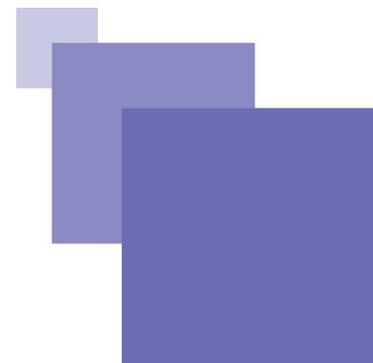


# WebDev : initiation & perfectionnement

0.2

PASCAL MIETLICKI - CC-BY-NC-SA -  
[HTTP://CREATIVECOMMONS.ORG/LICENSES/BY-NC-SA/3.0/](http://creativecommons.org/licenses/by-nc-sa/3.0/)

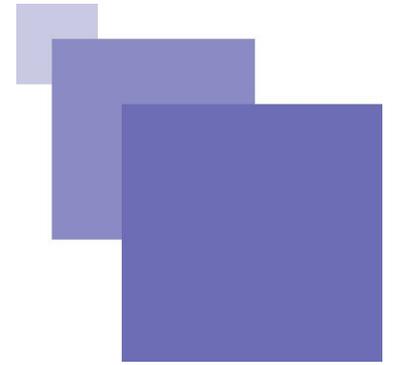
# Table des matières



<b>Objectifs</b>	<b>5</b>
<b>Introduction</b>	<b>7</b>
<b>I - Quelques Notions</b>	<b>9</b>
A. Rappels.....	9
1. Principe.....	9
<b>II - Apprentissage WebDev</b>	<b>11</b>
A. Différents types de site.....	11
1. Site dynamique.....	12
B. Concepts généraux / Informations utiles.....	12
1. Raccourcis.....	15
C. Interface : prise en main.....	16
1. Code des champs.....	16
2. Ordre d'exécution.....	16
3. Exemple de code.....	16
4. Utilisation des groupes.....	31
5. Tables.....	32
6. Zones répétées.....	33
7. Popup.....	37
8. Multilinguisme.....	40
9. Cookies.....	40
10. Pages internes et composants.....	41
11. Refactoring.....	42
12. Upload / Download de fichiers.....	43
13. Crypter les données d'une table.....	44
14. Gestion des droits.....	44
D. Règles de "bonne conduite".....	45
E. Analyse et BDD.....	45
1. Création de l'analyse.....	47
2. Création des fichiers de l'analyse.....	49
3. Création des rubriques des fichiers.....	53
F. WLangage.....	54
1. Syntaxe de base.....	54
2. Fonctions utiles.....	58
3. Les boucles.....	60
4. Interaction avec la BDD.....	60
5. Boîtes de dialogue.....	65
6. Passage de paramètres.....	66
7. Fonctions d'impression.....	66

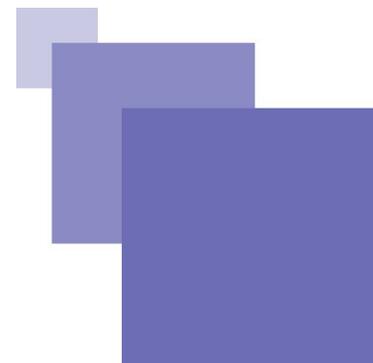
8. Répertoire du serveur.....	67
<b>III - Création d'un projet</b>	<b>69</b>
A. Créer un projet.....	69
B. Gestion de projet.....	71
C. RAD.....	72
D. GDS.....	73
E. Charte de programmation.....	74
F. Ambiance du projet.....	74
G. Choix de la langue.....	75
H. Choix de la base de données.....	75
I. Pages et modèles.....	76
1. Notion de première page dynamique.....	78
2. Modèle de page.....	79
J. Utilisation d'AJAX.....	79
<b>IV - Administration et débogage</b>	<b>81</b>
A. Débogage.....	81
1. Débogage "simple".....	81
2. Audit.....	85
B. Déploiement d'un site.....	88
C. GDS.....	90
<b>Conclusion</b>	<b>93</b>
<b>Index</b>	<b>95</b>

# Objectifs



- Maîtriser les fondamentaux pour réaliser un site Web sous WebDev
- Prendre en main l'environnement de développement WebDev

# Introduction



Webdev est un IDE (*Integrated Development Environment*<sup>1</sup>) pour le développement de sites Intranet et Internet.

Il permet de concevoir *différents types de sites*<sup>2</sup> :

- Dynamiques (avec base de données)
- Semi-dynamiques : une partie dynamique et une partie statique
- Statiques (sans base de données)

WebDev est composé de différents modules d'édition dont les principaux sont :

- Éditeur de projet pour la gestion du projet
- Éditeur de pages
- Éditeur d'analyses pour la modélisation de votre BDD
- Éditeur de code avec coloration syntaxique, débogueur, assistants, aide contextuelle et un outil de recherche
- Éditeur de styles (pour les CSS<sup>3</sup> ou modèles WebDev)
- Éditeur de requêtes
- Éditeur d'installation pour la mise en place des sites créés via FTP ou un installateur spécifique

WebDev respecte plutôt bien l'approche WYSIWYG.

Ce module d'apprentissage va vous permettre d'apprendre à réaliser une application complète.

## Avantages

- Prise en main rapide
- Rapidité de développement
- Un "cliquodrome" qui facilite le développement d'applications Web pour les néophytes
- Une multitude d'outils "clef en main"

## Inconvénients

- Code généré non optimal
- *Programmation procédurale*<sup>4</sup>
- Une interface "chargée" : beaucoup de modules/options surchargent l'interface
- Questionnement sur l'aspect "sécurité" du code généré : effet "boite noire"
- Difficulté à adapter les modules intégrés lors de besoins spécifiques (exemple du groupware surtout compatible avec *active directory*<sup>5</sup>)
- Des effets de bord lorsqu'on utilise d'autres produits que ceux fournis (exemple de MySQL au lieu d'Hyperfile la BDD intégrée par défaut)

1 - [http://fr.wikipedia.org/wiki/Environnement\\_de\\_d%C3%A9veloppement\\_int%C3%A9gr%C3%A9](http://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9)

2 - <http://doc.pcsoft.fr/fr-fr/?2030012>

3 - [http://fr.wikipedia.org/wiki/Feuilles\\_de\\_style\\_en\\_cascade](http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade)

4 - [http://fr.wikipedia.org/wiki/Programmation\\_proc%C3%A9durale](http://fr.wikipedia.org/wiki/Programmation_proc%C3%A9durale)

5 - [http://fr.wikipedia.org/wiki/Active\\_directory](http://fr.wikipedia.org/wiki/Active_directory)

**En définitive, l'outil répondra très bien à des besoins liés à des applications de gestion par exemple ou si vous n'avez pas ou peu de connaissances en programmation. Pour des applications plus complexes où la sécurité, la rapidité et l'interopérabilité sont primordiales, il vaudra mieux privilégier une autre approche (*symfony*<sup>6</sup>, *zend*, *sensa touch*<sup>7</sup>, *bootstrap*<sup>8</sup>, etc).**

6 - <http://symfony.com/>

7 - <http://www.sencha.com/products/touch>

8 - <http://twitter.github.com/bootstrap/>

# Quelques Notions

## A. Rappels

*WWW*<sup>9</sup> (World Wide Web) est une application d'Internet permettant de consulter avec un navigateur des informations enrichies (hypertexte) tel que des images, de la vidéo, du texte via différentes pages. *HTTP* (pour *HyperText Transfer Protocol*) est le protocole de communication communément utilisé pour transférer les ressources du web. *HTTPS* est la variante sécurisée de ce protocole.<sup>10</sup> Sous WebDev, cela se traduit par la création d'une ou plusieurs "pages" qui constitue un site.

### 1. Principe

Les pages Web sont visualisées via un navigateur Web (Internet Explorer, Google chrome, Firefox, Opera, etc). Ce navigateur va interpréter le contenu d'un fichier au format *HTML*<sup>11</sup> pour un rendu visuel à l'utilisateur. On parle donc de page HTML.



#### *Remarque : Différence HTTP / HTML*

*HTTP*<sup>12</sup> est le protocole de communication qui englobe un ensemble de méthode. En gros, cela est similaire aux langues, pour que 2 interlocuteurs (un formateur et un étudiant par exemple) se comprennent, il faut qu'ils parlent la même langue, c'est la même chose pour le protocole : le serveur et le client doivent se comprendre à travers un protocole de communication commun qu'est le *HTTP*<sup>13</sup>.

*HTML*<sup>14</sup> est le format de données interprétées, le "langage" permettant d'interpréter les données, c'est un langage de balisage permettant d'écrire de l'*hypertexte*<sup>15</sup>.



#### *Complément : Notion de statique / dynamique*

A la base le Web était statique, c'est à dire n'affichant que le contenu interprété des pages HTML sans interaction avec l'utilisateur.

La notion d'affichage "dynamique" permettant une meilleure interaction avec

9 - <http://fr.wikipedia.org/wiki/Www>

10 - <http://fr.wikipedia.org/wiki/HTTP>

11 - <http://fr.wikipedia.org/wiki/HTML>

12 - <http://fr.wikipedia.org/wiki/HTTP>

13 - <http://fr.wikipedia.org/wiki/HTTP>

14 - [http://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language](http://fr.wikipedia.org/wiki/Hypertext_Markup_Language)

15 - <http://fr.wikipedia.org/wiki/Hypertexte>

L'utilisateur est notamment apparu avec le *javascript*<sup>16</sup>.

Cela s'est amplifié avec des langages de script interprétés côté serveur permettant un rendu "personnalisé" à l'utilisateur tel que php, perl, cgi, python, etc.

16 - <http://fr.wikipedia.org/wiki/JavaScript>



# Apprentissage WebDev



Différents types de site	11
Concepts généraux / Informations utiles	12
Interface : prise en main	16
Règles de "bonne conduite"	45
Analyse et BDD	45
WLangage	54

## A. Différents types de site

### *Statique, semi-dynamique et dynamique*

Un **site statique** est exactement la même chose qu'une page HTML de base laissée sur un serveur Web, c'est à dire un contenu défini à l'avance qui n'évoluera pas en fonction de l'utilisateur, ne présentera pas d'interaction avec l'utilisateur.

Un **site semi-dynamique** est un site statique mais interfacé avec une base de données. Le contenu sera récupéré de la base, par exemple, un catalogue de produit qui sera affiché à l'utilisateur.

Un **site dynamique (le plus commun)** est généralement interfacé avec une base de données, il permet une interaction avec l'utilisateur, le contenu évoluant en fonction des actions effectués par l'utilisateur via des traitements effectués par le serveur (calcul, accès aux bases, etc) et des traitements effectués par le navigateur (contrôle des entrées, affichage, etc).



### *Remarque : Pour rappel*

Un traitement effectué côté navigateur équivaut généralement à un script javascript exécuté directement par le navigateur, le serveur n'étant alors pas sollicité.

Un traitement effectué côté serveur équivaut à un langage de script type php, il y a alors un "aller-retour" avec le serveur avant l'envoi de la réponse à l'utilisateur.



### *Attention : Différence importante !*

Ces notions sont structurantes sous WebDev. En effet, des fonctions sont parfois uniquement accessible en code serveur (cas le plus fréquent) ou vice-versa uniquement en code navigateur.

Il faut donc faire attention à cela en s'assurant d'utiliser la fonction au bon endroit ou en forçant l'exécution d'un code serveur (via ExécuteTraitement par exemple)

dans le cas où la fonction n'existerait pas côté navigateur.

## 1. Site dynamique

Pour un site dynamique, WebDev permet de générer des scripts PHP ou AWP.

Dans le cas d'un site PHP, un serveur d'applications WebDev ne sera alors pas nécessaire, il sera possible de déployer le programme directement sur un serveur apache par exemple. **Cependant, les fonctionnalités seront limitées.**

Dans le cas d'un site AWP (Active WebDev Page), toute la "valeur ajoutée" de WebDev sera présente mais il faudra alors avoir à sa disposition un serveur d'applications WebDev payant pour pouvoir exécuter le code généré.

### *Avec ou sans session*

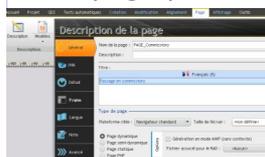
**Un site dynamique classique WebDev contient des sessions**, cela permet de "cloisonner" les variables et données d'un utilisateur au sein d'un même conteneur. La session étant limitée dans le temps, cela a l'avantage de "sécuriser" l'accès. Le problème est qu'un tel site, au vu des URL générées, ne sera pas référençable pour les moteurs de recherche type google.

**On peut aussi désactiver les sessions**, l'application se comportant alors comme toute application Web standard, **les fonctions WebDev seront plus limitées** mais le site sera référençable.



### *Remarque : Mélange session / sans session*

WebDev permet de "mixer" les 2 concepts (avec et sans session) via un clic droit sur la page puis dans "dynamique", on indique avec ou sans contexte :



Un site sans session générera des pages du type :

`http://localhost/sav/page_listing.awp`

Pour sauvegarder les informations, on pourra passer des paramètres soit dans l'URL (type GET) soit utiliser des contextes (type POST).

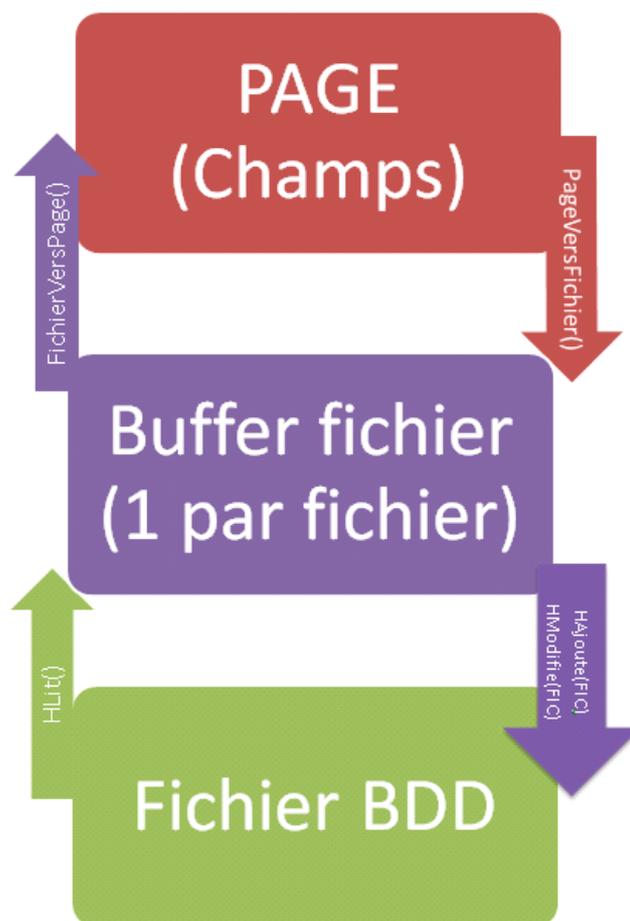
## B. Concepts généraux / Informations utiles



### *Remarque : Fonctionnement général*

Par défaut, WebDev gère une sorte de "buffer" intermédiaire qui sera rempli à partir de la base de données ou à partir de l'affichage (et vice-versa).

Voici un schéma générique expliquant ce fonctionnement :



*Fonctionnement interne buffer*

### Champs de saisie liés à la BDD

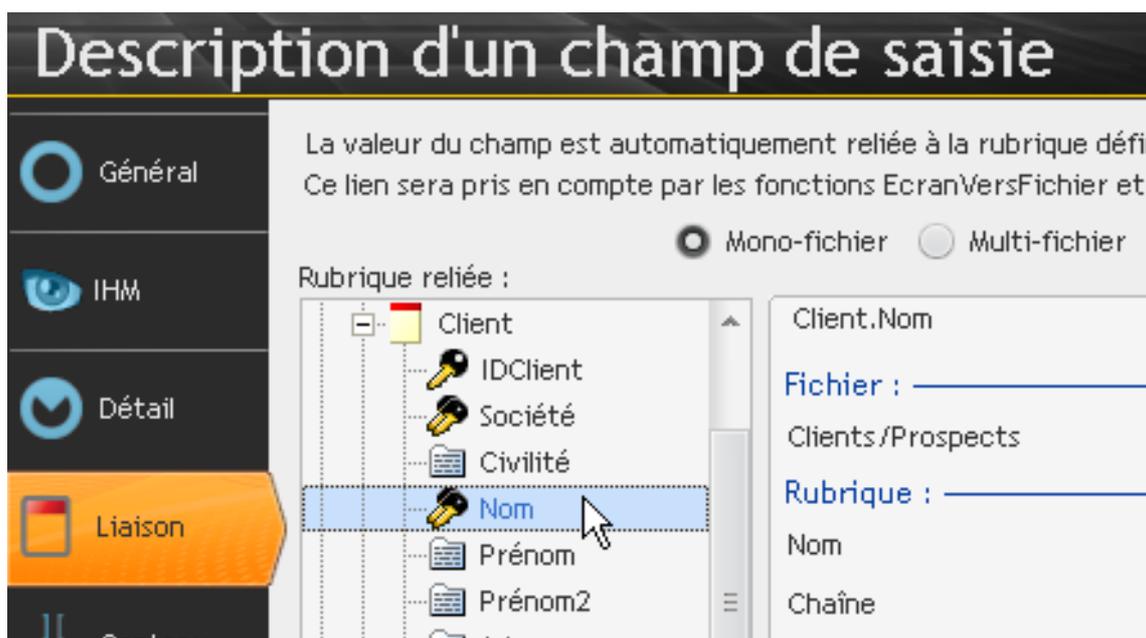
Pour que le code qui se trouve dans la partie suivante fonctionne correctement, il faut que vos champs de saisie soient liés aux éléments de votre analyse, c'est à dire associés aux attributs voulus de la table.

Ainsi au lieu d'ajouter un champ de saisie standard sans liaison, allez dans "Description" :



*Champ de saisie : description*

Puis dans "Liaison", relier le champ de saisie à l'attribut de la table voulu :

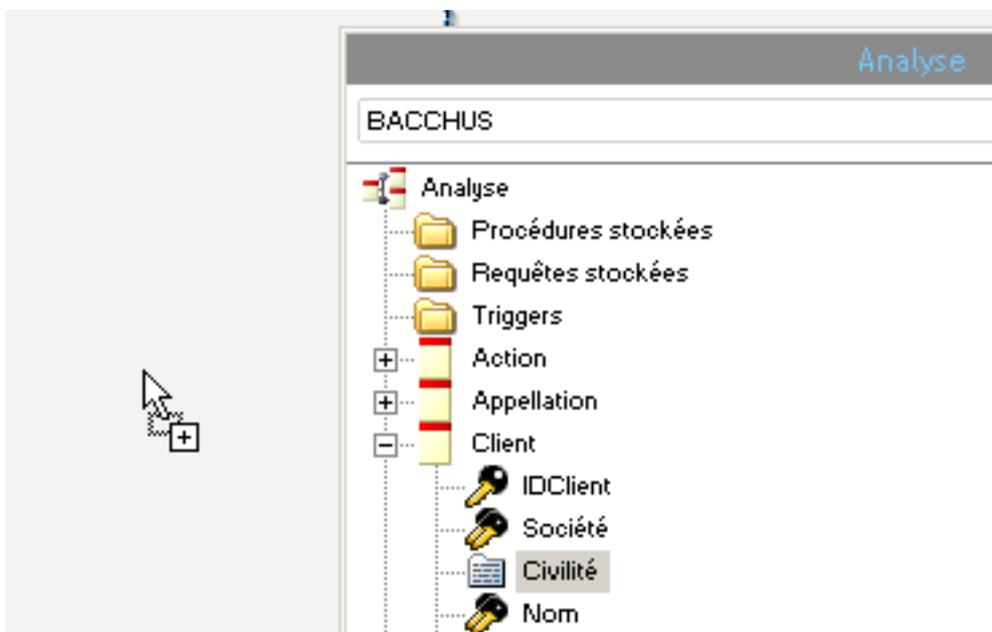


Champ de saisie : lien vers l'attribut de la table



### Méthode : Faites un glisser-déposer depuis l'analyse

Vous pouvez aussi faire un glisser-déposer depuis l'analyse pour créer automatiquement un champ de saisie "relié" :



Champ de saisie : glisser-déposer depuis analyse



### Conseil : Optimisation du code

Le schéma du fonctionnement général via un buffer intermédiaire a plusieurs impacts dont certains très pratiques.

Prenons le code suivant :

```
//SAI_NOM et SAI_SOC sont des champs de saisie
CLIENT.NOM = SAI_NOM
CLIENT.SOCIETE = SAI_SOC
HAJOUTE (CLIENT)
```

Si on crée les 2 champs de saisie en indiquant qu'ils sont liés au nom et à la société de la table CLIENT. On peut remplacer ce code par :

```
PageVersFichier()
HAjoute (CLIENT)
```

De même, on peut remplacer un code qui fait l'inverse, afficher la valeur des enregistrements dans des champs de saisie, comme :

```
SAI_NOM = CLIENT.NOM
SAI_PRENOM = CLIENT.PRENOM
SAI_SOC = CLIENT.SOC
SAI_ADRESSE = CLIENT.ADRESSE
SAI_VILLE = CLIENT.VILLE
SAI_TEL = CLIENT.TEL
```

Par :

```
FichierVersPage()
```



### *Conseil : Même action sur 2 objets : fonction ExécuteTraitement*

Si une action doit être effectuée sur 2 objets différents dans la même page ou que vous souhaitez lancer du code serveur depuis un objet du navigateur, vous pouvez utiliser la fonction ExécuteTraitement :

#### **ExécuteTraitement(Objet, trtClic)**

Le 2ème paramètre est le type d'action devant être "émulée", cela peut être un simple clic (trtClic), un double clic (trtDoubleClic), une procédure, etc.



