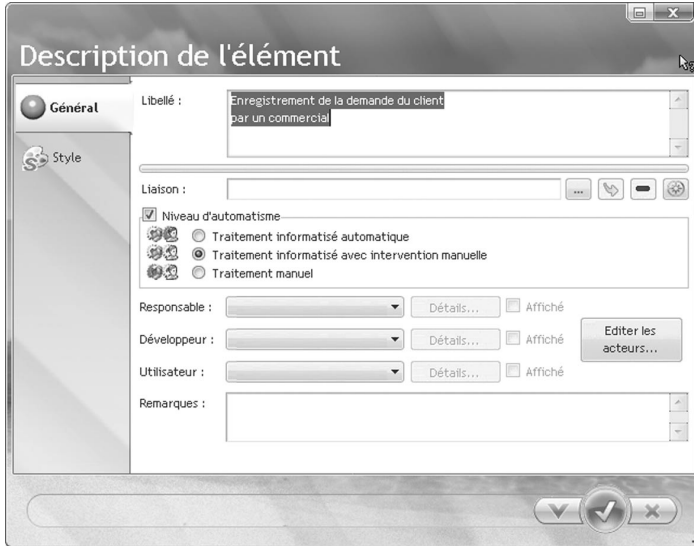


- ▶ Dans notre exemple, ouvrez la sous-modélisation correspondant à la gestion des devis. Vous retrouvez les différentes étapes de la gestion des devis :



- ▶ Regardons en détail l'élément "Base du siège". Une flèche est présente en haut à gauche de cet élément. Cette flèche indique une liaison. Si vous double-cliquez sur la flèche, l'analyse de l'application s'ouvre automatiquement. C'est une fonctionnalité importante de l'éditeur de modélisation : tout élément de la modélisation peut être lié à un élément de votre projet (fenêtre, code, ...).
- ▶ Tous les éléments de la modélisation ont des caractéristiques. Ces caractéristiques peuvent être affichées en sélectionnant l'option "Description" du menu contextuel de l'élément.

Par exemple :



Cette fenêtre permet d'indiquer :

- la liaison de l'élément avec un fichier externe (analyse, fenêtre, ...)
- le niveau d'automatisme de l'élément : traitement réalisé automatiquement, manuellement, ...
- les acteurs intervenant dans ce traitement (responsable, utilisateurs, développeurs, ...). Ces informations peuvent provenir directement des intervenants décrits au niveau du Centre de Suivi de Projets.

Pour plus de détails sur le centre de modélisation, consultez l'aide en ligne.

LEÇON 9.13. MULTILANGUE

Ce que vous allez apprendre dans cette leçon ...

- Qu'est-ce qu'une application multilingue ?
- Etape par étape, création d'une application multilingue.



Durée estimée : 20mn

Qu'est-ce qu'une application multilingue ?

C'est une application qui peut s'exécuter en anglais, en français, en allemand ou dans une autre langue.

Une seule et même application peut donc être utilisée dans plusieurs langues. Mais comment est-ce possible ?

C'est ce que nous allons voir dans cette leçon.

Nous allons manipuler un projet qui pourra s'exécuter en français ou en anglais, selon le choix de l'utilisateur.

Pour voir un exemple d'application Internationale, sélectionnez l'option "? .. Guide d'auto formation .. Application multilingue". Pour changer la langue, utilisez l'option "Fichier .. Langue".

Les principales étapes d'une application multi-langue sont :

- Le choix des langues du projet.
- L'internationalisation de l'analyse.
- L'internationalisation des éléments du projet (fenêtres, états, champs, aide, ...).
- L'internationalisation des messages présents dans le code.
- L'internationalisation des fonctions WLangage.
- La programmation du changement de langue dans l'application.