### *Conseil : Exécuter du code serveur depuis du code navigateur*

Beaucoup de fonctions sont accessibles dans la partie serveur de WebDev mais pas dans la partie navigateur. Pour pallier à ce problème, vous pouvez "forcer" du code serveur depuis du code navigateur. Pour ce faire, l'astuce consiste à :

- Créer un bouton "caché" (invisible)
- Y mettre le code voulu dans la partie serveur (partie jaune)
- Puis dans le code navigateur, utilisez la fonction ExécuteTraitement :  
ExécuteTraitement(nomBoutonCache, trtClic)

## **1. Raccourcis**

- CTRL + TAB : permet de basculer entre les pages ou passer du code à la page
- CTRL + W : pour afficher / fermer les barres d'outils
- CTRL + / : permet de mettre toute la zone sélectionnée en commentaire
- CTRL + SHIFT + / : permet de décommenter toute la zone sélectionnée
- CTRL + SHIFT + V : Copier/Coller historisé

- CTRL + D : pour dupliquer la sélection
- CTRL + E : pour rechercher un élément dans le projet
- ALT + flèche : pour déplacer le code sélectionné
- F1 : accès à l'aide contextuelle
- F5 : affiche l'ordre de navigation
- F6 : affiche le nom des champs
- F7 : affiche le contour des champs
- F12 : options d'édition



### Remarque

F12 permet aussi le débogage dans un navigateur (type firefox ou chrome)

## C. Interface : prise en main

### 1. Code des champs

Partie verte : code exécuté côté serveur

Partie jaune : code exécuté côté navigateur



### Complément : Pour passer à javascript

Lorsqu'on clique sur "WL" (partie verte), on passe du WLangage au Javascript. Cela permet donc de développer ses propres fonctions javascript dans la page.

Il faut savoir que, in fine, le code WLangage sera transformé en javascript par WebDev.

### 2. Ordre d'exécution

1. Navigateur
2. Page envoyée au serveur
3. Code serveur
4. Page renvoyée au navigateur (page actualisée)

### 3. Exemple de code

#### Ajout d'un enregistrement

```
//On vide les buffers fichiers
//Pour éviter d'être pollué par d'anciens enregistrements potentiels
//qui serait dans les buffers
HRAZ(Client)
HRAZ(Action)
//On remplit les buffers fichiers
//Permet de récupérer les informations saisies
//dans les champs de saisie de la page
PageVersFichiers()
//Ajout dans le fichier ou table Client
```

```
Hajoute(Client)
//Ajout dans le fichier Action
Action.idClient = Client.idclient
HAjoute(Action)
//Fin de l'enregistrement
Info("Vous êtes maintenant enregistré")
PageAffiche(PAGE_Menu)
```

### Utilisation des transactions

Le code précédent peut être amélioré en créant une transaction.

Pour ce faire utilisez, avant un ajout ou toute modification sur la base, **HTransactionDebut()**

Pour valider la transaction, utilisez **HTransactionFin()**.

Pour annuler une transaction, utilisez **HTransactionAnnule()**.

### Logout

```
//On affiche la page de fin
//Et on ferme le programme
PageAffiche(pageFin)
FinProgramme()
```

### Reconnexion à un site

```
//Lance un site dynamique WebDev à partir d'une page
//Le site dynamique doit se trouver sur le même serveur Web
SiteDynamiqueAffiche("MonSite")
```

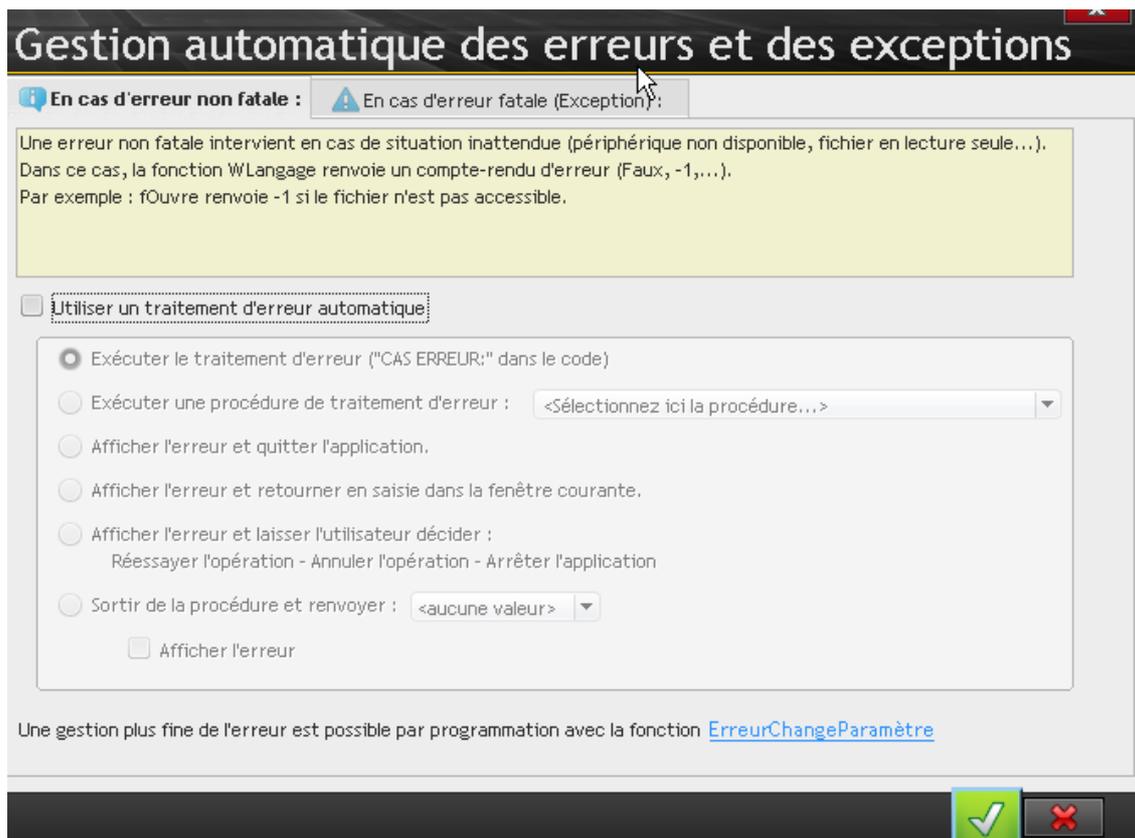
## a) Gestion des erreurs et exceptions

Pour gérer une erreur ou une exception dans votre code, il faut cliquer sur la partie "Si Erreur : par programme" et/ou "Quand Exception : par programme".

Clic de <b>BTN_OK</b> (serveur)* <b>AAA</b> Si le nom de la configuration est vide on en fait rien <b>Taille(\$SAI_Configuration) = 0</b> ALORS // Simule un appel au bouton annuler	<b>Si Erreur : par programme</b> <b>Quand Exception : par programme</b> En cas d'erreur : vous avez choisi que votre programme doit gérer cette erreur cliquez ici pour que l'erreur soit gérée automatiquement
---	---

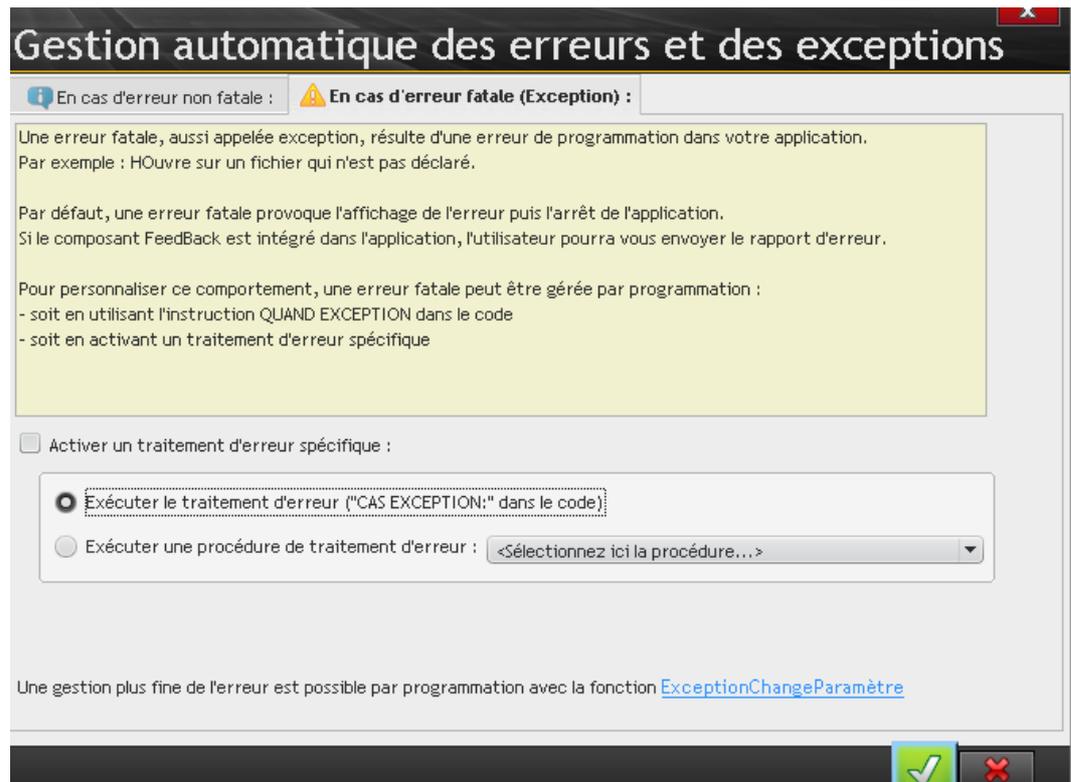
### *WebDev : Erreur et Exception*

Vous pourrez alors indiquer plusieurs "stratégies" comme déclencher une procédure de traitement, afficher un message à l'utilisateur, etc.



### WebDev : Gestion des erreurs

En cas d'exception, étant bloquant par définition, les options sont moins fournies soit vous utilisez du code de traitement via "CAS EXCEPTION :" soit vous déclenchez le code d'une procédure.



WebDev : Gestion des exceptions



### Remarque : La gestion des erreurs ou exception s'imbrique

Il faut savoir qu'une erreur non traitée par le bloc ou la partie l'ayant déclenchée sera alors traitée au niveau plus haut et ainsi de suite.

**Ainsi vous pouvez implémenter une gestion des erreurs au niveau du code du projet pour être sûr qu'elles soient traitées.**



### Conseil : Pensez à écrire un traitement

Il est évident que des erreurs surviendront (un programme qui fonctionne est un programme bogué qui s'ignore), pensez donc à récupérer les erreurs et exceptions pour les traiter.

Voici un exemple de code accessible dans le code du projet de Arts et culture :

```
CAS EXCEPTION:
HFermeConnexion (gcnxBDD)
HTransactionAnnule ()
Erreur ("Une erreur fatale a été détectée. Merci de
réessayer ultérieurement. Si le problème persiste, veuillez
écrire à dsi.dsi-da@ac-toulouse.fr",
ExceptionInfo (errComplet), "", "Arrêt de la procédure.")
SI gbModeDebug ALORS
fichierInfoErreur est une chaîne =
ComplèteRep (fRepDonnées ())+"Exception_"+DateHeureSys ()
+".txt"
fSauveTexte (fichierInfoErreur, ExceptionInfo (errComplet))
fichierDump est une chaîne = ComplèteRep (fRepDonnées ())
+"Exception_Dump_"+DateHeureSys ()+".wdump"
dbgSauveDumpDébogage (fichierDump)
```

```
fichiersJointes est un tableau de chaînes =
[fichierInfoErreur,fichierDump, gsFichierAudit]
EnvoiEmailErreur(ExceptionInfo(errComplet),fichiersJointes)
FIN
```

Une procédure de gestion des erreurs simple déclenché par le projet en cas d'erreur permet d'annuler la transaction et de prévenir l'utilisateur qu'une erreur s'est produite :

```
PROCEDURE GestionErreur()
HTransactionAnnule()
Info("Une erreur est survenue. Merci de réessayer
ultérieurement. Si le problème persiste, veuillez écrire à
dsi.dsi-da@ac-toulouse.fr", ExceptionInfo(errComplet), "",
"Arrêt de la procédure.")
```

## b) Contrôle des éléments

Si vous êtes familiarisé avec les termes "failles XSS", vous savez alors déjà à quel point il est important de bien contrôler vos champs de saisie voir même tous les champs.

Pour ce faire, **ne faites pas une simple vérification via du code javascript des éléments, faites là aussi et surtout avant l'insertion en base.**

Une vérification côté navigateur est une bonne idée car elle "décharge" le serveur, est traitée rapidement par le navigateur Web côté client. Cependant, une personne mal intentionnée pourrait avoir désactivé javascript et vous voilà à la portée de l'insertion de n'importe quoi dans votre base.

**Faites donc systématiquement une vérification côté navigateur et côté serveur.**

### *Exemple de vérification simple côté navigateur*

```
SI ChaîneOccurrence(SAI_Mail..Valeur,"@ac-toulouse.fr")<>1
ALORS
POPUP_Erreur.LIBHTM_PopupErreur..Valeur="L'email du
coordinateur est obligatoire et doit être celui académique
(en @ac-toulouse.fr)."
```

PopupAffiche(POPUP\_Erreur)

RETOUR

FIN

```
SI COMBO_Domaine..Valeur="" _OU_
COMBO_Dispositif..Valeur="" _OU_ SAI_Intitule..Valeur=""
_OU_ SAI_Mail..Valeur="" ALORS
POPUP_Erreur.LIBHTM_PopupErreur..Valeur="Toutes les données
sont obligatoires. Merci de sélectionner un établissement,
de saisir un intitulé, un dispositif, un domaine et une
adresse courriel valide."
```

PopupAffiche(POPUP\_Erreur)

RETOUR

FIN

### *Exemple de vérification simple côté serveur*

```
SI COMBO_Domaine..Valeur="" _OU_
COMBO_Dispositif..Valeur="" _OU_ SAI_Intitule..Valeur=""
```

```

_OU_SAI_Mail..Valeur="" _OU_
ChaîneOccurrence(SAI_Mail..Valeur, "@ac-toulouse.fr") <>1
ALORS
Info("Toutes les données sont obligatoires. Merci de
sélectionner un établissement, de saisir un intitulé, un
dispositif, un domaine et une adresse courriel en ac-
toulouse.fr valide.")
RETOUR
FIN

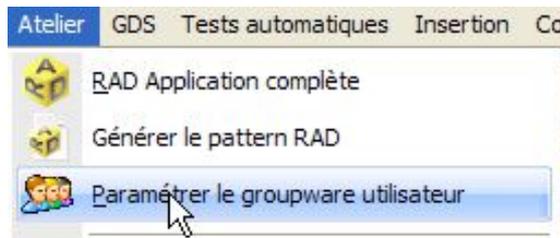
```

### c) Gestion des droits : exemple du groupware

Le groupware est un composant fourni en standard pour WebDev. Il permet de réaliser une gestion des droits. De base, il permet de se connecter à un annuaire Active Directory et embarque une base hyperfile. Pour nos besoins, il a fallu le modifier un peu pour se connecter à un annuaire LDAP et une base MySQL.

Ce composant est riche en fonctionnalités mais compliqué à mettre en oeuvre si on sort un peu du "cadre" Active Directory + HyperfileSQL.

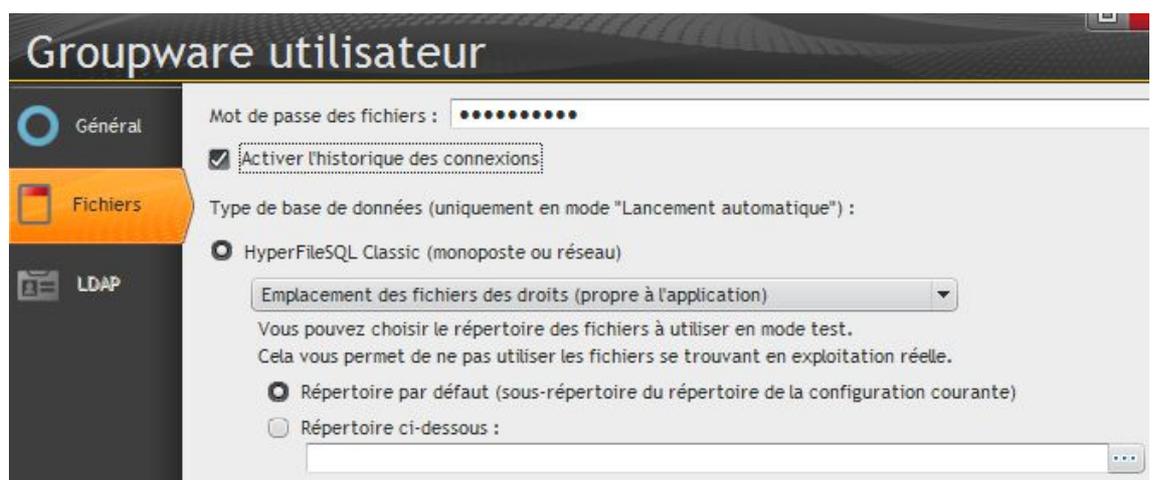
Pour utiliser le groupware WebDev, il faut aller dans "Atelier" -> "Paramétrer le groupware utilisateur" :



WebDev : Paramétrage du groupware

### i Création des éléments de base

Le groupware fonctionne nativement si vous utiliser HyperFile et Active Directory. Par contre si vous voulez utiliser une base externe (MySQL ou autre) et que vous n'avez qu'un LDAP, il va vous falloir changer les options par défaut et coder un peu... D'abord modifier vers "intégration personnalisable" :



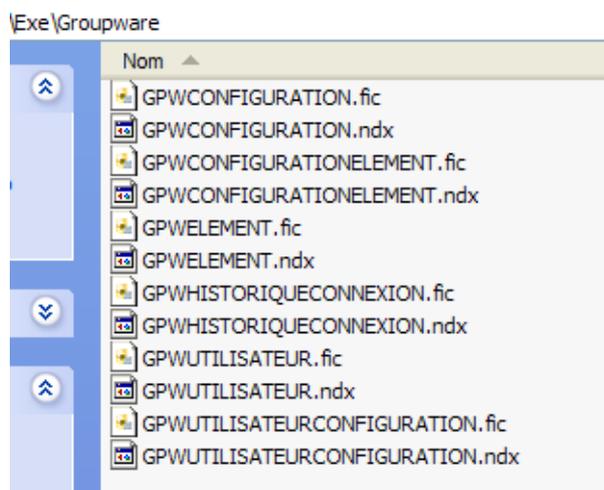
WebDev : Groupware - génération et création des fichiers

Le but est de récupérer le SQL des tables qui va être automatiquement créé par WebDev.

Si vous exécutez votre projet, vous allez voir apparaître les fichiers .fic dans le sous-répertoire Groupware du répertoire Exe du projet :

**C:\Mes Sites\VOTREPROJET\Exe\Groupware**

Si vous ne les trouvez pas, ils peuvent aussi apparaître dans le répertoire Gpw\_nomprojet.



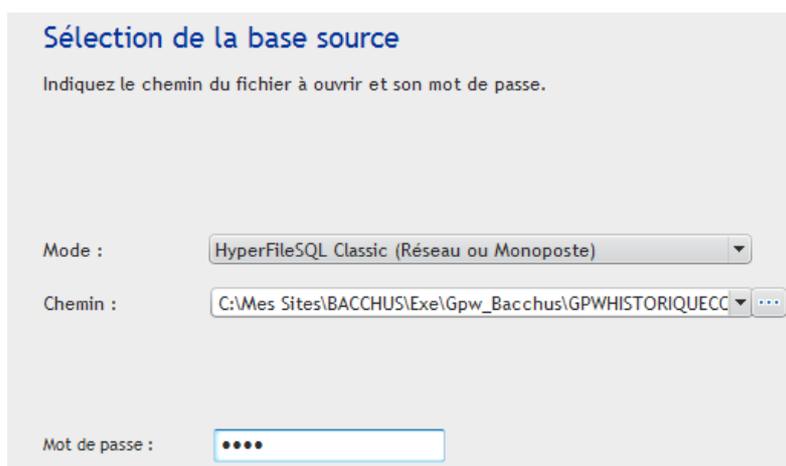
WebDev : Fichiers du groupware



### Attention

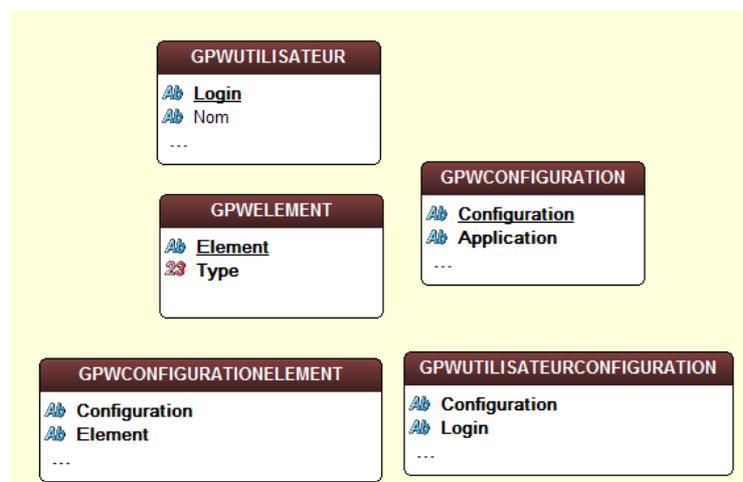
Souvenez vous du mot de passe des fichiers car vous en aurez besoin par la suite

Nous allons donc récupérer ces fichiers pour les intégrer dans l'analyse de votre projet. Pour ce faire, vous ouvrez votre analyse et vous faites un glisser/déposer des fichiers .fic dans l'analyse. Un assistant se lance et vous demande notamment d'indiquer le mot de passe (celui rentré précédemment lors de l'activation du groupware) :



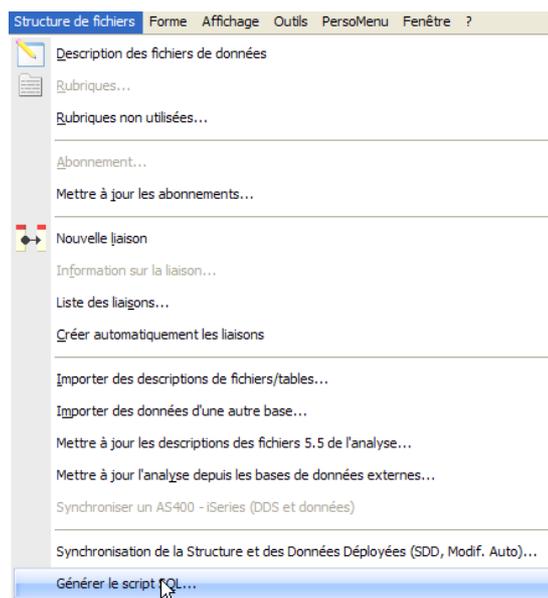
WebDev : Import des tables du groupware

Vous importez tous les fichiers pour arriver à ça :



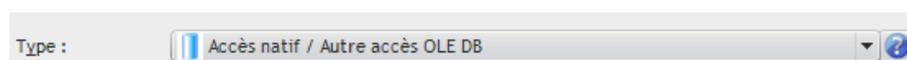
### WebDev : Analyse groupware

Vous pouvez alors exporter le SQL pour créer les tables dans une base externe en allant dans "Structure de fichiers" -> "Générer le script SQL" puis vous suivez l'assistant pour le générer pour la base de destination voulue (ici MySQL) en sélectionnant les tables du groupware (qui commencent par GPW).



### WebDev : Génération script SQL du groupware

Puis modifier les tables pour indiquer que ce sont des tables provenant d'une source extérieure en faisant un clic droit sur chaque table puis "Description du fichier de données" et enfin modifier le type :



### WebDev : Modification du type de BDD

Et indiquer la connexion associée à cet accès.



### Complément

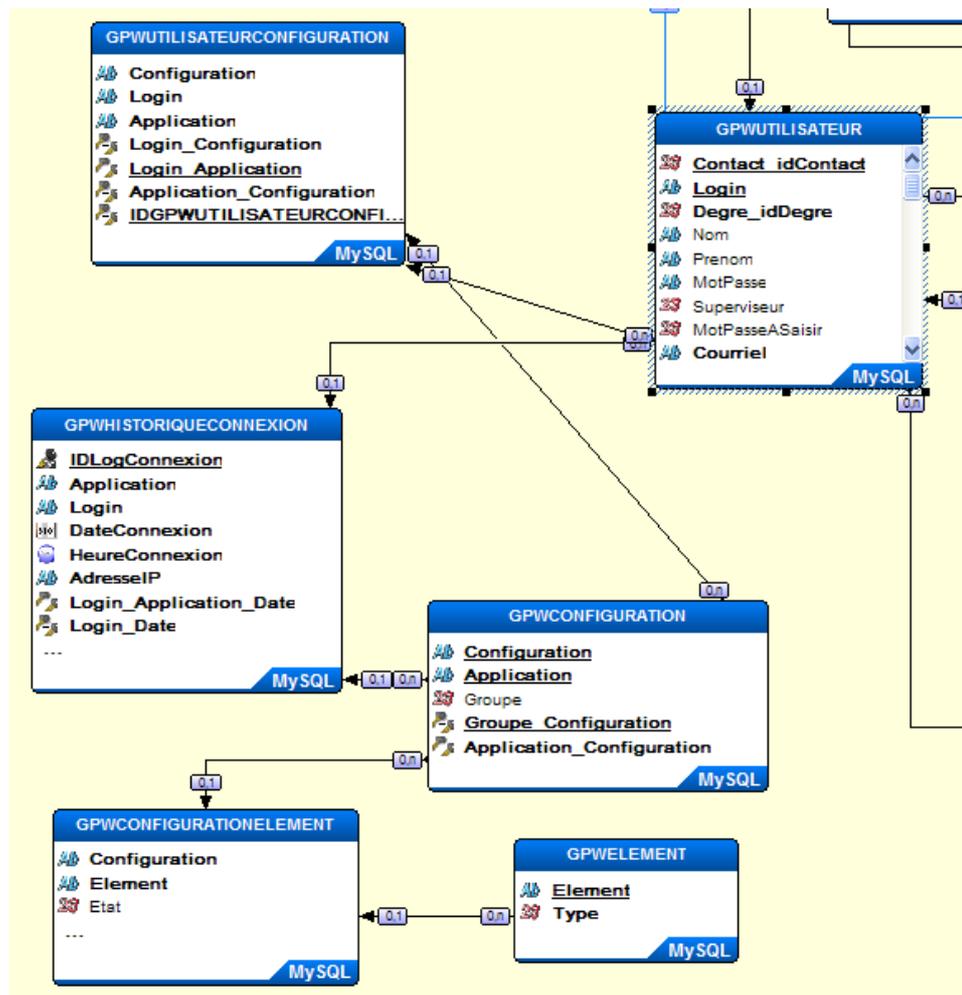
Pour un accès mySQL il faut avoir installé au préalable le driver mysql pour

WebDev, disponible sur <http://www.pcsoft.fr/st/telec/modules-communs-17/mysql.htm> (pour la version 17) ou <http://www.pcsoft.fr/st/telec/modules-communs-18/mysql.htm> (pour la version 18)

Vous remarquez que, par défaut, il n'y a aucune liaison entre les tables (la magie de WebDev). Je vous conseille donc de les créer. Pour les créer, il suffit de savoir que :

- Chaque "GPWUTILISATEUR" peut avoir plusieurs "GPWUTILISATEURSCONFIGURATIONS" et que la clef unique du fichier est "Login"
- Chaque "GPWUTILISATEUR" peut avoir plusieurs "GPWHISTORIQUECONNEXION" et que la clef unique du fichier est "Login"
- Chaque "GPWCONFIGURATION" peut avoir plusieurs "GPWUTILISATEURSCONFIGURATIONS" et que la clef unique du fichier est "Configuration"
- Chaque "GPWCONFIGURATION" peut avoir plusieurs "GPWCONFIGURATIONELEMENTS" et que la clef unique du fichier est "Configuration"
- Chaque "GPWELEMENT" peut avoir plusieurs "GPWCONFIGURATIONELEMENTS" et que la clef unique du fichier est "Element"

Ce qui donne :



*WebDev : Liaison entre les tables du groupware*

Vous pouvez alors relier les tables aux autres éléments de votre analyse. **De cette façon, le groupware est réellement lié au reste de votre analyse.**

## ii Codage des accès

Pour lancer la connexion au groupware, vous pouvez créer un formulaire pour vous connecter (avec un libellé du type "Administration du site").

Ensuite dans le code du bouton, nous allons coder les accès à la base de données :

```

SI PAS
gpwOuvreConnexion (Dev_BDD,gsUtilBDD,gsMdpBDD,gsIPBDD,gsNomBDD,hAccèsNatifMySQL) ALORS
Erreur(ErreurInfo())
SINON
nRes est un entier = gpwOuvre()
SI nRes <> gpwOk ALORS
SELON nRes
CAS gpwUtilisateurInconnu : Erreur("Utilisateur inconnu.")
CAS gpwMotPasseInvalide : Erreur("Mot de passe invalide.")
CAS gpwErreur :
FIN
SINON

```

```

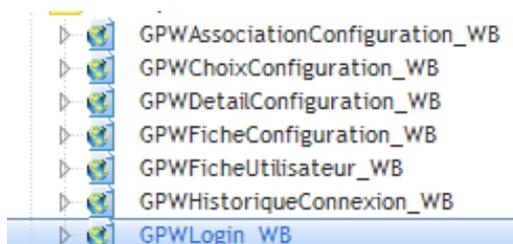
HChangeConnexion ("*", Dev_BDD)
gsProfil = gpwRecupInfoUtilisateur (gpwInfoGroupe)
gsLogin = gpwRecupInfoUtilisateur (gpwInfoLogin)
HLitRecherchePremier (GPWUTILISATEUR, Login, LoginEntree)
gsMail_utilisateur=GPWUTILISATEUR.Courriel
prenom_=GPWUTILISATEUR.Prenom
nom_=GPWUTILISATEUR.Nom
PageAffiche (PAGE_Accueil)
FIN
FIN

```

La procédure importante est **gpwOuvreConnexion**, elle va démarrer tout le groupware et notamment les popup de connexion de l'utilisateur.

### iii Modification des pages du groupware pour utiliser un LDAP

Le groupware a généré plusieurs pages accessibles dans votre projet :



*WebDev : Pages du groupware*

GPWLogin\_WB est la première popup, elle contient un formulaire demandant le nom d'utilisateur et le mot de passe.

#### Quelques explications

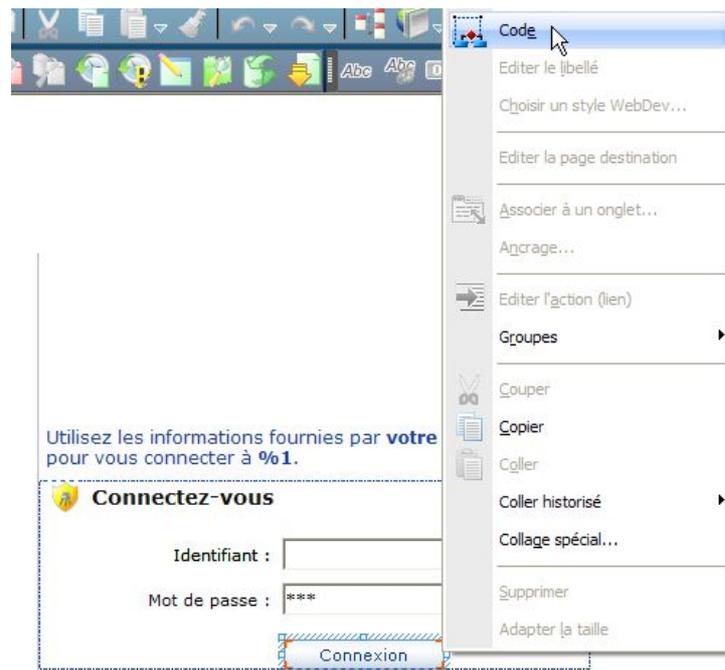
Le groupware ne peut pas se connecter directement à un ldap. Il va se connecter à la BDD et juste vérifier le Login et le mot de passe (attribut MotPasse de la table GPWUTILISATEUR). Or, nous ne connaissons pas le mot de passe LDAP de l'utilisateur et cela constituerait une faille de sécurité de le stocker dans une base...

La solution consiste à mettre un mot de passe "bidon" c'est à dire que pour chaque création d'utilisateur dans le groupware (ayant donc accès à l'application), nous allons créer le profil avec le mot de passe "ldap". Celui-ci n'est utilisé que pour "satisfaire" le groupware. En effet, lorsqu'un utilisateur se connecte, on vérifie d'abord que la connexion au ldap est ok puis on le connecte au groupware avec le code :

```
gpwLogin(SAI_Login,"ldap")
```

Cette connexion au groupware n'est effectuée que si la connexion LDAP est ok (nom utilisateur et mot de passe correct).

On va donc regarder le code du bouton connexion de la popup GPWLogin\_WB :



WebDev : Groupware - code pour la connexion

### Code à insérer pour la connexion

```

SI SAI_Login="" _OU_ SAI_MotPasse="" ALORS
Erreur("Veuillez renseigner le nom d'utilisateur et le mot
de passe!")
RETOUR
FIN
// Si on a déjà passé la première étape
dbgAssertion(Taille(gsLoginSuperviseur) = 0)
SI Taille(gsLoginSuperviseur) > 0 ALORS RETOUR
// Initialisation de l'état qui sera calculé lors de
l'initialisation du login
gnValeurRenvoyee = gpwOk
// Par défaut on ne réaffiche pas la page ni ne force un
focus
gnRepriseSaisie = cRepriseSaisieSans
// Si le login saisie est celui du superviseur (en cas de
problème LDAP ou autre)
SI SAI_Login=gsNomUtilSuperviseur ALORS
// Vérification du Login
VerifLogin()
FIN
// Si on doit redonner le focus a un champ c'est que l'on
doit réafficher la page courante
SI gnRepriseSaisie <> cRepriseSaisieSans ALORS
PageActualise()
RETOUR
FIN
// LE CODE CI-DESSOUS NE FONCTIONNE PAS
// LE GROUPWARE NE FONCTIONNE QU'AVEC UN AD
// Permet d'indiquer le gpwEnModeLDAP à Vrai
// Ne fonctionne pas car cette fonction envoie les uid à la

```

```

mode AD (login@domaine.tld)
// gpwOuvreSessionLDAP("192.168.0.1",
389,3,Vrai,"ou=racine,c=fr")
// Ecrit la valeur dans la variable du groupeware
SI gnValeurRenvoyee = gpwOk ALORS
bConnexionOK est un booléen = Faux
//Si l'utilisateur n'est pas le superviseur -> on effectue
une connexion LDAP
SI GPWUTILISATEUR.Login<>gsNomUtilSuperviseur ALORS
//Connexion via ldap
bConnexionOK = AuthentificationLDAP(SAI_Login,SAI_MotPasse)
SI bConnexionOK = Vrai ALORS
gnValeurRenvoyee = gpwLogin(SAI_Login,"ldap")
SINON
gnValeurRenvoyee = gpwMotPasseInvalide
FIN
SINON
// Connexion normale en vérifiant le nom util/mot de passe
dans la BDD
gnValeurRenvoyee = gpwLogin(SAI_Login, SAI_MotPasse)
FIN
SI gnValeurRenvoyee<>gpwOk ALORS
Erreur("L'authentification a échoué."+RC+RC+ErreurInfo())
gnRepriseSaisie = cRepriseSaisieLoginLDAP
PageActualise()
RETOUR
FIN
// si l'utilisateur est superviseur
SI gpwRecupInfoUtilisateur(gpwInfoSuperviseur) _OU_
ChaîneOccurrence(gpwRecupInfoUtilisateur(gpwInfoGroupe),"DA
EPS")>0 ALORS
// Affiche la cellule spécialisée
CELL_Superviseur..Visible = Vrai
// Grise la cellule principale
GR_Login..Etat = Grisé
// Mémorise le nom de l'utilisateur
gsLoginSuperviseur = SAI_Login
LIB_Superviseur..Valeur =
ChaîneConstruit(LIB_Superviseur..Valeur,
gsLoginSuperviseur)
// Et donne le focus au bon bouton
gnRepriseSaisie = cRepriseSaisieOkSuperviseur
RETOUR
FIN
FIN
// Ferme la page en indiquant que le login a réussi
FermePageLogin()

```

### Code de la connexion LDAP

```

PROCEDURE AuthentificationLDAP(login_ est une chaîne, mdp_
est une chaîne)
// Initialisation de la structure
LDAPRAZ()
LDAPSession.Hôte = gsNomLDAP
LDAPSession.Port = portLDAP
LDAPSession.Version = versionLDAP
LDAPSession.Utilisateur = ""

```

```

LDAPSession.MotDePasse = ""
// Connexion
LDAPConnecte("LdapCherche")
// Recherche des informations de l'utilisateur.
sDn est une chaîne =
LDAPRecherche("LdapCherche","ou=people,ou=racine,c=fr","uid
="+login_)
// Dn vide
SI sDn="" ALORS
Erreur("Utilisateur introuvable dans l'annuaire")
// Dn trouvé, réinitialisation de la structure
REVOYER Faux
SINON
LDAPDéconnecte("LdapCherche")
LDAPSession.Utilisateur = sDn
LDAPSession.MotDePasse = mdp_
// Validation du mot de passe de l'utilisateur
SI LDAPConnecte("LdapAuth") ALORS
Email est une chaîne =
UTF8VersChaîne(LDAPValeur("LdapAuth", sDn, "Mail",
1),alphabetAnsi)
//Info("Email de cet utilisateur : " + Email)
LDAPDéconnecte("LdapAuth")
REVOYER Vrai
SINON
Erreur("Mot de passe incorrect")
REVOYER Faux
FIN
FIN
Modification de la création d'une fiche utilisateur
Vous devez interfacier la création d'un utilisateur avec
votre LDAP. Dans GPWFicheUtilisateur_WB, voici le code du
bouton "Ok" :
// Si le nom de la configuration est vide on en fait rien
SI Taille(Login) = 0 ALORS
// Simule un appel au bouton annuler
ExécuteTraitement(BTN_Annuler, trtClic)
RETOUR
FIN
SI Login<>gsNomUtilSuperviseur ALORS
// Initialisation de la structure
LDAPRAZ()
LDAPSession.Hôte = gsNomLDAP
LDAPSession.Port = portLDAP
LDAPSession.Version = versionLDAP
LDAPSession.Utilisateur = ""
LDAPSession.MotDePasse = ""
// Connexion au ldap
LDAPConnecte("LdapCherche")
sDn est une chaîne
// Recherche des informations de l'utilisateur.
SI ChaîneOccurrence(Login..Valeur, "@domaine.fr")<>1 ALORS
sDn =
LDAPRecherche("LdapCherche",gsDnBaseLDAP,"uid="+Login)
SINON
sDn =
LDAPRecherche("LdapCherche",gsDnBaseLDAP,"mail="+Login)
FIN
SI sDn="" ALORS

```

```

Erreur("Utilisateur introuvable dans l'annuaire")
LDAPDéconnecte("LdapCherche")
RETOUR
// Dn trouvé, réinitialisation de la structure
SINON
LDAPDéconnecte("LdapCherche")
LDAPSession.Utilisateur = sDn
SI LDAPConnecte("LdapInfo") ALORS
Type = UTF8VersChaîne(LDAPValeur("LdapInfo", sDn,
"title"),alphabetAnsi)
Login = UTF8VersChaîne(LDAPValeur("LdapInfo", sDn,
"uid"),alphabetAnsi)
Courriel = UTF8VersChaîne(LDAPValeur("LdapInfo", sDn,
"mail"),alphabetAnsi)
Prénom = UTF8VersChaîne(LDAPValeur("LdapInfo", sDn,
"givenName"),alphabetAnsi)
MotPasse = "ldap"
//Info("Email de cet utilisateur : " + Courriel)
LDAPDéconnecte("LdapInfo")
SINON
Erreur("Connexion au LDAP")
LDAPDéconnecte("LdapInfo")
RETOUR
FIN
FIN
FIN
//GPWUtilisateur.Login = Login
//Info(Login)
// Transfer de la valeur des champs dans les variables du
fichier
PageVersFichier()
// Nouvel utilisateur
SI bNouveau ALORS
// Ajoute l'enregistrement dans le fichier
// Association de la configuration par défaut à
l'utilisateur
HRAZ (GPWUTILISATEURCONFIGURATION)
GPWUTILISATEURCONFIGURATION.Login=Login
GPWUTILISATEURCONFIGURATION.Configuration=gpwAucun
GPWUTILISATEURCONFIGURATION.Application=ProjetInfo()
// Ajout de l'enregistrement
SI PAS HAjoute(GPWUTILISATEURCONFIGURATION) ALORS
Erreur("Impossible de fixer la configuration par défaut à
l'utilisateur."+RC+HErreurInfo())
FIN
FIN
FIN
// Modification de l'utilisateur
SINON
// Modifie l'enregistrement dans le fichier
SI PAS HModifie(GPWUTILISATEUR) ALORS
Erreur("Impossible de modifier
l'utilisateur."+RC+HErreurInfo())
FIN
FIN
// Réaffiche la page qui a ouvert la page courante avec un
rafraichissement de la table
GPWAssociationConfiguration_WB.AfficheTable()
PageActualise (GPWAssociationConfiguration_WB)

```

```
// Ferme notre contexte pour liberer de la mémoire
ContexteFerme ("")
```

Voilà, ce code vous permet de récupérer les informations utilisateur depuis votre serveur LDAP. Vous pouvez donc les "griser" sur la page de création de l'utilisateur. Ce code va donc créer les informations nécessaires dans la table utilisateur du groupware pour que vos utilisateurs puissent se connecter.

#### iv Configuration "fonctionnelle" du groupware et des droits utilisateurs

Il vous reste à configurer votre groupware pour accorder l'accès à certains éléments ou autre en fonction du profil. Pour ce faire, vous devez vous connecter en tant que superviseur. Le mot de passe par défaut du Login superviseur est... superviseur. Je vous conseille de le modifier. **Vous aurez alors accès au paramétrage des droits (pour cacher des éléments du menu, cacher des actions, des formulaires, etc).**

**Attention : sachez que le groupware ne peut pas rendre visible des champs invisibles par défaut.** Il fait exactement l'inverse, il cache des éléments à certains profils ou groupes d'utilisateur. Pensez à donc à appliquer les restrictions sur le modèle de page pour qu'elles s'appliquent à l'ensemble du site (pour cacher des éléments du menu, etc).