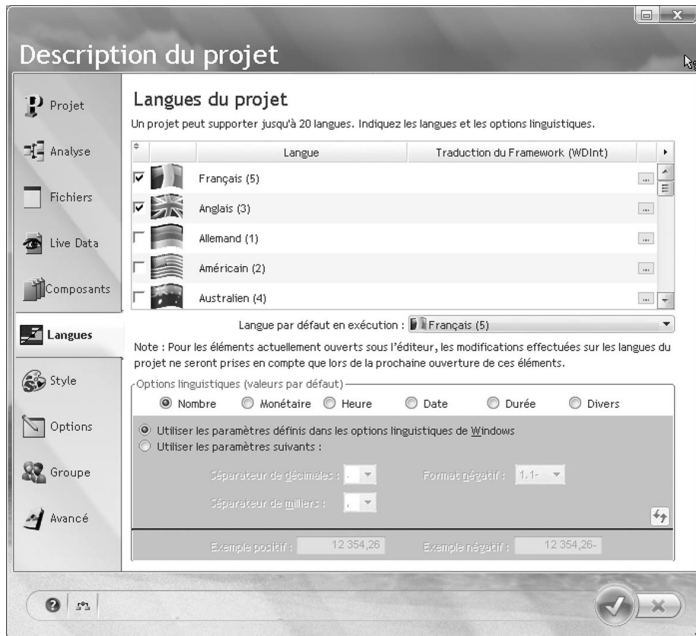
Nous allons appliquer ces différentes étapes au projet "WD Bourse International". Ce projet, disponible en Français, va être traduit en Anglais et en Arabe (avec inversion du sens d'écriture). La version complète de ce projet multi-langues est disponible dans le sous-répertoire "\Autoformation\Corrigés\WD Bourse International".

Choix des langues du projet

- ▶ Ouvrez le projet "WD Bourse International.WDP" (dans le sous-répertoire "Autoformation\Exercices\WD Bourse International" du répertoire d'installation de WinDev). L'analyse associée est "WD Bourse". Vous pouvez vérifier cette information dans la description du projet (option "Projet .. Description du projet", onglet "Analyse").

La première opération consiste à choisir les langues du projet.

- ▶ Affichez la description du projet si ce n'est déjà fait (option "Projet .. Description du projet").
- ▶ Cliquez sur l'onglet "Langues". Notre application va gérer le Français, l'Anglais et l'Arabe : sélectionnez ces trois langues.



- ▶ Cet onglet permet également de paramétrer les options linguistiques concernant les nombres, les monétaires, les dates, ... pour la langue sélectionnée. Voyons un exemple :
 - Cliquez sur la langue "Anglais".
 - Cliquez sur "Date"
 - Par défaut, les options linguistiques de Windows sont utilisées. Si vous sélectionnez l'option "Utiliser les paramètres suivants", il est possible de définir le format de date utilisé ainsi que la traduction des jours et des mois.
 - Conservez l'option "Utiliser les paramètres définis dans les options linguistiques de Windows".
- ▶ Validez en cliquant sur "OK".



Notes

Dans les options linguistiques, il est possible de choisir le sens d'écriture de la langue (case "Divers", option "Sens d'écriture"). Il est ainsi possible de créer des interfaces avec une langue s'écrivant de droite à gauche.

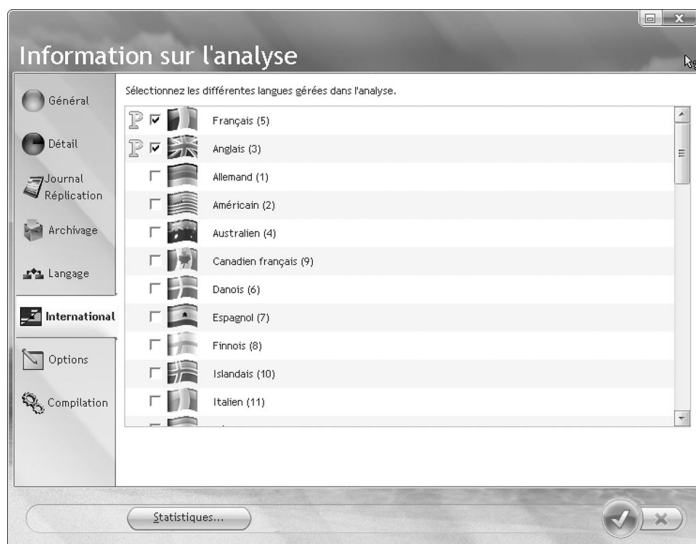
Internationalisation de l'analyse

Par défaut, une analyse est créée dans une langue et ne peut pas être traduite. Cependant, certaines informations peuvent être saisies dans plusieurs langues (notes du dossier, informations partagées, ...). Si votre application utilise le module Etats et Requêtes, il est également possible de traduire les noms de fichiers et les noms de rubriques. Cette traduction peut être réalisée dans l'onglet "Etats et Requêtes" (de la description de fichiers ou de rubriques).