Vous pouvez aussi créer des restrictions par codage. En utilisant, par exemple, le code :

```
gsProfil = gpwRecupInfoUtilisateur(gpwInfoGroupe)
Si gsProfil<>"superviseur" alors
//On applique des restrictions
BTN_Admin..Visible=Faux
FIN
```

## 4. Utilisation des groupes

Pour associer des éléments et, ainsi, simplifier le code, on peut les "grouper".

Cela va vous permettre de simplifier le code en n'écrivant qu'une seule instruction pour, par exemple, rendre tous les éléments d'un groupe invisible.

Pour ce faire, sélectionnez l'ensemble des éléments voulus puis allez dans "Page" -> "Groupes" -> "Associer la sélection".



WebDev : Création d'un groupe d'objets

Cliquez ensuite sur "Nouveau" et donner un nom au groupe.



WebDev : Création d'un groupe d'objets

Cela vous permettra ensuite dans le code d'utiliser le nom du groupe pour effectuer une action groupée.

Par exemple :

```
GR_GroupeDeTest..Visible = Faux
```

## 5. Tables

Une table permet d'afficher des données automatiquement (basé sur un fichier ou une requête) ou par programmation.

Un glisser/déposer de la liste de vos fichiers va vous permettre d'insérer une table affichant les attributs (ou rubriques) que vous aurez sélectionnés.

```
TableAjouteLigne (nomTable, valCol1, valCol2, ...)
TableAjoute (nOmtable, valCol1+TAB+valCol2+TAB+...) // -> TAB
étant un séparateur de colonne sous WebDev
TableSupprimeTout (nomTable)
nbLignes = TableOccurrence (nomTable)
nbLignes = nomTable..Occurrence // -> aussi supporté en
code navigateur
TableTrie (nomTable, "+nomCol") // -> Tri croissant
TableTrie (nomTable, "-nomCol") // -> Tri décroissant
TableTrie (nomTable, "-nomCol1", "+nomCol2") // -> Tri
multiple
TableSelect (nomTable) // -> pour récupérer la ligne
sélectionnée
```

## 6. Zones répétées

Les zones répétées permettent d'afficher un ensemble de champs par enregistrement.

Elles peuvent être horizontales ou verticales et se prêtent bien à l'affiche de produits par exemple, la mise en page étant plus ergonomique que celle d'un tableau.



### Attention

On ne doit pas utiliser directement le nom des champs en programmation, on doit

d'abord associer un champ d'affichage formaté qu'on associe à la zone répétée via la création d'un nouvel attribut. Cet attribut est ensuite associé aux attributs, colonnes de la table voulue.

Pensez à bien ajouter des attributs à la zone répétée que vous associez au champ de la table sur lequel il s'applique.

## Description d'une zone répétée

Nom zone :	ZR_REQ_Intervenant_Intervention_Projet	Nouveau	Insérer	Supprimer	
Attribut :	ATT_Periodicite	ATT_NbHeurePeriode	ATT_NbHeureGlobal	ATT_Observation	ATT_Prenom
Champ :	LIBF_Periodicite	LIBF_NbHeurePeriode	LIBF_NbHeureGlobal	LIBF_Observation	LIBF_Prenom
Propriété :	Valeur	Valeur	Valeur	Valeur	Valeur

WebDev : Association des attributs aux champs d'une zone répétée

## Description d'une zone répétée

Nom zone :	ZR_REQ_Intervenant_Intervention_Projet	Nouveau	Insérer	Supp	
Attribut :	ATT_Periodicite	ATT_NbHeurePeriode	ATT_NbHeureGlobal	ATT_Observation	ATT_Prenom
Champ :	LIBF_Periodicite	LIBF_NbHeurePeriode	LIBF_NbHeureGlobal	LIBF_Observation	LIBF_Prenom
Propriété :	Valeur	Valeur	Valeur	Valeur	Valeur

- Général
- IHM
- Contenu
- Liaison
- Style

La valeur du champ est automatiquement reliée à la rubrique définie ci-dessous.  
Ce lien sera pris en compte par les fonctions EcranVersFichier et FichierVersEcran.

Rubrique reliée :  Mono-fichier  Multi-fichier

- REQ\_Dispositif\_2d\_simplifie
- REQ\_Dispositif\_Degré
- REQ\_Domaine\_degré
- REQ\_Domaine\_Dispositif
- REQ\_Domaine\_Sur\_Dispositif
- REQ\_Domaine\_Sur\_LibDispositif\_Degré
- REQ\_Domaines\_de\_compétence
- REQ\_Etablissements
- REQ\_Etablissements\_Departement
- REQ\_Etablissements\_Sigle
- REQ\_Etat\_Drac1
- REQ\_IEN00\_HG
- REQ\_Intervenant\_Intervention\_Projet
  - periodicite
  - nbHeurePeriode

REQ\_Intervenant\_Intervention\_Projet.nbHeu

Fichier : \_\_\_\_\_

REQ\_Intervenant\_Intervention\_Projet

Rubrique : \_\_\_\_\_

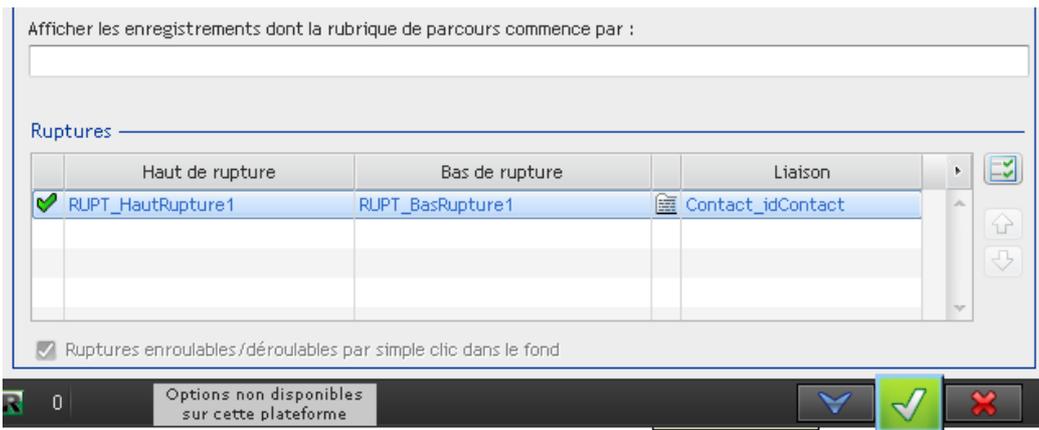
Entier sur 4 octets      Taille : 4 octet(s)

WebDev : association d'un attribut à un champ de la table d'une zone répétée

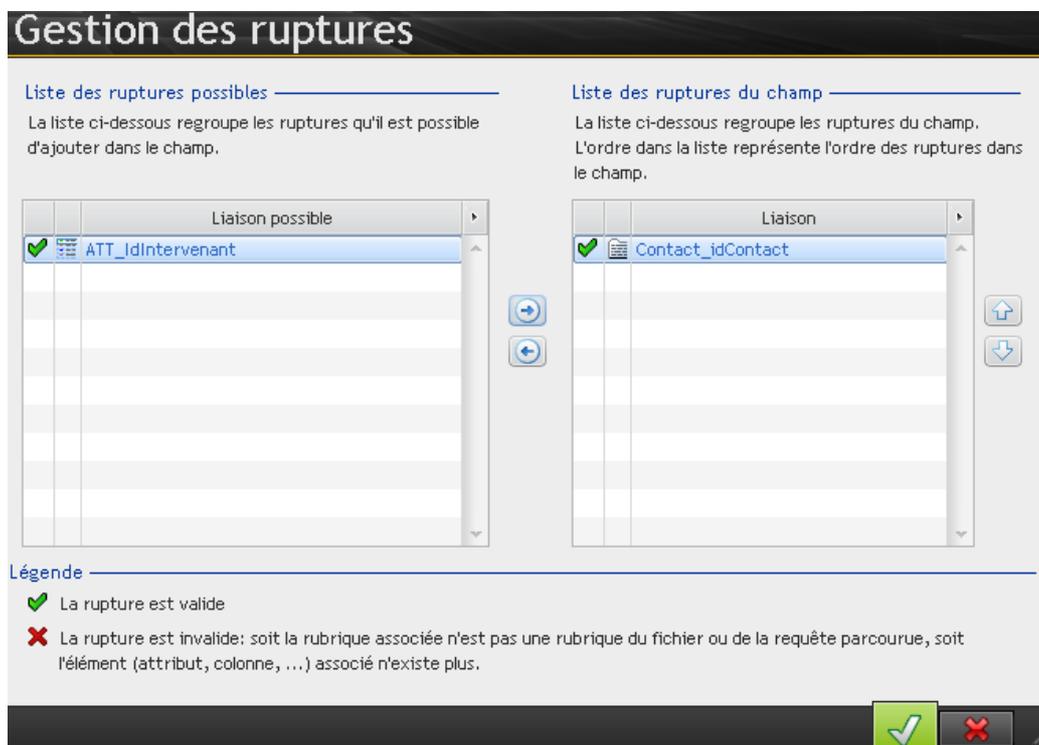


### Complément : Utilisation des ruptures

Si l'on souhaite pouvoir afficher des sous-éléments dynamiques sans répéter l'élément principal, par exemple, afficher les informations d'une société et afficher l'ensemble de ses filiale. On peut alors créer une rupture, cela va ajouter un haut et un bas de rupture à la zone répétée, concepts similaires à un entête et pied de page. On va alors pouvoir ensuite indiquer dans le haut de la rupture (ou le bas), les informations qui n'apparaîtront qu'une fois, celles discriminantes et, dans le contenu de la zone, les éléments se répétant.



WebDev : rupture dans une zone répétée



WebDev : création d'une rupture dans une zone répétée

Exemple d'une zone répétée avec un haut de rupture (en blanc) et un contenu se répétant (en gris), le contenu affichera les différentes interventions effectuée par le même intervenant (1 ou plusieurs). Cette zone répétée affichera donc la liste des intervenants et leurs interventions associées.



Image 1 WebDev : Exemple d'une zone répétée

## WLangage

```
ZoneRépétéeAjouteLigne (nomZR, valAtt1, valAtt2)
ZoneRépétéeSupprimeTout (nomZr)
```



### Exemple : Afficher une liste de fichiers dans une zone répétée

On imagine un affichage avec une vignette, un lien vers un fichier à télécharger, un champ indiquant la taille et un autre indiquant la date ainsi qu'un bouton "Actualiser" pour rafraîchir la zone.

Une fois la zone répétée créée et la mise en forme faite, voici les différentes fonctions à implémenter pour remplir la peupler :

```
//Pour récupérer la taille
fTaille("cheminFichier")
//Pour récupérer la date
fDate("cheminFichier")
//Pour le lien
//Afficher le nom fFichier
//Pour le disque fDisque
//Pour l'extension fExtension
//Pour le répertoire fRépertoire
fExtraitChemin("path", fFichier)
//Lors d'un clic sur le lien
FichierAffiche("cheminComplet", "application/unknown",
"nomFichier.ext")
//Fonction pour tester si image
//Renvoie Vrai ou Faux
fEstUneImage("cheminComplet")
```

```
//Code du bouton "Actualiser"
ZoneRépétéeSupprimerTout (nomZR)
//Dans une boucle
ZoneRépétéeAjouteLigne (nomZR, att1, att2, ...)
```

Pour jouer sur le CSS, on peut ajouter un champ div avec display : none si, par exemple, le fichier n'a pas d'image associée.

```
<DIV style="display:none">
```

On pourrait ajouter 2 attributs à la zone répétée "Att\_HTML\_Avant" et "Att\_HTML\_Apres" que l'on manipule ensuite en tester si le fichier est une image ou non. Si ce n'est pas une image, alors on peuple ces champs.

```
SI fEstUneImage("cheminComplet") = Faux ALORS
nomZR.Att_HTML_Avant = "<div style='display :none'>"
nomZR.Att_HTML_Apres = "</div>"
```

## 7. Popup

### Création d'une popup

Les popups vous permettent d'afficher des informations comme des formulaires sur votre site Web. Elles se différencient de PageAfficheDialogue par le fait qu'elles sont gérées côté navigateur et donc que les éléments en arrière plan restent actifs (ne sont pas grisés comme lors de l'utilisation de PageAfficheDialogue).

Pour créer une popup, il suffit d'aller dans "Page->Popup->Nouvelle popup" ou dans "Création->Popup".

Le code d'une popup se fera du côté navigateur (partie verte) car tout est géré par le "client" c'est à dire le navigateur de l'internaute qui accède au site Web.

Le code WLangage lié à une popup est :

```
PopupAffiche (nomPopup)
PopupFerme ()
```

### Fonction d'un bouton submit

Une petite explication sur le fonctionnement d'un bouton de soumission d'un formulaire est important car fortement lié à une popup. Voici les actions qui s'enchaînent lors d'un clic sur un bouton de validation :

1. Code du clic du bouton côté navigateur (bouton appelant, partie verte) -> où se trouve le code PopupAffiche(nomPopup)
2. Page envoyée au serveur
3. Code du clic du bouton côté serveur (bouton appelant, partie jaune)
4. Page renvoyée au navigateur



### Méthode : Code de la popup avec ou sans AJAX

Imaginons le code d'une popup ayant un bouton "Annuler" et un bouton "Supprimer", lancer via un PopupAffiche(nomPopup) depuis la page appelante via un bouton de type submit.

#### Sans AJAX

```
//Code du bouton Annuler
//Partie navigateur
PopupFerme ()
//CETTE PARTIE SERT A INTERROMPRE LE CYCLE DU BOUTON
//AYANT APPELE LA POPUP
//EN EFFET, SI ON ANNULE, ON A PAS BESOIN D'EXECUTER LA
PARTIE SERVEUR DU BOUTON APPELANT
//SANS CE RETOUR, LA PAGE SERAIT RAFRAICHIE ET LES DONNEES
DEJA SAISIES SERAIENT PERDUES
RETOUR
```

```
//Code du bouton Supprimer
//Partie serveur sans AJAX
//AU CHOIX DANS LE CODE DE LA POPUP
//OU DANS LE CODE DU BOUTON APPELANT SANS AJAX
TableSupprime(tableProduit)
TableAffiche(tableProduit)
```

#### Avec AJAX

```
//Code du bouton Annuler
//COMME AJAX EST ACTIVE LE CODE DU BOUTON APPELANT COTE
SERVEUR NE RAFRAICHIRA PAS LA PAGE
//PUISQUE ENGLOBE DANS DU JAVASCRIPT
```

```
//Code bouton "Supprimer"
//Partie navigateur
PopupFerme ()
```

```
//Code du bouton Supprimer
//Partie serveur avec AJAX activé
TableSupprime(tableProduit)
TableAffiche(tableProduit)
```



## Fondamental : Partie page appelante

J'ai besoin :

- De code navigateur uniquement :  
On interrompt alors le cycle "submit" du bouton appelant via un PopupAffiche(nomPopup) suivi d'un retour

```
//Code du bouton appelant
//Partie navigateur
PopupAffiche(nomPopup)
//Cela sert à ne pas exécuter le code
//du bouton appelant partie serveur
RETOUR
```

- De code navigateur + code serveur :  
On active AJAX dans le bouton partie code serveur

```
//Bouton appelant
//Partie navigateur
PopupAffiche(nomPopup)
```

```
//Bouton appelant
//Partie serveur
//Code a exécuter suite action popup
//Comme HRAZ(Objet) si ajout objet
//Ou HLit(Objet, hNumEnCours) si modification objet
FichierVersPage()
```



## Exemple : Création d'une popup de modification ou d'ajout d'un produit

### Page appelante

//Code bouton "Nouveau" appelant

```
//Code bouton "Nouveau" appelant
//Partie navigateur
PopupAffiche(nomPopup)
```

```
//Code bouton "Nouveau" appelant
//Partie serveur -> AJAX activé !
gbModif = Faux
//HRAZ permet de positionner Produit..NouvelEnregistrement
à Vrai
HRAZ(Produit)
FichierVersPage()
```

//Code du bouton "Modifier" appelant

```
//Code du bouton "Modifier" appelant
```

```
//Partie navigateur
PopupAffiche(nomPopup)
```

```
//Code du bouton "Modifier" appelant
//Partie serveur
//gbModif n'est pas utile si on utilise la fonction
//Fichier..NouvelEnregistrement (positionné grâce au HRAZ)
//qui nous permet alors de savoir si c'est un ajout ou une
modification
gbModif = Vrai
//Pour recharger la ligne si clic sur Nouveau puis annuler
HLit(Produit, hNumEnCours)
//On bascule les données de hNumEnCours sur le formulaire
FichierVersPage()
```

### Code de la popup

```
//Bouton "Annuler"
//Partie navigateur
PopupFerme()
RETOUR
```

```
//Bouton "Valider"
//Partie navigateur
//Contrôles à effectuer
//Vérification des champs de saisie
//Si controles OK
PopupFerme()
```

```
//Bouton "Valider"
//Partie serveur
//Contrôles à effectuer
//Vérification des champs de saisie
//Si controles OK
PageVersFichier()
//2 solutions ici soit on utilise gbModif
//soit on utilise Produit..NouvelEnregistrement
//Si gbModif alors
Si Produit..NouvelEnregistrement ALORS
HAjoute(Produit)
SINON
HModifie(Produit)
FIN
TableAffiche(tableProduit)
//Si AJAX activé ou que le bouton n'est pas de type submit
//PopupFerme()
```

## 8. Multilinguisme

Pour chaque donnée textuelle (libellé, contenu, etc), une icône WebDev  vous permet d'aller chercher sur Internet la traduction (s'appuie sur google translate à priori). Vous pouvez définir un projet multilingue dans la description du projet afin d'indiquer à WebDev les différentes langues de votre projet.

Une fois vos différents contenus traduits, vous pourrez changer de langues dynamiquement dans le code (via un clic sur un bouton par exemple) en utilisant la fonction Nation() :

```
//Bouton anglais
//Partie serveur
Nation(nationAnglais)
//Réaffichage de la page d'accueil
PageUtilise(pageAccueil)
```



### Remarque : Pas de code après PageUtilise

Le code situé après la fonction PageUtilise est du code mort puisqu'il ne sera jamais exécuté car la fonction PageUtilise ferme la page et la recharge.

## 9. Cookies

Les cookies vous permettent de stocker des données qui seront sauvegardées dans le navigateur du client. Extrêmement utilisé par la plupart des applications Web, cela a l'avantage de personnaliser le contenu.

Nous l'avons utilisé sous ArtsEtCulture pour stocker sauvegarder les filtres sélectionnés sur le tableau des commissions. Ainsi, lors de la prochaine visite, l'utilisateur ne perd pas ses filtres.

```
//Pour écrire un cookie
CookieEcrit("nomCookie","Valeur") // -> limité à 4000
caractères maximum
//Pour récupérer le contenu d'un cookie
CookieLit("nomCookie")
```



### Fondamental : Accéder aux contenus des Cookies pour l'ensemble du projet

Dans la plupart des cas, il va vous falloir définir des variables globales qui contiendront le contenu de vos cookies si vous souhaitez pouvoir y accéder dans l'ensemble du projet.



### Exemple : Récupération du cookie

```
//Partie code serveur du projet
//Gestion des préférences enregistrées dans le cookie
SI gsCommissionDegré="" _ET_ gsCommissionDispositif="" _ET_
gsCommissionDomaine="" _ET_ gsCommissionDépartement="" _ET_
gsCommissionVille="" ALORS
filtreCommission est une chaîne =
CookieLit("FILTRE_PAGE_COMMISSION")
SI filtreCommission<>"" ALORS
//ExtraitChaîne permet d'extraire une partie de la chaîne
sur un séparateur
//Ici le séparateur est une virgule
gsCommissionDegré = ExtraitChaîne(filtreCommission,1,",")
gsCommissionDispositif =
ExtraitChaîne(filtreCommission,2,",")
gsCommissionDomaine = ExtraitChaîne(filtreCommission,3,",")
gsCommissionDépartement =
```

```
ExtraitChaîne(filtreCommission,4,"")
gsCommissionVille = ExtraitChaîne(filtreCommission,5,"")
FIN
FIN
```



### Exemple : Écriture du cookie

```
//Code exécuté à chaque modification d'un filtre
//Partie serveur
//Plusieurs valeurs dans un même cookie séparées par une virgule
CookieEcrit("FILTRE_PAGE_COMMISSION",COMBO_Degre..ValeurAffichée+ "," +
COMBO_Dispositif..ValeurAffichée+ "," +COMBO_Domaine..ValeurAffichée+ "," +
COMBO_Dptmt..ValeurAffichée+ "," +COMBO_Ville..ValeurAffichée)
```

## 10. Pages internes et composants

Le champ page interne vous permet de réutiliser des composants sur n'importe quelle page. Ce champ permet d'inclure une page et son code dans le champ d'une autre page. C'est une sorte de WebApp.

Les composants sont des "applications" qui peuvent être intégré à votre projet. Ils s'appuient donc généralement sur des pages internes à ajouter à votre projet via un champ page interne.

Par exemple, le composant "Panier" qui est fourni dans WebDev vous permet d'ajouter un panier et toutes les fonctions existantes (ajout d'élément au panier, calcul du tarif, etc) à votre projet.