Les champs créés à partir des rubriques de l'analyse ont par défaut le libellé de la rubrique spécifié dans l'analyse. Si un libellé a été spécifié dans les informations partagées de la rubrique, ce libellé sera utilisé lors de la création du champ.

Pour gérer plusieurs langues dans une analyse :

1. Dans la description de l'analyse (option "Analyse .. Description de l'analyse"), sélectionnez l'onglet "International".
2. Indiquez les différentes langues gérées par l'analyse et validez.



Pour saisir les informations partagées dans plusieurs langues :

1. Affichez la description des rubriques d'un fichier (option "Structure de fichiers .. Rubriques").
2. Pour chaque rubrique, affichez les paramètres du champ relié à la rubrique sélectionnée

(informations partagées). Pour cela, cliquez sur la double flèche en bas de l'écran.



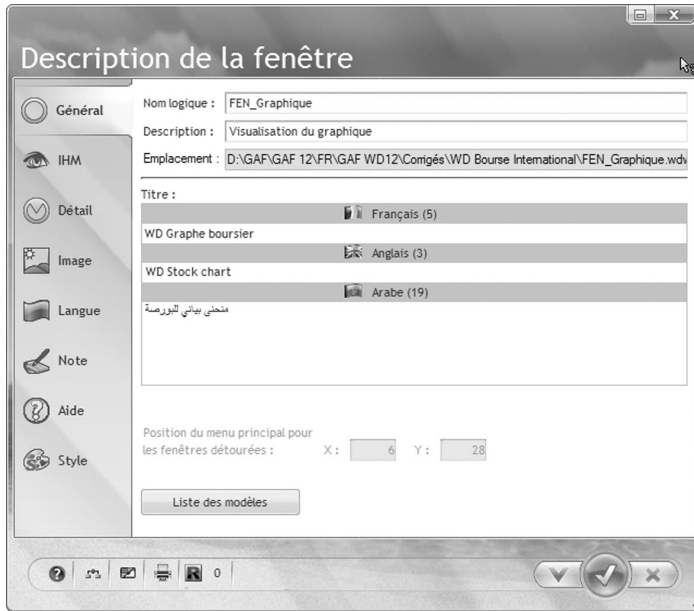
Internationalisation des éléments du projet

Tous les éléments du projet peuvent devenir multilingues : fenêtres, états, aide, ...
Regardons tout d'abord les caractéristiques d'une fenêtre et de ses champs.

Caractéristiques à traduire

- ▶ Ouvrez la fenêtre "FEN_Graphique.WDW".
- ▶ Affichez la description de la fenêtre (option "Description" du menu contextuel de la fenêtre).

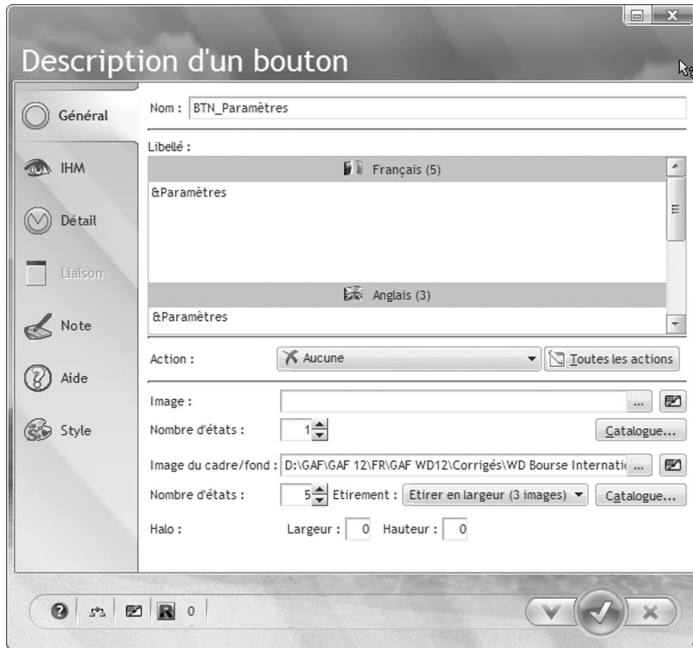
Sélectionnez l'onglet "Langues" : les trois langues sélectionnées dans le projet apparaissent.