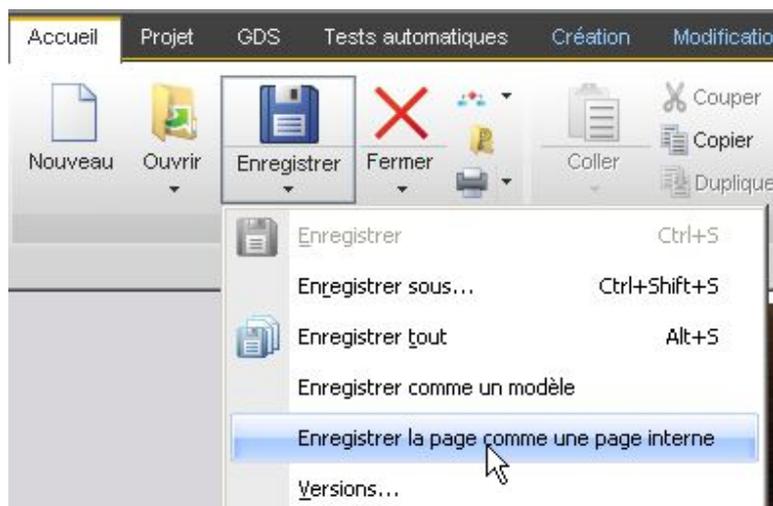
### Complément : Type de composant

Un composant ayant comme extension .WDI est fourni avec une documentation.  
Un composant ayant comme extension .WDK est fourni sans documentation.



### Exemple : Transformer une page en page interne

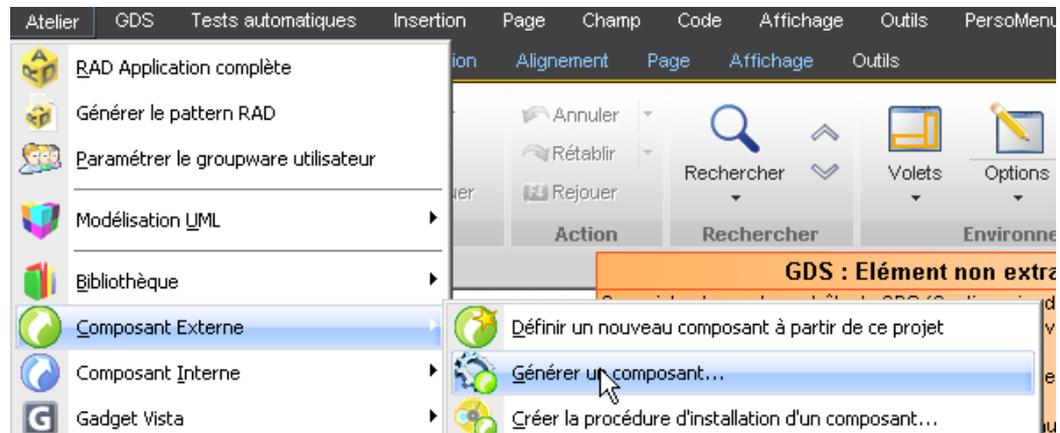
Une fois votre page créée, faites "Fichier->Enregistrer comme page interne" ou dans l'onglet "Accueil" -> "Enregistrer la page comme une page interne".



WebDev : Sauvegarde d'une page comme page interne

## a) Création d'un composant

Imaginons que vous ayez réalisé un projet avec du code que vous souhaiteriez pouvoir intégrer à d'autres projets. Ce projet comprend des pages spécifiques et des procédures associées. Pour ce faire, il vous suffit d'aller dans la partie "Atelier" puis "Composant externe" et "Générer un composant".



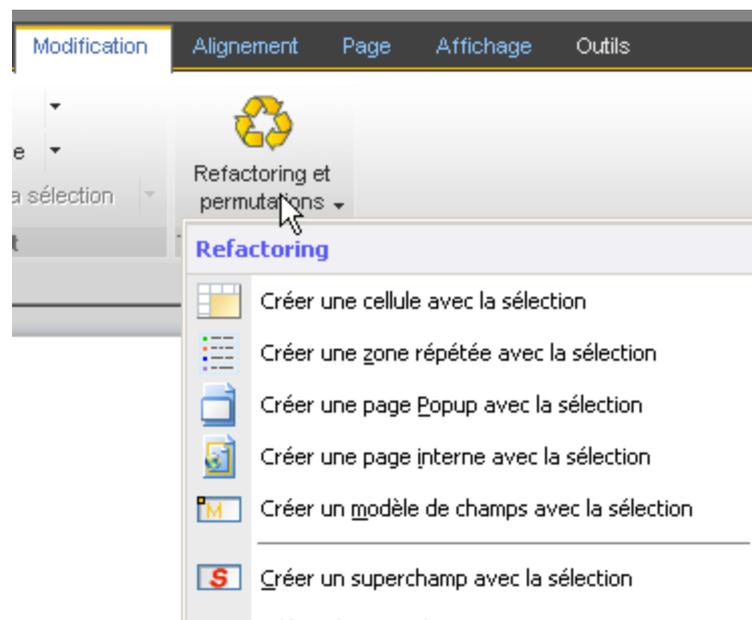
WebDev : Création d'un composant

## 11. Refactoring

### *Un mécanisme pour transformer un élément*

Le refactoring de WebDev vous permet de "transformer" un élément vers un autre type. Par exemple, transformer une table en liste, transformer une cellule en popup, etc.

L'option se trouve dans "Modification" puis "Refactoring et permutations" :



WebDev : Transformation d'éléments



### *Exemple : Transformer une cellule en page popup*

Dans la description de la cellule, dans la partie IHM, cocher "Le champ peut être superposé".

Puis dans "Refactoring et permutations", partie "permutations" -> "Cellule / Page popup".

## 12. Upload / Download de fichiers

L'upload/download de fichiers se fait via l'utilisation du champ upload. Le plus simple est d'utiliser les formulaire prédéfinis accessible dans "Création", partie "Bouton" :

WebDev : Upload de fichiers

Il existe 3 versions possibles :

- Vierge : limité à 4Mo sans jauge
- Envoi 1 seul fichier
- Envoi de plusieurs fichiers

**Les 2 dernières versions nécessite le plugin flash.**

## 13. Crypter les données d'une table

Dans la partie analyse, il suffit de faire un clic droit sur la table voulue puis "Description->Détail->Protection", vous pouvez alors crypter les données selon différents encodages.

## 14. Gestion des droits

Comme indiqué précédemment, vous pouvez utiliser le groupware pour implémenter une gestion fine des droits.



### Conseil : Utilisez le modèle de page

Vous pouvez utiliser le modèle de page pour afficher ou masquer les menus.

On peut créer une variable globale au projet comme `estAdmin` par exemple qui sera initialisée lors de la connexion de l'utilisateur puis, avec du code au sein du modèle de page, récupérer les informations du profil et afficher ou masquer les éléments en fonction de ces informations.

## D. Règles de "bonne conduite"

- Éviter l'utilisation de variables globales (global au projet)
- Encapsuler le code, dès que possible, dans une fonction (programmation procédurale)
- Faire **un contrôle systématique des champs de saisie côté navigateur et côté serveur** (règle de sécurité)
- Commenter votre code à minima à 30% (au mieux 1 ligne de commentaire pour 3 lignes saisies)
- Supprimer les éléments devenus inutiles (test, etc) au fur et à mesure
- **Tester votre code et vos modifications** avant de les déployer ou de les sauvegarder dans le GDS
- Pour des éléments globaux (devant être accessible à l'ensemble du site comme des popups par exemple), pensez à utiliser le modèle de page
- Si **vous avez 2 boutons dans votre page qui doivent effectuer la même action** (un en haut et l'autre en bas par exemple), codez la fonction dans un des boutons et utilisez la fonction **ExécuteTraitement(NomBtnAvecCode, trtcllic)**
- **Utilisez du code de débogage** en faisant appel sans modération à la fonction `dbgAssertion("condition à tester", "texte à afficher si non vérifiée")`
- **Profitez au maximum de la dépendance des objets** en privilégiant l'utilisation d'objets liés aux tables de votre analyse. En effet, cela vous permettra d'alléger votre code. Par exemple, pour filtrer l'affichage d'une zone répétée ou d'une table, vous pourrez alors faire :  
`HFiltre(Produit, ID, 6)`  
`TableAffiche(tableProduit)` ou `ZoneRepeteeAffiche(zrProduit)` -> n'affichera que le produit ayant l'id 6
- **Lancer systématiquement un audit statique** et dynamique de votre projet et corriger les erreurs remontées par ces audits. Le plus simple et rapide à corriger étant les erreurs remontées par l'audit statique.

## E. Analyse et BDD

L'analyse WebDev est la pierre angulaire de votre projet, c'est la base sur laquelle vous allez développer votre projet. Il vaut donc mieux prendre un maximum de temps pour bien spécifier celle-ci.



### Conseil : Ne négligez pas cette partie

La base de données est garante de la cohérence et de la consistance des données

de votre application, c'est donc **l'élément structurel le plus important de votre projet**. Elle est parfois négligée car n'est pas un élément visible de vos utilisateurs contrairement à l'aspect graphique par exemple. Cependant, une bonne analyse vous permettra de limiter les bugs, les lenteurs et apportera surtout une meilleure maintenabilité. Vous devez modéliser cette base en utilisant la *méthode Merise*<sup>17</sup> ou *UML*<sup>18</sup>. Votre analyse doit respecter la *forme normale de Boyce - Codd*<sup>19</sup>.

**Cette tâche est celle la plus importante de votre projet et se doit d'être faite dans les règles de l'art.**



### Remarque : Une analyse peut servir à plusieurs projets

Plusieurs projet WebDev peuvent faire référence à la même analyse. Une application de gestion complète peut être séparé en plusieurs modules et donc plusieurs projets mais, à condition que l'analyse ait été prévue pour, faire référence à la même analyse.

### Types de connexions

- *ODBC*<sup>20</sup> : logiciel permettant d'accéder de façon unifiée à plusieurs bases de données (surcouche d'abstraction) implémentée initialement par Microsoft
- *OLEDB*<sup>21</sup> : devait remplacer ODBC et y ajouter de nouvelles fonctionnalités mais, dans les faits, ODBC est toujours majoritairement utilisé
- "Natif" : fourni par WebDev, ce sont des pilotes spécifiques à chaque type de base (MySQL, postgresQL, etc) pour s'y connecter. Ils sont parfois payants. **Le terme natif est trompeur car ils ont leurs limitations.**



### Attention : "Natif" est un terme marketing

Natif ne veut pas forcément dire ce que l'on pourrait penser à première vue soit une connexion complètement compatible avec WebDev.

En effet, par exemple pour le pilote "natif" MySQL est limité par rapport à l'utilisation d'hyperfile (base de données développée par PC Soft) sur énormément d'aspects. Vous aurez par exemple des problèmes lors de l'utilisation du groupware qui utilise par défaut une base hyperfile mais aussi pour la gestion des identifiants automatiques des tables, la gestion des transactions, les types de champ, etc.

Voir la liste des limitations sur : [http://doc.pcsoft.fr/fr-fr/?5517002&name=acces\\_natif\\_mysql\\_limitations\\_remarques](http://doc.pcsoft.fr/fr-fr/?5517002&name=acces_natif_mysql_limitations_remarques)



### Conseil : Privilégiez hyperfile

Comme indiqué en introduction, WebDev n'aime pas trop que l'on sorte du "cadre", il supportera beaucoup mieux une base hyperfile que tout autre type de base même avec une connexion dite "native". Lors de **nos expérimentations avec le pilote natif MySQL, nous n'avons eu que des problèmes ou contretemps lié à ce pilote dont, parfois, des bugs bloquants**. Il a été difficile de contourner tous les problèmes et nous avons même été obligé de réclamer une mise à jour à PC Soft.

En conséquence, **privilégiez l'utilisation d'hyperfile en client serveur.**

### Types de fichiers

- Répertoire .ANA : contient toutes les informations de l'analyse et les sauvegardes des différentes versions

17 - <http://fr.wikipedia.org/wiki/MERISE>

18 - [http://fr.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://fr.wikipedia.org/wiki/Unified_Modeling_Language)

19 - [http://fr.wikipedia.org/wiki/Forme\\_normale\\_\(bases\\_de\\_donn%C3%A9es\\_relationnelles\)](http://fr.wikipedia.org/wiki/Forme_normale_(bases_de_donn%C3%A9es_relationnelles))

20 - [http://fr.wikipedia.org/wiki/Open\\_Database\\_Connectivity](http://fr.wikipedia.org/wiki/Open_Database_Connectivity)

21 - <http://fr.wikipedia.org/wiki/OLEDB>

- .WDD : contient la description sommaire de l'analyse, la structure des différents fichiers de données
- .FIC : fichier de données
- .NDX : index / clés
- .MMO : mémos / blob
- .FTX : moteur de recherche dans la BDD pour 1 fichier (full-text)



### Remarque : Cryptage de l'analyse

Une analyse peut être protégée par un mot de passe.

## 1. Création de l'analyse

Vous pouvez partir du *modèle conceptuel de données*<sup>22</sup> ou bien directement du *modèle relationnel*<sup>23</sup>. En général, la méthode *Merise*<sup>24</sup> vous fait commencer par le *MCD*<sup>25</sup> (Modèle Conceptuel des Données).

### Ouvrir ou créer une analyse

Allez dans l'onglet *Projet*, cliquez sur "Charger l'analyse du projet" :



Projet : charger l'analyse

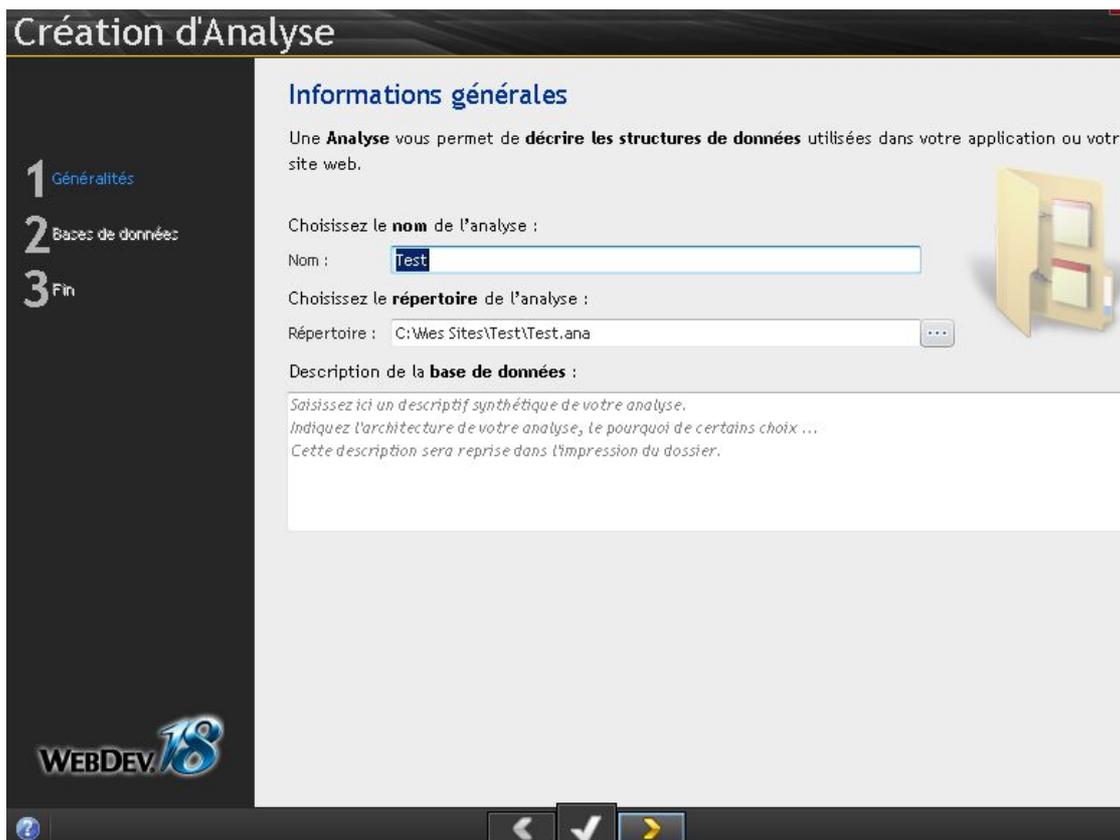
Indiquez le nom et l'emplacement (en général dossier en .ana) de votre analyse, **pensez aussi à y mettre une description détaillée de l'architecture de votre analyse.**

22 - [http://fr.wikipedia.org/wiki/Mod%C3%A8le\\_conceptuel\\_de\\_donn%C3%A9es](http://fr.wikipedia.org/wiki/Mod%C3%A8le_conceptuel_de_donn%C3%A9es)

23 - [http://fr.wikipedia.org/wiki/Mod%C3%A8le\\_relationnel](http://fr.wikipedia.org/wiki/Mod%C3%A8le_relationnel)

24 - [http://fr.wikipedia.org/wiki/Merise\\_\(informatique\)](http://fr.wikipedia.org/wiki/Merise_(informatique))

25 - [http://fr.wikipedia.org/wiki/Mod%C3%A8le\\_conceptuel\\_de\\_donn%C3%A9es](http://fr.wikipedia.org/wiki/Mod%C3%A8le_conceptuel_de_donn%C3%A9es)



Analyse : création

Indiquez les types de base de données utilisées par votre analyse.



Image 2 Analyse : sélection des SGBD



### Remarque

**Il est possible d'en sélectionner plusieurs en même temps.** Dans ce cas, l'outil vous demandera les paramètres et générera les différentes connexions aux bases. Il sera possible de basculer d'une connexion à l'autre au sein de l'application avec les fonctions :

```
<Résultat> = HOuvreConnexion(<Nom de la connexion>)
<Résultat> = HChangeConnexion(<Nom du fichier> , <Nom de la
nouvelle connexion>), utilisez "*" comme Nom du fichier
pour indiquer que toutes les tables utiliseront cette
nouvelle connexion
<Résultat> = HFermeConnexion(<Nom de la connexion>)
```

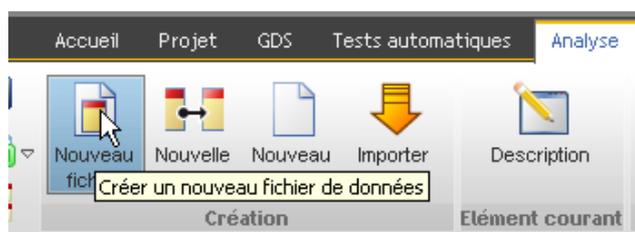


### Conseil : Privilégiez hyperfile

La base de données hyperfile étant soutenue et gérée par PCSoft, société éditrice de WebDev, nous avons remarqué qu'elle est beaucoup mieux intégrée que les autres bases, nous vous conseillons donc, quand cela est possible, de privilégier l'utilisation de celle-ci. Essayez de l'utiliser en mode client / serveur car cela vous permettra d'y inclure, si nécessaire, des services ultérieures (interrogation BO, indicateurs, etc).