Sélectionnez l'onglet "Général" : le titre de la fenêtre doit être traduit.
Fermez cette fenêtre.

- Affichez la description du bouton "Paramètres". Regardons les différentes zones concernées

par la gestion des différentes langues



Dans la zone "Libellé", les trois langues sont disponibles. Faites défiler la zone "Libellé".

Comme le projet n'a pas encore été traduit, le contenu de la langue "Français" a été recopié dans les autres langues.

Dans la zone "Image du cadre", le bouton  permet de configurer la gestion du multilingue. Cliquez sur ce bouton.



Dans la fenêtre qui s'affiche, il est possible d'utiliser une image différente pour chaque langue. Cette fonctionnalité est très utile si vous utilisez des images contenant du texte. Fermez cette fenêtre.

Cliquez sur l'**onglet "Aide"**. Cet onglet contient les différents messages d'aide associés au champ. Ici aussi, ces informations doivent être traduites.

Le même type d'informations est également à traduire pour :

- tous les champs des fenêtres
- les fenêtres
- les états,
- les champs des états
- les messages texte présents dans l'éditeur de code.

Comment traduire toutes ces informations ?

Pour traduire ces informations, WinDev met plusieurs moyens à votre disposition :

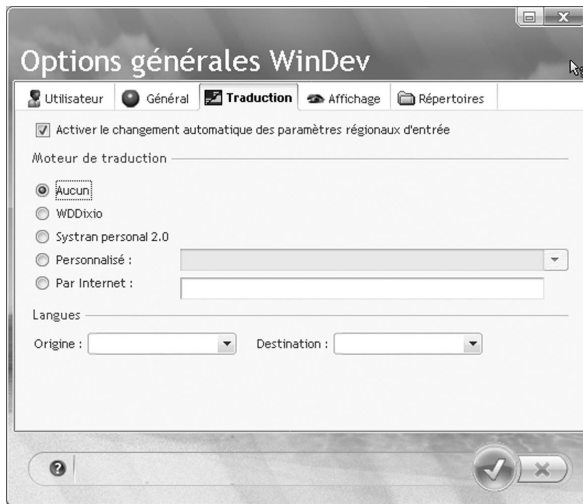
- une traduction directe des messages dans les différents éditeurs.
- une traduction réalisée grâce à un outil externe (WDMMSG et WDTRAD)

Saisie directe des traductions

Les traductions sont directement saisies dans l'interface du produit. Par exemple, le libellé du bouton "Paramètre" devient "Setting" en anglais. Il suffit d'ouvrir la fenêtre de description du champ et de saisir dans la langue voulue la traduction correspondante.


Si vous disposez d'un logiciel de traduction ou d'un site de traduction, vous pouvez configurer WinDev pour utiliser ce logiciel :

1. Sélectionnez l'option "Outils .. Options de WinDev".
2. Affichez l'onglet "Traduction".



3. Spécifiez :

- Si les paramètres régionaux doivent être automatiquement activés selon la langue en cours de saisie. Dans ce cas, si la langue nécessite un alphabet spécifique, cet alphabet sera automatiquement sélectionné.
- Le logiciel ou le site à utiliser pour la traduction.
- Les langues gérées.

4. Lorsque les paramètres de traductions sont définis, il est possible d'utiliser le bouton  pour utiliser le logiciel défini pour la traduction.

Cas particuliers

Traduction directe des menus

La traduction des options d'un menu peut être réalisée directement à partir de l'éditeur de fenêtres.

Pour traduire le menu d'une fenêtre :

1. Ouvrez la fenêtre "FEN_Menu.WDW".
2. Sélectionnez l'option "Affichage .. Langue affichée..." et sélectionnez la langue à visualiser sous l'éditeur.