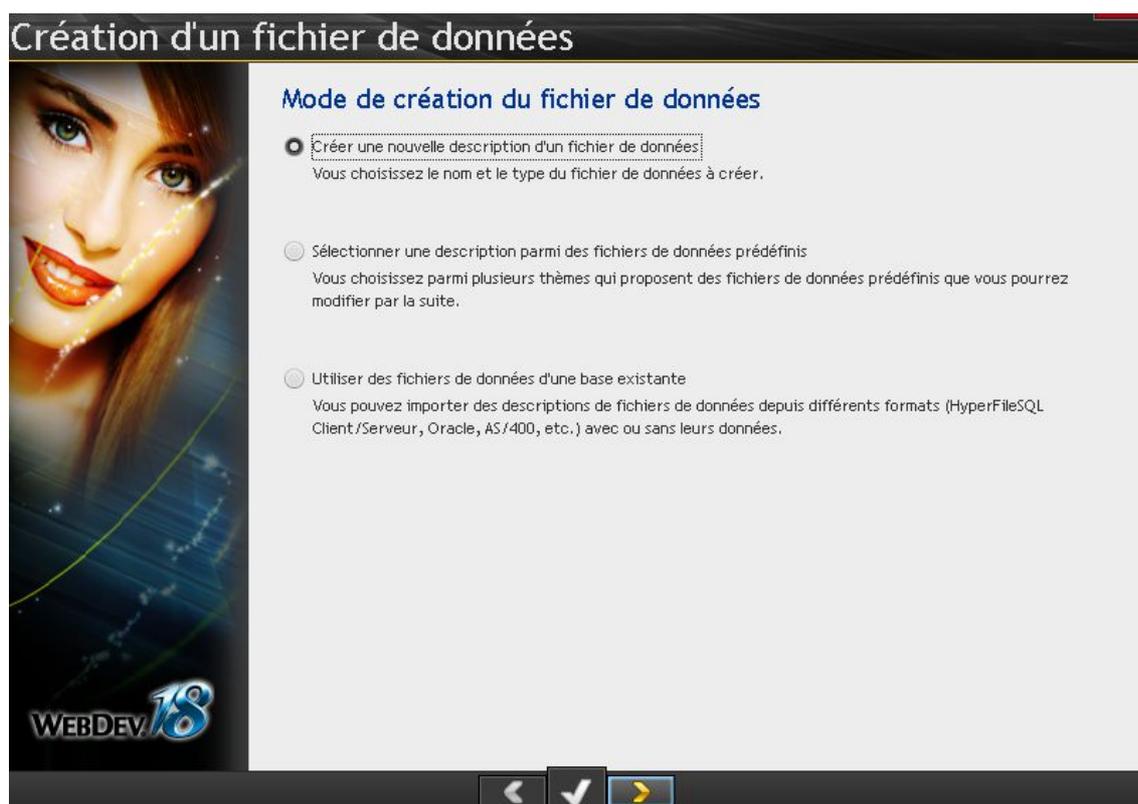
## 2. Création des fichiers de l'analyse

Dans l'onglet "Analyse", il suffit de cliquer sur "Nouveau fichier" :



Analyse : créer un nouveau fichier (ou table)

Un menu apparaît alors :



Analyse : menu de création d'un fichier (ou table)



### Remarque : Option intéressante : fichiers prédéfinis et récupérer une ou plusieurs tables d'une base de données existantes

Ces 2 options vous permettent de ne pas avoir à tout ré-écrire. La première (fichier

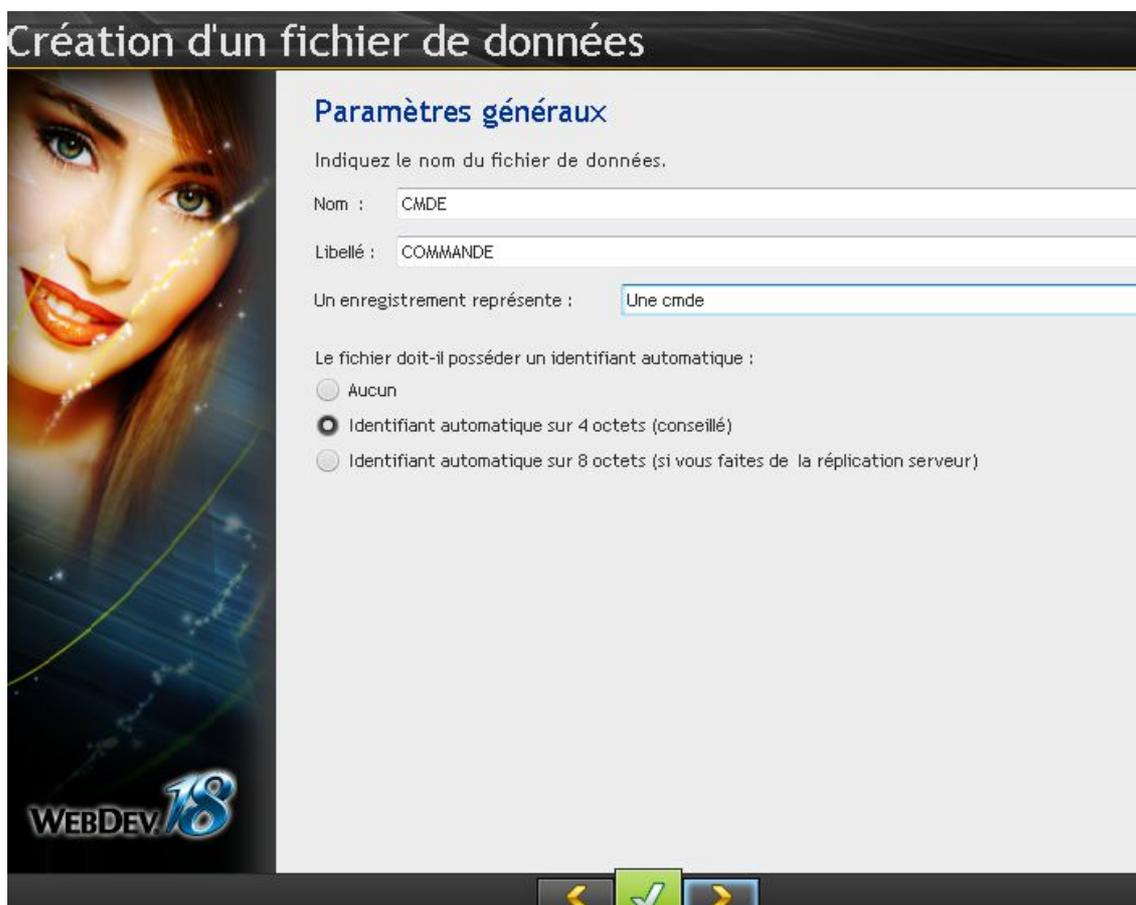
prédéfinis) sont des fichiers standards fournis dans WebDev qui répondront à de multiples besoins, la seconde est intéressante dans le cas où vous ré-écrivez une application ayant une base de données déjà existante (bien entendu, dans le cas où celle-ci a été réalisée dans les règles de l'art).

Un fichier est défini dans l'analyse par son nom. Ce nom logique sera notamment utilisé dans votre code afin d'accéder au fichier (par exemple, table "CLIENT").

Le nom physique du fichier est, par défaut, identique au nom logique.

Un libellé est associé à chaque fichier. Ce libellé est notamment utilisé par le RAD ou les champs de page lors de la création des pages Web. Par exemple, vous pouvez avoir une table "CMDE" avec un libellé "COMMANDE" pour que ce soit plus parlant lors de l'affichage.

Vous pouvez choisir la création d'un identifiant automatique calculé par le moteur de base de données (incrémental et unique).



*Analyse : options lors de la création du fichier (ou table)*



### *Fondamental : Spécificités Hyperfile et options avancées*

Les fichiers de données d'une base hyperfile locale seront créés, par défaut, dans le répertoire de l'exécutable de l'application (répertoire Exe du site). Vous pouvez modifier ce répertoire d'accès :

- au niveau de l'analyse
- via du code WLangage

Dans les options avancées, vous pouvez indiquer qu'un fichier pourra avoir une taille de plus de 2Go, ce qui n'est nécessaire que pour des fichiers ayant plusieurs

millions d'enregistrements.

**Pour les données sensibles, vous pouvez demander le cryptage d'un ou plusieurs fichiers**, via un mot de passe à saisir.

Un champ mémo, sous WebDev, est un champ *BLOB*<sup>26</sup>, champ généralement conséquent, il est donc possible de le "compresser" sous WebDev.

**CLIENT**

Nombre de rubriques et index : 2 Taille en octets : 21  Afficher dans l'ordre physique

Clé	Nom	Libellé	Type	Taille
IDCLIENT	Identifiant de CLIENT	Id. automatique	4	
photo	Photo client	Son, image, binaire...	8	

Notes de la rubrique (ut...  
Saisissez ici un descrip...  
Indiquez les cas partici...  
les informations utiles e...

Créé par : PMI  
Mis à jour par : PMI

Journaler la rubrique  
La rubrique est...

*Analyse : journaliser une rubrique*

26 - [http://fr.wikipedia.org/wiki/Binary\\_large\\_object](http://fr.wikipedia.org/wiki/Binary_large_object)

Pour accéder à la description des fichiers de données, vous pouvez faire un clic droit puis "Description du fichier de données" et allez dans l'onglet "Détail". Vous aurez alors accès aux possibilités de cryptage et de compression des types mémos (ou BLOB) :

*Analyse : chiffrement et compression mémo des fichiers (ou table)*

### 3. Création des rubriques des fichiers

La création des rubriques (ou attributs) des fichiers (ou table) est plutôt simple à prendre en main. Un rubrique est définie par :

- Un nom (accessible en code par NOMTABLE.nom)
- Un libellé (important, c'est ce qui sera affiché par défaut lorsque vous ajouterez les rubriques sur une page comme un champ de saisie)
- Un type (texte, entier, réel, mémo ou blob, date, interrupteur, liste, combo, image, etc)
- Une taille ou un format (longueur, masque de saisie, etc)



#### *Remarque : Indiquez les libellés*

Bien définir vos libellés vous permettra de gagner du temps lors de la réalisation des pages. En effet, cela s'affichera par défaut lorsque vous créerez un champ de saisie. Par exemple, associez le libellé "Date de naissance" à la rubrique (ou attribut) "dateNaissance".

**Description des rubriques et index d'un fichier de**  
**CLIENT**

Nombre de rubriques et index :  Taille en octets :   Afficher dans l'ordre physique

Clé	Nom	Libellé	Type	Taille
	IDCLIENT	Identifiant de CLIENT	Id. automatique	4
	Sexe	Sexe	Interrupteur	1

Paramètres du champ relié à la rubrique sélectionnée (Informations partagées)

Type de champ

**Général**

Libellé : Français (5)

Options : Français (5)

Masculin  
Féminin

*Analyse : spécifier les propriétés d'une rubrique (options interrupteur, etc)*

## F. WLangage

Le WLangage est le langage interne à WebDev et WinDev en général. C'est un langage avec une syntaxe standard, très verbeux et disposant de fonctions fournis par le framework de développement. L'ensemble de la documentation sur les fonctions intégrées au langage se trouve sur : <http://doc.pcsoft.fr/fr-FR/>



### *Remarque : Pensez à tester le retour des fonctions utilisées*

La plupart des fonctions WebDev renvoient faux en cas d'erreur. Pensez donc à tester systématiquement le retour d'une fonction pour vous assurer qu'il n'y a pas d'erreurs et intégrer un code de traitement des erreurs.



### *Remarque : Français ou anglais au choix*

Pour les plus chauvins d'entre nous, il est possible d'écrire tout le code en français, ce qui est la langue native de ce langage mais vous pouvez basculer de la langue

anglais à la langue française indifféremment, le compilateur comprenant autant la syntaxe "if" que "si" par exemple. On peut même mixer les 2 si on le souhaite mais nous le déconseillons pour des raisons évidentes de lisibilité.

## 1. Syntaxe de base

```
// : permet d'insérer un commentaire
```

```
SI condition
ALORS
action quand condition vraie
SINON
action quand condition fausse
FIN
```

```
SELON variable
CAS 1 :
CAS 2 :
FIN
```

```
TANT QUE condition
//Traitement
FIN
```

```
POUR VAR = ValMin A ValMax
//Traitement
FIN
```

```
POUR TOUT FIC
//Traitement
FIN
```

```
//Opérateur ternaire
expr ? "vrai" sinon "faux"
```

### a) Déclaration de variables

```
maVar est un <type>
maChaine est une chaîne
monEntier est un entier
maChaine = "TOTO"
monEntier = 3
maChaine1, maChaine2 sont des chaînes
```



#### *Remarque : Initialisation multiple par défaut impossible*

On ne peut initialiser à une valeur par défaut des déclarations multiples de variable. On ne peut le faire que lors de la déclaration d'une seule variable :  
maChaine1 est une chaîne = "test"

## b) Variables et portée

### Code du projet

Le code du projet permet de définir des variables globales au projet (à utiliser avec parcimonie). Mais cela peut être utile comme, par exemple, l'initialisation de la base de données (adresse ip, nom util, etc).

On peut accéder au code du projet en allant dans Projet -> Code

### Code de page

<Code de déclaration globale de page> : accessible via un clic droit sur la page puis "Code de la page". Une variable déclarée dans ce code sera globale à la page.

<Chargement (onload) de page> : une variable déclarée à ce niveau sera globale côté navigateur.

<autres evenements> : variable locale à l'événement.



### Remarque : Partager des données entre navigateur et serveur

On peut partager des données entre navigateur et serveur en utilisant un champ caché comme un bouton par exemple qui sera lancé via ce type de code :

ExécuteTraitement(btnVoulu, trtClic, parametre)

## c) Propriétés des objets

Objets : boutons, champs, etc

Les propriétés des objets sont les positions, les couleurs, le libellé, le contenu, etc.

Pour accéder aux propriétés d'un objet, il suffit de faire :

nomObjet..nomPropriete

Pour lui affecter une valeur :

nomObjet..nomPropriete = "Valeur"

```
saiCA..Couleur = Rouge
saiCA..Couleur = RVB(0, 255, 255)
saiDateDemande..Etat = Actif
saiDateDemande..Etat = AffichageSeulement
saiDateDemande..Etat = Gris 
btnModifier..Visible = Vrai
btnModifier..Visible = Faux
```

## d) Proc dures et fonctions

En WLangage, **il n'existe pas de distinction entre les proc dures et les fonctions**. Les proc dures et les fonctions sont g r es de la m me fa on. **Une proc dure comme une fonction peut retourner ou non un r sultat**.

Les collections servent   "ranger" les proc dures par domaine.

Pour cr er une proc dure, il suffit de faire un clic droit sur Proc dure puis cr er une nouvelle collection, par exemple "Fcts gestion" puis faire un clic droit sur la collection et cliquer sur "Nouvelle proc dure".

On peut d clarer une proc dure ou une fonction de la mani re suivante :

FONCTION addition (nEntier1, nEntier2)

PROCEDURE addition (nEntier1, nEntier2)

En r alit , les termes fonction et procedure repr sente la m me chose.

```
PROCEDURE addition (nEntier1, nEntier2)
nSomme est un entier
nSomme = nEntier1 + nEntier2
RENOYER nSomme
```

On peut affecter le résultat d'un calcul ou d'un code directement sur RENOYER

```
PROCEDURE calculTTC(local nPrix, local nTVA = 19,6)
RENOYER = nPrix * (1 + nTVA/100)
```



### Attention : Attention aux réels

Évitez l'utilisation d'un nombre réel dans une boucle. En effet, **la représentation d'un nombre réel est limitée et n'est jamais précise à 100%, c'est pourquoi il faut éviter ce type de code :**

X est réel = 19,6

X = X - 19 - 0,6

//Ce traitement a de fortes chances de ne jamais être exécuté

Si X = 0 ALORS

//Traitement

FIN



### Remarque : Passage par référence par défaut

Les variables passées en paramètre sont envoyées par référence par défaut (inout).



### Complément : Passage par valeur

Pour passer un paramètre par valeur à la fonction (sans modification de la variable si on l'utilise directement dans le code), il faut alors déclarer des variables locales à la fonction de cette manière :

```
PROCEDURE addition (local nEntier1 est un entier, local nEntier2 est un entier)
```

### Appel de procédure

```
monEntier1 est un entier = 3
monEntier2 est un entier = 2
maSomme = addition(monEntier1, monEntier2)
Info(maSomme) // -> affiche 5
```



### Complément : Valeur par défaut

Pour affecter une valeur par défaut aux paramètres d'une fonction :

```
PROCEDURE addition (local nEntier1 est un entier, local nEntier2 est un entier = 0)
```

### Procédure avec variables dynamiques

```
PROCEDURE addition(*)
nSomme est un entier
POUR i=1 _A_MesParamètres..Occurrence
nSomme += MesParamètres[i]
FIN
RENOYER nSomme
```



### Complément : Différence entre A et \_A\_

L'utilisation de **\_A\_** (A avec des underscores à gauche et à droite) **permet de spécifier au compilateur de n'évaluer qu'1 seule fois la borne max**. C'est une optimisation simple puisque l'on sait que le nombre de paramètres à chaque itération reste fixe.

**Certains cas ne s'y prête pas**, un exemple simple serait un code qui récupérerait une valeur au hasard, comme cette valeur change il faut la réévaluer à chaque fois, par exemple :

POUR i=0 A Hasard(1,100)

#### e) Transtypage automatique

Sous WebDev, le transtypage se fait de façon automatique.

```
maChaine est une chaîne = "2"
nEntier = maChaine // -> nEntier aura alors la valeur 2
```

#### f) La logique booléenne

##### Différence entre ET, \_ET\_, OU et \_OU\_

Le ET équivaut au & de PHP : les conditions composées de ET sont toujours évaluées dans leur totalité (même si la première condition est fausse).

Le **\_ET\_** équivaut au && de PHP : les conditions composées de **\_ET\_** sont évaluées de manière optimisée (s'il la première partie de l'expression est fausse, la suite de l'expression n'est pas évaluée).

Le OU équivaut au | de PHP : les conditions composées de OU sont toujours évaluées dans leur totalité (même si la première condition est vraie).

Le **\_OU\_** équivaut au || de PHP : les conditions composées de **\_OU\_** sont évaluées de manière optimisée. Si la première partie de l'expression est vraie, la suite de l'expression n'est pas évaluée.

```
//Pour éviter d'évaluer toute la condition puisque la
première est déjà fausse
//Ce code effectuera toujours le traitement 2
SI 1>2 _ET_ 2=2 ALORS
//Traitement1
SINON
//Traitement2
FIN
```



### Conseil

Dans la plupart des cas, il est plus logique d'utiliser **\_OU\_** et **\_ET\_**.

#### g) Code des objets divers

##### **i Code pour la gestion des listes**

```
ListeAjoute(nomListe, "Valeur à ajouter")
ListeSupprimeTout(nomListe)
ListeSelect(nomListe) // -> renvoie l'indice de la ligne
sélectionnée (-1 si rien n'est sélectionné)
```

```

indiceSelectionne = nomListe // -> un appel direct à la
liste renvoie son indice
valeur = NomListe[indice]
valeurSelectionnee = nomListe[nomListe] // -> ou nomListe
tout court
valeurSelectionnee = nomListe..ValeurAffichee
nbElement = ListeOccurrence(element) // -> renvoie le nombre
d'éléments présents dans une liste

```

## h) Recyclage du code

### *Comment afficher plusieurs fenêtres de code ?*

Il suffit d'aller dans les options d'affichage et de désactiver "Recycler les fenêtres de code" ou "Réutiliser la même fenêtre de code".

## 2. Fonctions utiles

```

//Pour ouvrir une URL
//Destination peut être NavEnCours ou NouveauNav
FichierAffiche("Adresse HTTP", Destination)
//Pour récupérer l'adresse du site
//Permet de faire des tests sur l'URL
SiteAdresse()
//Pour extraire une chaîne
//Ici on récupère le 2ème élément séparé par une virgule
dans une chaîne
ExtraitChaîne(filtreCommission,2,",")
//Pour remettre le focus sur un champ, une fenêtre ou une
page spécifiée
RepriseSaisie(saiSaisiel)
//Pour tester si le projet est en "test"
//Cela veut dire test local (test depuis le poste
développeur)
//Ne prend pas en compte un projet exporté sur une instance
de test
Si EnModeTest() alors
//Ici indiqué les informations de test
//Par exemple des données de formulaire
//Ou des variables de connexions à la BDD
//Pour basculer automatiquement vers la BDD de test
saiEmail = "test@test.com"
saiMdp = "mdp"
FIN
//Pour récupérer la date du jour
//Res = "20130717"
DateDuJour()
//Liste les fichiers d'un répertoire et retourne la liste
des fichiers listés
ResListeFichier = fListeFichier("C :\path", frRécuratif)
POUR TOUTE CHAÎNE UnFichier DE ResListeFichier SEPAREE PAR
RC
// Traitement
FIN

```

```
//Code navigateur seulement
```

```
//Timer permet un appel régulier à une fonction
//Prend en paramètre le nom de la procédure
//La fréquence en centièmes de seconde
//Optionnellement, en 3ème paramètre, un numéro de timer
//qui servira d'identifiant permettant de remplacer un
timer existant par exemple
//Peut être utile pour afficher un diaporama par exemple
Timer("nomProcédure", 100)
//Pour animer la modification d'une propriété entre 2
valeurs
//Cette fonction permet d'animer la modification de la
propriété ..X
//pour déplacer un champ par exemple
//Anime la modification de la hauteur du champ
//Propriété peut être "X", "Y", "Largeur", "Hauteur",
"Opacité", etc
//La durée est en centièmes de seconde
AnimationJoueSurPropriétéChamp(nomChamp, "Propriété",
valeurInitiale, valeurFinale, Durée)
```



### Exemple : Affichage diaporama

```
//Code chargement (onload) de la page
gIndice est un entier = 1
//Appel d'une fonction AfficheDiapo qui charge une image
//Utilisation de la fonction Timer pour changer l'image toutes les 3secondes
Timer("AfficheDiapo", 300)
```

```
//Code de la fonction AfficheDiapo
PROCEDURE AfficheDiapo()
//RC est le séparateur Retour Chariot sous WebDev
i est un entier = 1
imgDiapo = ExtraitChaine(gsListeFichiers,i,RC)
SI (i>ChaineOccurrence(gsListeFichiers,RC) ALORS
i=1
SINON
i++
FIN
```



### Remarque : Autre méthode pour afficher un diaporama

Une autre méthode aurait été d'utiliser une image dynamique en activant le défilement automatique.

## 3. Les boucles

### POUR TOUT

Imaginons le code suivant :

```
HLitPremier(fic,rub)
TANTQUE PAS HenDehors(fic)
//Traitement
HLitSuivant(fic,rub)
FIN
```

Peut s'écrire aussi :

```
POUR TOUT fic sur rub
//Traitement
FIN
```

De même, le code suivant :

```
HLitRecherchePremier(fic, rub, "Valeur")
TANTQUE HTrouve(fic)
//Traitement
HLitSuivant(fic, rub)
FIN
```

Peut s'écrire aussi :

```
POUR TOUT fic avec rub="Valeur"
//traitement
FIN
```

## 4. Interaction avec la BDD

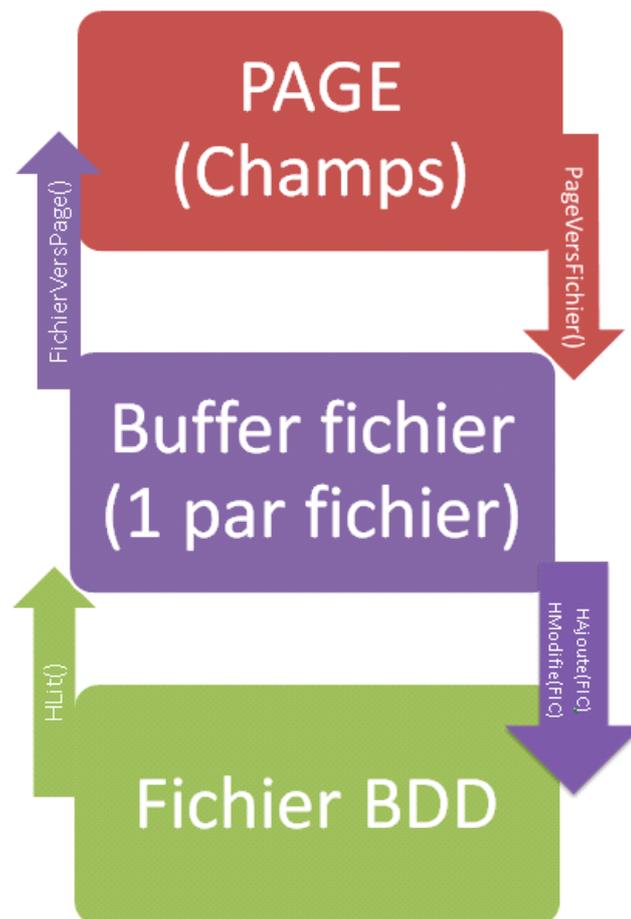


### *Remarque : Fonctionnement général*

---

Par défaut, WebDev gère une sorte de "buffer" intermédiaire qui sera rempli à partir de la base de données ou à partir de l'affichage (et vice-versa).

Voici un schéma générique expliquant ce fonctionnement :



Fonctionnement interne buffer



### Remarque : Utilisez FichierVersPage() et PageVersFichier()

Ce fonctionnement permet d'optimiser le code en utilisant des fonctions qui vont alimenter automatiquement l'enregistrement d'une table de la base de données vers les données (champs, formulaire, etc) de la page pour les afficher ainsi que son contraire, récupéré les données de la page pour les ajouter ou les modifier dans la base.

- Affichage des données de la table vers la page : on fait une recherche de l'élément à afficher (via HLitRecherche) puis on fait FichierVersPage() et toutes les données des champs de la page liés à la table que l'on manipule seront alimentés automatiquement
- Ajout ou modification de données depuis la page : on fait PageVersFichier() puis Hajoute(FIC) ou HModifie(FIC)

#### a) Création des fichiers ou table de la BDD

##### Création des fichiers ou table de la BDD

```
HCréationSiInexistant(Fic)
HCréation(Fic)
```



## Remarque

Les fichiers sont par défaut créés à la demande grâce à l'option "créer les fichiers de données automatiquement au moment de leur ouverture" dans "Projet -> Description" ce qui vous affranchi de devoir utiliser les 2 fonctions ci-dessus (HCréation ou HCréationSiInexistant).

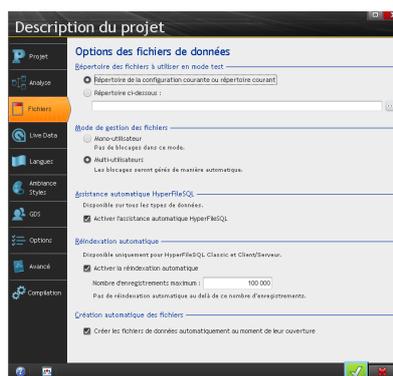


Image 3 Création automatique fichiers BDD

### b) Création d'un enregistrement

#### HAjoute

```
PageVersFichier
SI HAjoute(Client) = Faux ALORS
Erreur("Impossible d'ajouter le client")
// Détailler l'erreur en testant HErreurDoublon,
HErreurIntégrité
// HErreurBlocage
FIN
//Pour se positionner sur le nouvel enregistrement
TableAffiche(TableClient, taCourantPremier)
```



**Remarque :** *TableAffiche* : se positionner sur l'enregistrement venant d'être créé

```
TableAffiche(TableClient, taCourantPremier)
```

### c) Parcourir les fichiers et enregistrements

On peut lire la base de données en commençant par lire le premier enregistrement (ou le dernier) et boucler jusqu'au dernier enregistrement (ou le premier). La lecture sur une table fonctionne comme un pointeur sur un tuple ou enregistrement qui va s'incrémenter pour afficher le tuple suivant (ou précédent).

#### Lecture des enregistrements d'une table

```
FIC est le nom de la table à parcourir dans la base de
données
RUB est un argument optionnel et, si renseigné, doit être
une rubrique indexée
Booléen = HLitPremier(FIC,RUB) : pour lire le premier
enregistrement de la table
Booléen = HLitPrécédent(FIC,RUB) : pour lire
l'enregistrement précédent
Booléen = HLitSuivant(FIC,RUB) : pour lire l'enregistrement
suivant
```

```

Booléen = HLitDernier(FIC,RUB) : pour lire le dernier
enregistrement de la table
Booléen = HEnDehors(FIC) : indique si nous sommes en dehors
du fichier de données (dans une boucle par exemple pour
savoir si nous avons lu tous les enregistrements)

```

### *Exemple de lecture complète sur une table*

```

TANTQUE PAS HEnDehors(FIC)
//Traitement
HLitSuivant(FIC)
FIN

```



### *Complément : Opérateur POUR TOUT*

Pensez à utiliser aussi l'opérateur POUR TOUT qui peut alléger votre code. Le code précédent s'écrira aussi :

```

POUR TOUT FIC
//Traitement
FIN

```

### d) Rechercher des enregistrements

HLitRecherchePremier(FIC,RUB,"Valeur") : positionne sur le premier enregistrement du fichier de données en faisant une recherche à l'identique de la valeur indiquée

HLitRecherche(FIC,RUB,"Valeur") : permet de faire une recherche générique sur les chaînes de caractères. Cette recherche n'est pas générique sur les entiers, les réels, les dates, les monétaires, ...

Booléen = HTrouve(FIC) : renvoie Vrai si et seulement si l'enregistrement en cours existe et correspond à la valeur recherchée

### *Exemple de recherche simple*

```

HLitRecherchePremier(COMMANDE,NUMCMDE,15234)
SI HTrouve(COMMANDE) ALORS
//Traitement
SINON
INFO("Commande 15234 inexistante !")
FIN

```

### *Recherche avec doublons*

```

//Plusieurs clients ayant comme nom "Martin"
HLitRecherchePremier(CLIENT,NOM,"Martin")
TANTQUE Htrouve(CLIENT)
//Traitement
HLitSuivant(CLIENT)
FIN

```

## e) Filtrage des enregistrements

### *Exemple de code*

On utilise la fonction HFilter :

HFilter(FIC, RUB, BorneMin, BorneMax, "Condition")

Par exemple :

HFilter(VIN, numRegion, "20120101", "20121231", "TTC > 100 et TVA = 5,5")

```
//Pour filtrer les vins de la région sélectionné par
l'utilisateur
SI comboRegion <> "*" ALORS
HFilter(VIN, numRegion, comboRegion)
SINON
//Dans le cas où l'utilisateur souhaite afficher tous les
vins
HDesactiveFiltre(VIN)
FIN
//On met à jour l'affiche
//TableAffiche force la réactualisation
TableAffiche(tableVin)
//Recherche sur la table
HDesactiveFiltre(VIN)
HLitRecherchePremier(VIN, numRegion, 31)
```



### *Remarque : HFilter avec conditions multiples*

Si vous souhaitez utiliser HFilter mais avec plusieurs conditions, voici la syntaxe :

HFilter(Personne,"administrateur=0 et Projet\_idProjet="+Projet.idProjet)

Ce filtrage récupère les utilisateurs non administrateur qui sont sur un projet spécifique (dont on connaît l'identifiant). idProjet se trouve dans la table Projet.



### *Fondamental : N'oubliez pas HDesactiveFiltre une fois le traitement terminé*

Si vous utilisez un HFilter dans votre code et que, par la suite, vous effectuez un autre traitement sur la table (parcours de la table ou autre), alors **n'oubliez surtout pas de désactiver les filtres sur cette table**. En effet, les filtres restent activés sur la table et cela peut être source d'erreur car **si un filtre a été active (par exemple les vins de Bordeaux), vous aurez beau faire des HLitRecherche sur les vins de Midi-Pyrénées, vous n'en trouverez aucun** car le HFilter est activé sur la table.

## f) Exécuter une requête

Si vous disposez d'une requête réalisée via l'éditeur de Requêtes (Clic droit sur "Requêtes"->"Nouvelle requête"), vous pouvez exécuter une requête via :

HExécuteRequête(nomRequête)

Pour exécuter une requête SQL :

HExécuteRequêteSQL("VOTRE CODE SQL")



### *Remarque : Lecture obligatoire pour charger le résultat*

Il faut ensuite charger le résultat via un HLit sur la requête sinon celle-ci ne sera pas chargée.

## 5. Boîtes de dialogue

Le WLanguage dispose de différentes boîtes de dialogue simple qui vont demander des informations à l'utilisateur.

```
OuiNon("Voulez-vous quitter le programme?") // -> Renvoie
les constantes Oui (1)ou Non (0)en fonction de la réponse
de l'utilisateur
OKAnnuler("Supprimer cet élément ?") // -> Affiche une
boite de dialogue avec OK et Annuler. Renvoie Oui si
l'utilisateur clique sur Ok ou Non autrement.
Confirmer("Question ?") // -> Renvoie les constantes Oui ou
Non ou Annuler en fonction de la réponse de l'utilisateur
Erreur("Une erreur s'est produite") // -> Pour afficher un
message d'erreur
Info("Message d'information") // -> Pour afficher un
message d'information à l'utilisateur
```



### Remarque : Créer vos propres boîtes de dialogue

Ce sont des boîtes de dialogue "standard" qui génère généralement une popup javascript. Vous pouvez aussi **utiliser des boîtes de dialogue personnalisées via l'utilisation de la fonction PageAfficheDialogue mais faites attention car cela risque de limiter leur utilisation à du code non AJAX** (ou alors bien penser à créer une version AJAX).

### PageAfficheDialogue et PageFermeDialogue

PageAfficheDialogue est une fonction permettant rapidement de créer une popup simple. En effet, elle prend en paramètre le nom d'une page qu'elle va superposer à la page dans laquelle elle est appelée. Elle s'utilise très simplement :

PageAfficheDialogue(nomPage)

```
// Affiche une page permettant à l'utilisateur de quitter
le site
SI PageAfficheDialogue(PAGE_Sortir) = Vrai ALORS
// Affiche une page de fin
PageAffiche(PAGE_Fin)
// Ferme la session
FinProgramme()
FIN
```



### Remarque : Page appelée

Dans la page appelée, vous devez renvoyer une valeur soit via la fonction RENVOYER soit via PageFermeDialogue(Vrai).



### Complément : Mise en forme de la page appelée

Vous pouvez créer une cellule dans la page appelée pour imiter une vraie boîte de dialogue.

Il suffit d'aller dans "Création->Cellule". Vous pouvez ensuite y mettre les éléments voulus (champs de saisie, boutons, etc) et modifier le fond de la cellule via un clic droit sur la cellule puis dans "Description->Style", vous pouvez y mettre la couleur souhaitée (blanc par exemple).

## 6. Passage de paramètres

### *Dans une page*

Nous avons déjà vu le passage de paramètres dans une fonction. Mais vous pouvez aussi passer des paramètres à une page.

Pour cela, il suffit d'aller dans le code de la page dans la partie déclarations globales, et d'y créer une procédure ayant pour nom le nom de la page.

```
PROCEDURE PageVin (local param est une chaine)
```



### *Complément : Exemple de code : création ou modification d'un élément*

```
SI param = "M" alors
//On charge le formulaire
FichierVersPage()
SINON
//On met le formulaire à zéro
HRAZ(Vin)
FIN
```

```
//Code du clic du bouton Valider
PageVersFichier()
SELON param
CAS "C" :
//Création
HAjoute(Vin)
CAS "M" :
HModifie(Vin)
FIN
PageFermeDialogue()
```

```
//Code du bouton Nouveau (page appelante)
PageAfficheDialogue(PaeVin, "C")
//Pour rafraichir la page
TableAffiche(TableVin)
```

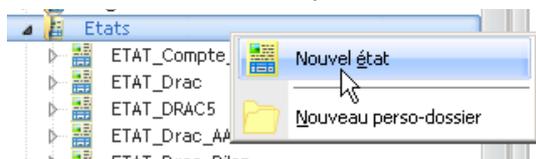
```
//Code du bouton Modifier
PageAfficheDialogue(PageVin, "M")
TableAffiche(TableVin)
```

## 7. Fonctions d'impression

Pour imprimer un élément (PDF ou imprimante), il faut :

- Créer un état
- Ajouter la fonction d'impression côté serveur

La création d'un état va vous permettre de créer et placer les éléments. Un état est un objet générique décrivant les informations du document à imprimer. Vous pourrez y associer les éléments de la base, y ajouter du texte et des photos et gérer la mise en forme de l'ensemble. La création d'un état se fait dans la partie "État" via un clic droit puis sélectionnez "Nouvel état" :



Vous créez ensuite votre état comme voulu.

### *Impression de l'état depuis une page*

Dans le code d'un bouton par exemple, il vous reste à intégrer les éléments pour imprimer l'état (iDestination, iImprimeEtat et FichierAffiche) :

iDestination(Format, CheminComplet) où format peut prendre plusieurs valeurs (iPDF, iHTML, iFax, iImprimante, etc)

**A savoir : iImprimante représente une imprimante connecté au serveur d'applications WebDev et non une imprimante locale.**

iImprimeEtat(nomEtat) génère le fichier en se basant sur l'état précédemment créé FichierAffiche(iDernierFichier(), typeMime) où type mime peut prendre plusieurs valeurs ("application/pdf", "application/jpeg", "text/plain", etc).

iDernierFichier permet de récupérer un objet représentant le dernier fichier généré (dernier résultat de l'appel à la fonction iImprimeEtat).

```
//Exemple d'impression au format PDF
iDestination(iPDF, ComplèteRep(fRepExe())
+Projet.idProjet+"Budget.PDF")
iInitRequêteEtat(ETAT_Drac_Budget,Projet.idProjet)
iImprimeEtat(ETAT_Drac_Budget)
// "Nom souhaité" est le nom final du fichier lors du
téléchargement par l'utilisateur
FichierAffiche(iDernierFichier(),"application/pdf", "Nom
souhaité")
fSupprime(iDernierFichier())
```



### *Attention : Nom unique !*

**Il faut un nom unique du fichier car 2 utilisateurs pourrait lancer l'impression en même temps.** Vous pouvez utiliser les fonctions DateSys(), HeureSys() ou DonneGuid() pour être certain d'obtenir un fichier au nom unique, nécessaire que pour la génération car la fonction FichierAffiche prend en 3ème paramètre le nom souhaité pour le téléchargement.

**N'oubliez pas de supprimer le fichier pour ne pas surcharger le serveur via la fonction fSupprime.**

## 8. Répertoire du serveur

La fonction fRepDonnées renvoie le path (ou chemin) des données du site (contient les données de la base et les données Exe du projet).

La fonction fRepWeb renvoie le path (ou chemin) du site Web à proprement dit (pages HTM, javascript et images).

fRepDonnees (MesDonnees) -> fichier BDD (.FIC, .NDX, etc) et répertoire Exe

└─┬─┬─┬─Refscol31

│ └─┬─┬─┬─Log

fRepWeb (MesSites) -> XXX\_WEB (Page HTM, Javascript, images) et fichiers .AWL

└─┬─┬─┬─REFSCOL\_AVIS

└─┬─┬─REFSCOL\_AVIS\_WEB

└─┬─┬─┬─FR

│ └─┬─┬─┬─res

└─┬─┬─┬─res

└─┬─┬─┬─\_images

│ └─┬─┬─┬─boutons

│ └─┬─┬─┬─cadres

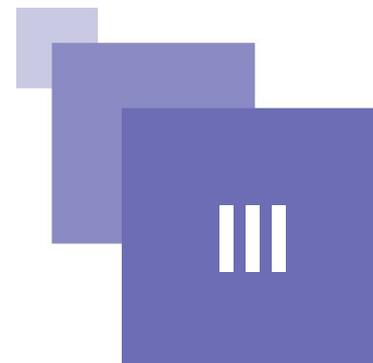
│ └─┬─┬─┬─fonds

│ └─┬─┬─┬─menu

│ └─┬─┬─┬─modeles

└─┬─┬─\_images\_Etats

# Création d'un projet



Créer un projet	69
Gestion de projet	71
RAD	72
GDS	73
Charte de programmation	74
Ambiance du projet	74
Choix de la langue	75
Choix de la base de données	75
Pages et modèles	76
Utilisation d'AJAX	79

Cette partie liste les différentes étapes de création d'un projet sans rentrer dans le détail de chaque élément. Elle permet de se faire une idée des fonctionnalités possibles.

## A. Créer un projet

Pour créer un projet, il suffit de cliquer sur "Nouveau" dans l'onglet "Accueil" puis sélectionner "Dynamique WebDev"

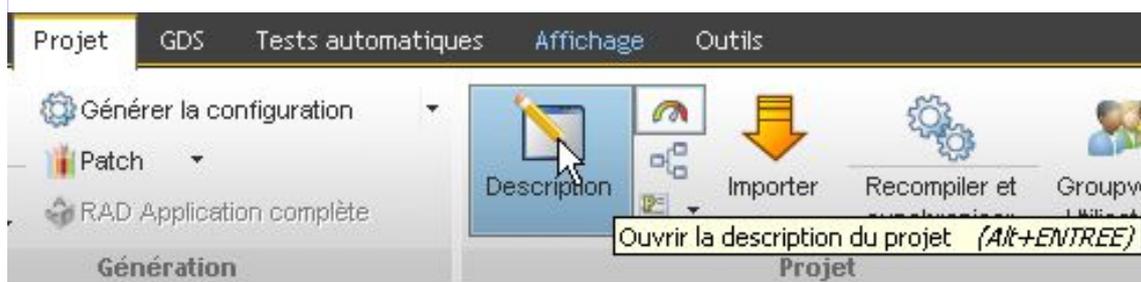


Création projet dynamique



*Remarque : Accéder à la description d'un projet existant*

Il suffit d'aller dans "Projet->Description"



Description du projet

La première fenêtre vous demande de définir le titre de votre projet mais aussi de sélectionner le nouveau répertoire qui va stocker les informations du projet.

Par défaut, le répertoire de tous les projets se trouve dans 'Mes Sites'. **WebDev conseille de conserver cette arborescence.**

Description du projet



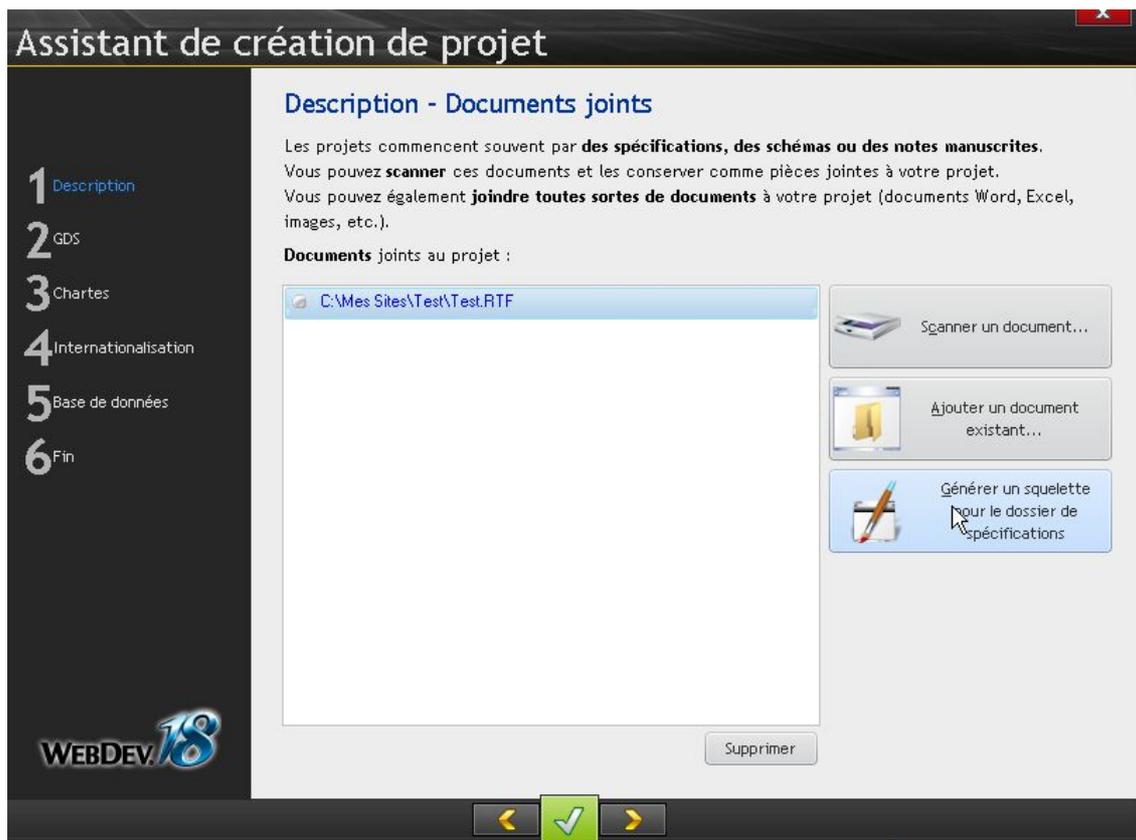
### Remarque

Il est possible de placer ce répertoire sur un serveur de fichier (partage de fichiers, samba, etc) pour le partager avec les différents membres d'une équipe de développement.

Le plus simple, autrement, est d'utiliser le GDS (Gestionnaire De Source).