Les options de menu s'affichent dans la langue sélectionnée. Si aucune traduction ne correspond à la langue sélectionnée, les options de menu s'affichent en français.

3. Saisissez directement le libellé des options de menu dans la langue sélectionnée.

Traduction directe des messages de programmation

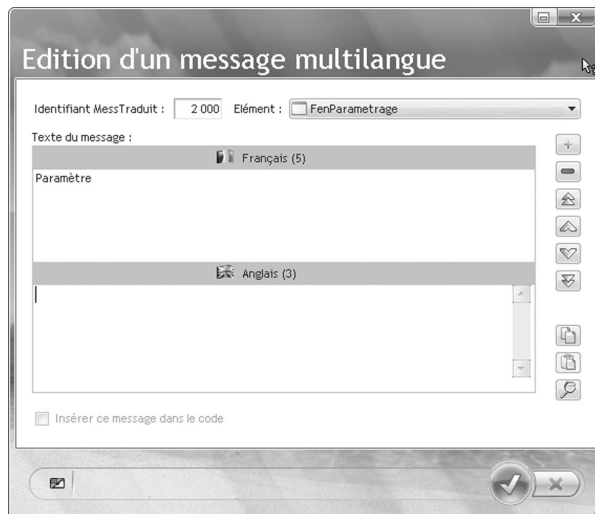
Tous les messages de votre programme peuvent également être saisis dans plusieurs langues.

Deux types de messages peuvent être rencontrés :

- les chaînes de caractères simples, par exemple :


```
// Mémoriser le type du graphe en lettre (Bar-charts, chandelle,...)
[INIIEcrit("Paramètres", "Nom", TypeGraphe[TypeGraphe], FichierIni)
```

Pour traduire ce type de message, utilisez l'option "Code .. Messages multilingues .. Traduction des messages". La fenêtre suivante apparaît :



Cette fenêtre permet de saisir tous les messages de votre programme dans toutes les langues du projet.

Un numéro est attribué à chaque message.

Lorsqu'une traduction est saisie pour le message, l'icone  apparaît sous l'éditeur de code.

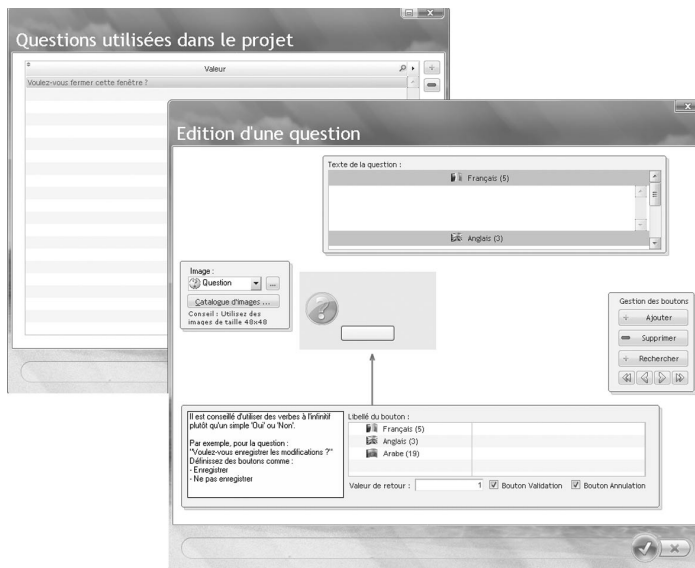
```
// Mémoriser le type du graphe en lettre (Bar-charts, chandelle,...)
[INIIEcrit("Paramètres", "Nom", TypeGraphe[TypeGraphe], FichierIni)
```

- les interrogations directives, par exemple :

```

FenCours - Clic sur Fermer *
  Initialisation de Fermer  Si Erreur : par programme  Quand E...
  Clic sur Fermer *      Si Erreur : par programme  Quand E...
    //1 : Fermer
    //0 : Ne pas fermer
    DélaiAvantFermeture(1000)
    SELON Dialogue ("Voulez-vous fermer cette fenêtre ?")
      // Fermer
      CAS 1
      // Ne pas fermer
      CAS 0
    FIM
    Ferme {}
  
```

Pour traduire ce type de message, utilisez l'option "Code .. Message multilingues .. Interrogations directives". La liste des interrogations directives s'affiche. Il suffit de double-cliquer sur la phrase à traduire.



Traduction directe des fichiers d'aide

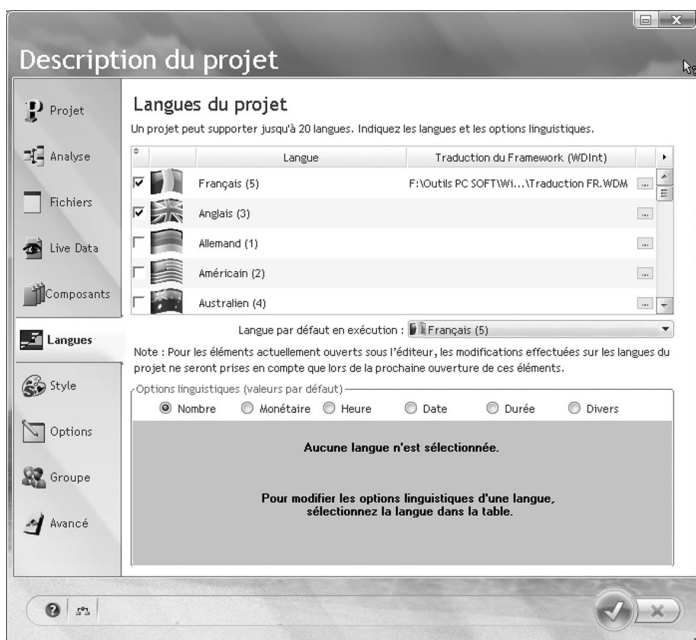
La traduction d'un système d'aide peut être réalisée directement sous l'éditeur d'aide. Il suffit de changer la langue d'affichage. Si le système d'aide n'a pas encore été traduit, le texte apparaîtra dans la langue originale (Français si le texte a été saisi en Français).

Traduction du résultat des fonctions WLangage et des messages d'erreurs du framework

De nombreuses informations et messages sont présents dans le framework WinDev. Par exemple, les noms des jours et des mois utilisés par les fonctions de gestion des dates proviennent du framework WinDev. Pour traduire une ou plusieurs librairies de ce framework, il est nécessaire d'utiliser WDINT (non fourni en standard avec WinDev).

Ce logiciel permet d'obtenir un fichier d'extension WDM. Pour utiliser ce fichier dans votre application :

- vous pouvez utiliser la fonction ChargeErreur.
- vous pouvez intégrer le fichier à la description du projet (onglet "Langues", option "Traduction du Framework (WDInt)").



Pour obtenir plus d'informations sur **WDINT**, contactez le Service Commercial de PC SOFT.

Traduction avec WDMMSG et WDTRAD

Il existe également un outil nommé **WDMMSG** (non fourni en standard avec WinDev) qui permet

- d'extraire tous les messages d'un projet (libellé des champs, message de code, titre des fenêtres, ...) pour les traduire,
- de ré-intégrer les messages traduits.

Les messages à traduire sont extraits dans un format texte, pouvant être configuré pour être utilisé avec la majorité des logiciels de traduction.

WDMMSG est également fourni avec un outil d'aide à la traduction, WDTRAD. WDTRAD permet de saisir simplement toutes les traductions des informations multilingues d'un projet.

Pour obtenir plus d'informations sur **WDMMSG** et **WDTRAD**, contactez le Service Commercial de PC SOFT.

Programmation du changement de langue

Par défaut, le projet s'exécute dans la langue d'exécution définie pour le projet (option "Projet .. Description du projet", onglet "Langues").

La fonction **Nation** permet de changer de langue.

Dans une application, le choix de la langue peut être défini selon deux méthodes :

- mémorisation de la langue dans un fichier de paramètres propre à l'application.
- sélection de la langue par une option de menu.

Utilisation d'un fichier de paramètres

Dans le cas d'un fichier de paramètres, le choix de la langue est totalement transparent pour l'utilisateur.

La langue peut être choisie par exemple lors de l'installation du projet ou lors de premier lancement de l'application, et pourra être modifiée par la suite.

Pour déterminer la langue dans laquelle s'exécutera l'application, il est nécessaire de lire ce fichier de paramètres dans le traitement d'initialisation du projet.

Le traitement est le suivant :

```
gsLangue est une chaîne; gsNomFic est une chaîne
//Lecture du fichier de paramètres
gsNomFic = "INTERNATIONAL.INI"
gsLangue = INILit("PARAMETRE", "LANGUE", "FRANCAIS", gsNomFic)
SELON gsLangue
  CAS "FRANCAIS" : Nation(NationFrançais)
  CAS "ANGLAIS" : Nation(NationAnglais)
  CAS "ARABE" : Nation(NationArabe)
FIN
```

Ce traitement nécessite quelques commentaires :

- La langue choisie est lue dans le fichier "International.INI"
- La fonction **Nation** permet de modifier la langue d'exécution du projet.
- La constante passée en paramètre à la fonction **Nation** correspond à la langue choisie.

Il est nécessaire d'utiliser la fonction **Nation** dans le traitement d'initialisation du projet car le changement de langue sera effectif dans la prochaine fenêtre qui s'ouvrira.

En effet, si la fonction **Nation** est appelée dans le code d'ouverture de la première fenêtre du projet, cette fenêtre sera dans la langue par défaut et le changement de langue ne sera effectif que pour les autres fenêtres.

Par contre, si la fonction **Nation** est appelée dans le code d'initialisation du projet, la première fenêtre du projet sera dans la langue sélectionnée par la fonction **Nation**.

Utilisation des options de menu

Dans le menu de la fenêtre "FEN_Menu.WDW", l'option "Fichier.. Langue" permet de changer la langue. Le code associé à ces options est le suivant :

```
- - Choix menu : &Fichier..&Langue.....&Français
Nation(NationFrançais)
Utilise(FEN_Menu)

- - Choix menu : &Fichier..&Langue.....&Anglais
Nation(NationAnglais)
Utilise(FEN_Menu)
```

La fonction **Utilise** permet de recharger la fenêtre passée en paramètre. Le changement de langue effectué par la fonction **Nation** est donc immédiatement effectif.

► Testez le projet.

Le fichier "International.INI" contient par exemple "ANGLAIS". Ainsi le menu s'affiche en anglais, de même que le titre de la fenêtre.

Gestion des alphabets spécifiques

Si votre application gère des langues utilisant des alphabets spécifiques (grec, coréen, ...), changer par programmation :

- l'alphabet utilisé par toutes les polices écran (fonction **ChangeAlphabet**). Cet alphabet pourra également être utilisé pour les données enregistrées dans les fichiers Hyper File. Pour plus de détails, consultez l'aide en ligne (mot-clé : Multilangue).
- la langue du clavier manipulé par les utilisateurs de l'application (fonction **ChangeClavier**).



CONCLUSION

Le cours est maintenant terminé !

Ce cours a abordé un ensemble de sujets, mais pas la totalité des fonctionnalités de WinDev, loin s'en faut !

Vous êtes maintenant familiarisé avec les principaux concepts.

Nous vous conseillons de consacrer une journée supplémentaire à explorer les différents choix de menu de WinDev, pour chacun des modules.

Explorez également les exemples livrés avec WinDev : certains sont simples et ne traitent que d'un sujet, d'autres sont très complets. Ces exemples vous montreront les différentes facettes de WinDev. La lecture du code source sera également instructive en général.

Faute de place, nous n'avons pas pu aborder tous les sujets (il y en a des centaines, voire des milliers !). WinDev offre de nombreuses possibilités non abordées ou non approfondies dans ce cours :

- groupware utilisateur
- fonctions sockets, HTTP, téléphonie
- création de gabarits ...
- états imbriqués, requêtes paramétrées ...
- compilation dynamique, appels de DLL, langages externes ...

Pour plus de détails sur toutes ces possibilités, n'hésitez pas à consulter l'aide en ligne.

Rappel : Pour recevoir directement des mises à jour intermédiaires et des conseils d'utilisation, abonnez-vous à la **LST** (revue trimestrielle + DVD), en français.

Nous vous souhaitons de bons développements, avec **WinDev 12, AGL n° 1 en France !**