## B. Gestion de projet

WebDev embarque un outil de suivi du projet vous permettant d'associer, pour chaque projet, les documents voulus (cahier des charges, spécifications fonctionnelles, etc). Cet outil peut être utile si c'est le choix que vous avez fait pour le suivi de vos projets Web. Dans tous les cas, vous pouvez très bien y intégrer tous les éléments décrivant le projet pour que le reste de l'équipe puisse les consulter. Il vous permet aussi de générer un squelette (extrêmement basique) des spécifications de votre application qui peut être édité directement au sein de l'assistant.



Gestion de projet

## C. RAD

### Mode de création

Vous pouvez créer un projet vierge (de base où tout est à faire) ou un projet basé sur un "RAD" (Rappid Application Development) qui est un modèle embarquant tout le nécessaire tel que l'analyse (base de données) et les éléments structurant du modèle.



### Conseil : RAD : pas une solution "magique"

Un RAD a le mérite de pouvoir vous mettre le pied à l'étrier mais il ne saurait être une solution ultime. Tout d'abord, les modèles de RAD sont limités (forum, CMS, Agence de voyage, Location de DVD, SAV, eCommerce, site associatif, galerie photos) et, lorsque généré depuis une analyse existante, incomplet.

Cependant, **vous devriez utiliser les modèles proposés pour analyser et comprendre les analyses associées (base de données) et voir des exemples de code WLangage dans le but de vous approprier les "bonnes pratiques" WebDev.** Vous pourriez très bien partir d'un site associatif et faire évoluer l'analyse, ajouter les pages nécessaires et le code pour créer l'application voulue.

## D. GDS

Vous devrez ensuite indiquer si vous souhaitez inclure ce nouveau projet au GDS (Gestionnaire De Source). Le GDS est l'outil de versioning de WebDev permettant de stocker les éléments du projet dans une base de données.

Comme svn ou git pour un projet symfony par exemple, il est indispensable dans le cas du travail en équipe et très utile dans le cas contraire. **Bref, utilisez systématiquement pour tout nouveau projet important.**

Cela permet de :

- Conserver l'historique des modifications effectuées
- Visualiser les différences entre 2 versions d'un même élément
- Revenir à une version plus ancienne
- Partager des éléments entre plusieurs projets

**Assistant de création de projet**

### Gestionnaire De Sources (GDS)

Le Gestionnaire De Sources (GDS) est un puissant outil de versioning permettant de stocker des projets dans une base de données. Il est fondamental dans le cas du travail en équipe et est également utile si vous travaillez seul. Le GDS permet entre autre :

- de partager des éléments entre plusieurs projets (fenêtres, états, classes, images, composants, ...)
- de conserver l'historique des modifications
- de visualiser les différences entre 2 versions
- de récupérer une version plus ancienne

Voulez-vous utiliser le Gestionnaire De Sources pour votre projet ?

**Oui, utiliser le GDS**  
Je veux utiliser le GDS pour stocker les éléments du projet. [Aide du GDS](#)

**Non, ne pas utiliser le GDS**  
Je ne veux pas utiliser le GDS.

**?** Vous pourrez **modifier** cette option **plus tard** dans la fenêtre de description du projet (Menu Projet..Description du projet, onglet Groupe/GDS).

WEBDEV 18

Ajout projet GDS



### Attention : GDS pas exempt de défauts et de bugs

Le GDS de WebDev ne fonctionne pas du tout comme les outils de versioning standards. **Par exemple, vous ne pourrez pas travailler en même temps sur le même élément d'un projet**, là où les autres outils travaillent ligne par ligne ce qui permet de laisser les développeurs travaillaient sur le même fichier.

Sous le GDS WebDev, le premier qui prend la ressource la bloque jusqu'à ce qu'il est terminé et la libère. C'est seulement une fois libéré que l'autre développeur pourra la bloquer à son tour pour l'éditer. **Pensez donc bien à libérer systématiquement la ressource une fois que vous avez terminé.**

Le GDS de WebDev gère **aussi très mal le versioning de l'analyse (la modélisation de la base de données)**. En effet, pour revenir à une ancienne

version de l'analyse, il va vous falloir basculer vers les anciennes versions plusieurs fichiers, quasiment tous ceux se trouvant dans le répertoire nomprojet.ana

## E. Charte de programmation

- Les **noms de variables, classes, constantes ou fonctions doivent être en français**
- Pas d'accent ou caractères spéciaux dans les noms de variables, classes, constantes ou fonctions
- Les **noms de fonctions ou de méthodes sont composés d'un verbe d'action**, par exemple :  
ajouterEtablissement()  
lireAffectation()  
ecrireEnseignant()  
supprimerProjet()
- Les **noms de variables, classes, méthodes ou fonctions suivent la notation camelCase**, par exemple :  
monProjet  
monProjet.ajouterEtablissement(monEtablissement)
- Les **constantes sont désignées par des noms en majuscule dont les mots sont séparés par tiret-bas**, par exemple :  
MA\_CONSTANTE

## F. Ambiance du projet

L'ambiance du projet est l'aspect graphique de votre application. Par défaut, WebDev en propose plusieurs par défaut avec des palettes de couleurs associées. Cependant, les organisations ayant généralement leur propre charte graphique, il faudra y appliquer des modifications pour changer l'image de fond par exemple.

**Il semble qu'il ne soit pas aisé de créer une ambiance** pour la rajouter dans celles par défaut.



*Conseil : Sélectionnez une ambiance et une palette proche de la charte graphique de votre organisation*

Il est tout à fait possible de modifier la charte graphique de votre application ultérieurement en modifiant le modèle de page et l'image de fond. La seule difficulté réside dans le CSS (choix de la police, de la taille, des couleurs de surbrillance, etc) qui peut être long et fastidieux à modifier.

Le mieux serait donc de **sélectionner une ambiance et une palette de couleurs qui colle au mieux à la charte graphique de votre organisation puis de modifier le modèle de page pour changer l'image de fond et les styles du menu.**

## G. Choix de la langue

Pour les applications multilingues, l'assistant vous demande de sélectionner les langues qui seront utilisées par votre projet. Une multitude d'options (type monétaire, heure, date, etc) sont liées aux langues choisies.

## H. Choix de la base de données

Ce choix permet de sélectionner le type de base de données voulu ainsi que la connexion. Dans le cas d'un nouveau projet, c'est là que WebDev va initialiser l'analyse et les connexions vers les bases de données utilisées par cette analyse.

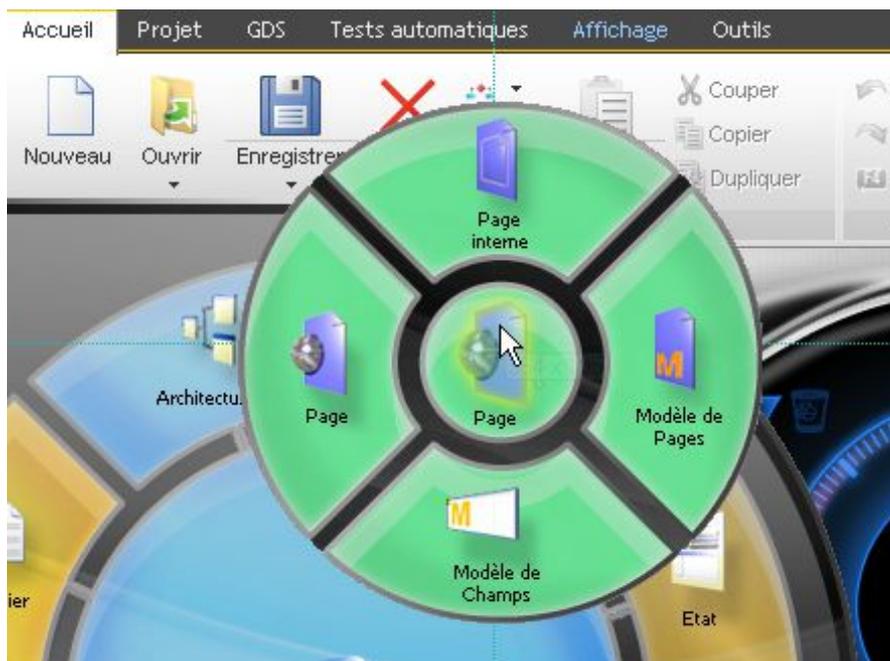
Vous pourrez aussi sélectionner une base de données existante pour, par exemple, réécrire un projet existant sous WebDev.



Projet : création BDD

## I. Pages et modèles

Cliquez sur "Nouveau" puis "Page"



Projet : création page

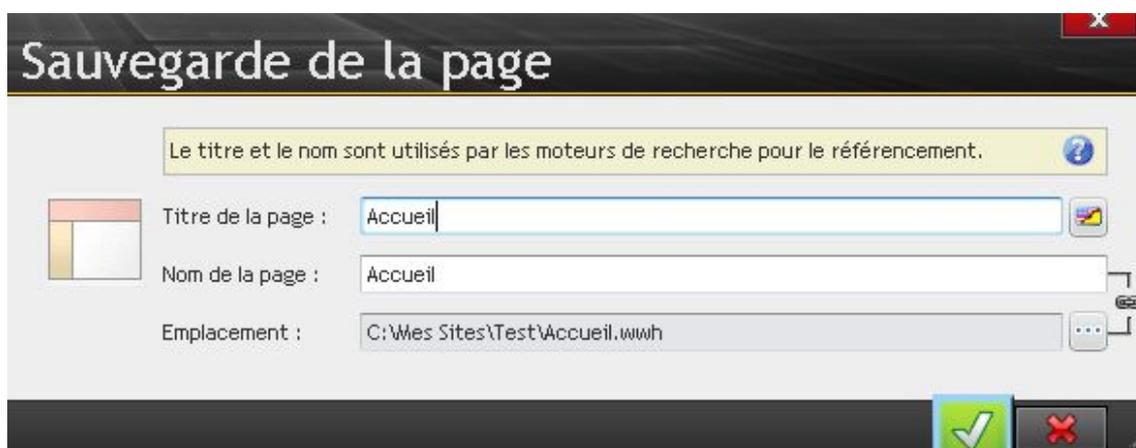
Vous accédez alors à un menu indiquant plusieurs types de pages possibles si vous utilisez le RAD ou une ambiance pour votre projet. Souvent, vous utilisez des pages simples pour lesquels vous modifierez l'aspect au fur et à mesure.



### Conseil : Utilisez un modèle de page

Nous en parlons un peu plus loin dans cette partie mais pensez à utiliser un modèle de page. Cela a pour avantage de pouvoir partager divers éléments entre les mêmes pages (code, popups, etc) et de définir une charte graphique commune à toutes vos pages. Il vous suffira alors de modifier le modèle pour que vos pages se mettent à jour (moyennant une ré-actualisation de ces pages).

Lorsque vous sauvegarderez pour la première fois votre page, un menu apparaîtra afin d'indiquer le nom de la page et son emplacement.



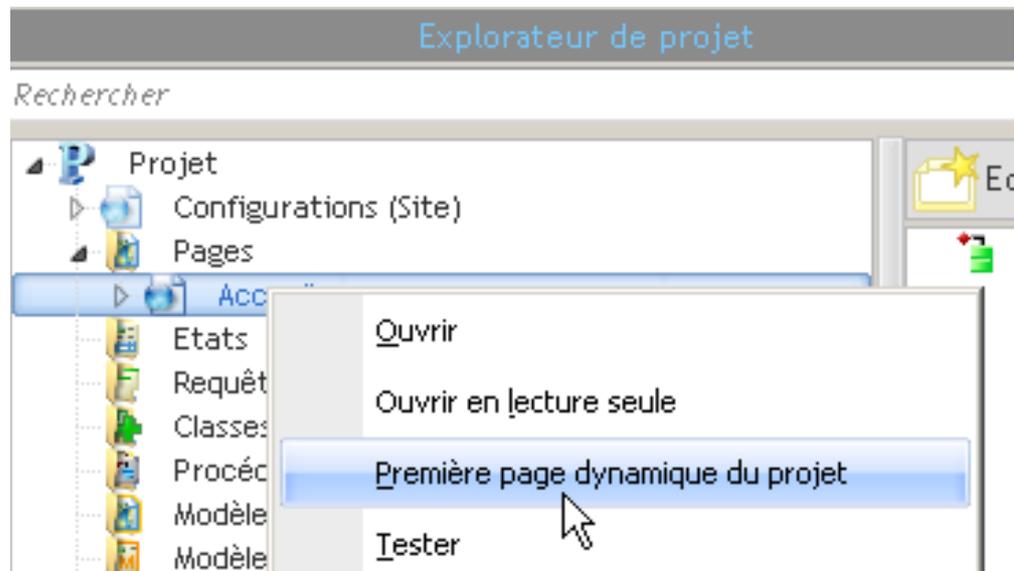
Page : nom et emplacement

## 1. Notion de première page dynamique

La première page dynamique du projet est, comme son nom l'indique, la page principale auquel accéderont les utilisateurs. Cela peut être une page de connexion dans le cas d'un site protégé, une page publique de présentation, etc.

Celle-ci est obligatoire.

Pour indiquer qu'une page est la première page dynamique du projet, il suffit de faire un clic droit et de l'indiquer :



Projet : première page dynamique

## 2. Modèle de page

Une page particulière mais importante est le "modèle de page". Celui-ci vous permet de définir une page globale sur laquelle toutes les pages vont se baser. Cela présente plusieurs avantages :

- Possibilité d'y inclure du code partagé par toutes vos pages
- Définition d'une charte graphique commune
- Ajout de popups partagées par toutes les pages

Le modèle de page se crée comme une page standard.



*Remarque : Une modification nécessite de réactualiser les pages dépendantes*

Lorsque vous modifiez le modèle de page, il faut absolument réactualiser les pages qui en dépendent.

### a) Modèles utilisés par une page

Ouvrez la page voulue puis allez dans l'onglet page puis dans "Modèles -> Liste des modèles utilisés"



Image 4 Page : liste des modèles

## b) Mis à jour depuis le modèle

Lors de la modification d'un modèle de pages, afin de mettre toutes les pages dépendantes à jour, il suffit de cliquer sur "Modèles -> Mettre à jour les pages associées"

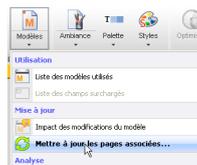


Image 5 MAJ pages depuis modèle

## J. Utilisation d'AJAX

Au sein de votre projet, vous serez tenté d'utiliser **AJAX**<sup>27</sup>. En effet, WebDev propose de modifier les composants potentiels (un bouton par exemple) en simple appel AJAX via un simple clic de souris.

**Cependant, sachez que cela ne s'applique pas à toutes les situations et qu'il faut donc l'utiliser à bon escient.**

Clic de **BTN\_OK\_Enseignant ( CELL\_Enseignant ) (serveur) \*** **AJAX Activé**  
 COMBO Domaine..Valeur="" OU COMBO Dispositif..Valeur="" OU SAI Intitul

WebDev : action - désactivation d'AJAX

Plusieurs limites existent :

- Le serveur ne renvoie qu'une partie des données
- Ne gère pas les dépendances d'affichage
- Des fonctions ne sont pas supportées en AJAX



### Remarque : AJAX programmé

L'**AJAX programmé**<sup>28</sup> permet de lancer des requêtes AJAX "à la demande" côté navigateur et exécutant une requête vers le serveur et de récupérer et traiter l'information renvoyée. Cela se fait via utilisation de la fonction **AJAXExecute** ou **AJAXExecuteAsynchrone**. Cette dernière a le mérite de ne pas bloquer les traitements et donc les actions de l'utilisateur mais cela peut s'avérer problématique.

```
// AJAX supporté sur le navigateur en cours ?
SI AJAXDisponible() = Vrai ALORS
// Exécution de la procédure serveur "MaProc"
// Dès que le résultat de la procédure serveur "MaProc"
sera disponible,
// la procédure navigateur "ProcRésultat" sera exécutée
Res est un entier
Res = AJAXExecuteAsynchrone(MaProc, ProcRésultat,
NomClient, Bonus)
// Les traitements suivants s'exécutent sans attendre le
résultat
...
SINON
// Traitements sans utilisation de AJAX
```

27 - [http://fr.wikipedia.org/wiki/Ajax\\_%28informatique%29](http://fr.wikipedia.org/wiki/Ajax_%28informatique%29)

28 - <http://doc.pcsoft.fr/fr-FR/?3055103&name=ajax-programme&q=AJAX>

...  
FIN

# Administration et débogage

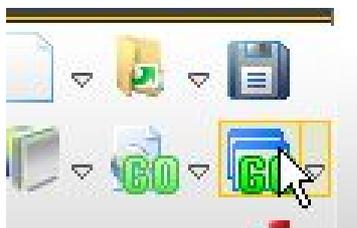
## IV

Débogage	81
Déploiement d'un site	88
GDS	90

## A. Débogage

### 1. Débogage "simple"

On peut déboguer un programme de façon tout à fait standard en y mettant des points d'arrêt puis en faisant un "go" sur le projet.



WebDev : "Go" pour lancer un projet



WebDev : Débogage - point d'arrêt

Une fois exécuté, on passe alors en mode débogage, le programme s'arrêtera sur le premier point d'arrêt rencontré. On pourra alors y voir l'état des variables, faire du pas à pas détaillée, ajouter le contenu des éléments à visualiser, ajouter une expression, etc. Bref un environnement de débogage standard comme ceux proposés dans Visual Studio par exemple.



WebDev : Menu de débogage



### Complément : Ajout d'assertion

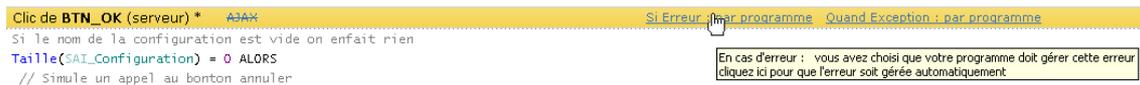
Une assertion est une expression à évaluer pour laquelle on est certain qu'elle devrait logiquement être correct.

Par exemple, on peut considérer que, lors de la suppression d'un client, il faut avoir sélectionné le client, on pourrait donc le vérifier via un dbgAssertion :

```
//Si HTrouve renvoie faux alors le message "Client non
trouvé !" apparait
dbgAssertion(HTrouve, "Client non trouvé !")
```

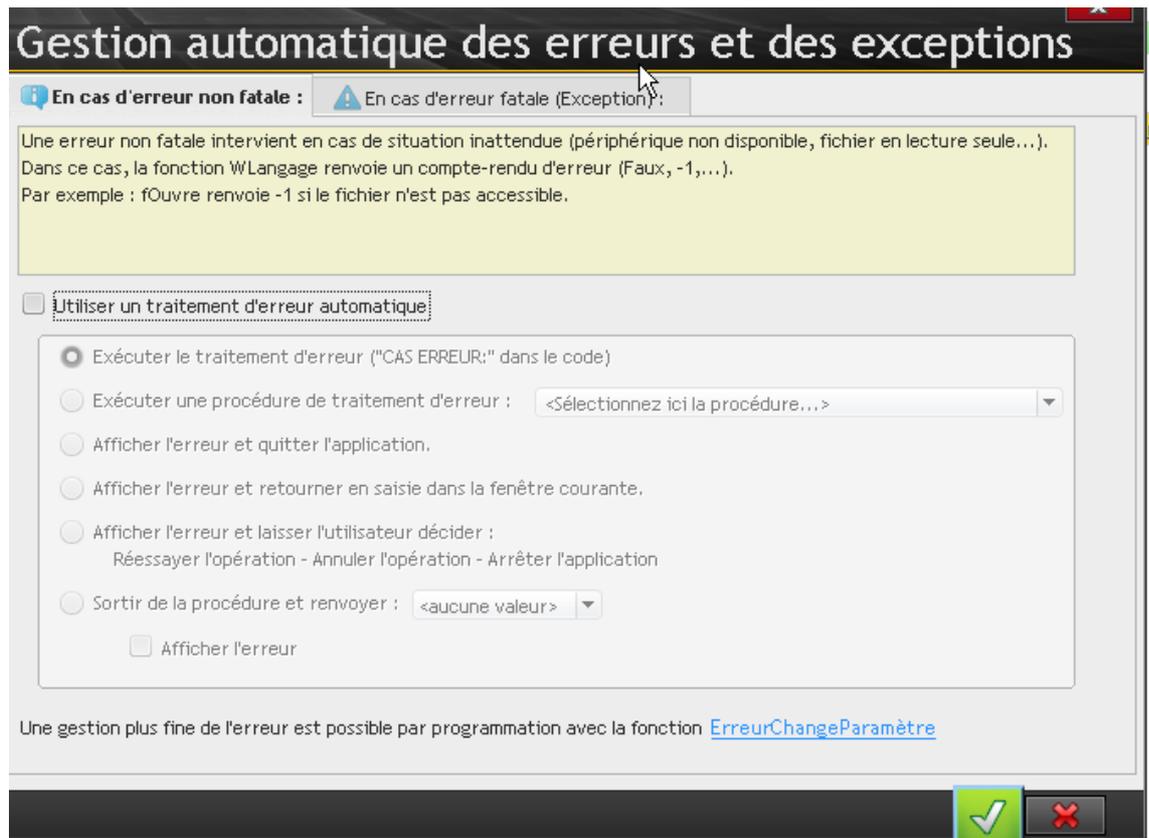
#### a) Gestion des erreurs et exceptions

Pour gérer une erreur ou une exception dans votre code, il faut cliquer sur la partie "Si Erreur : par programme" et/ou "Quand Exception : par programme".



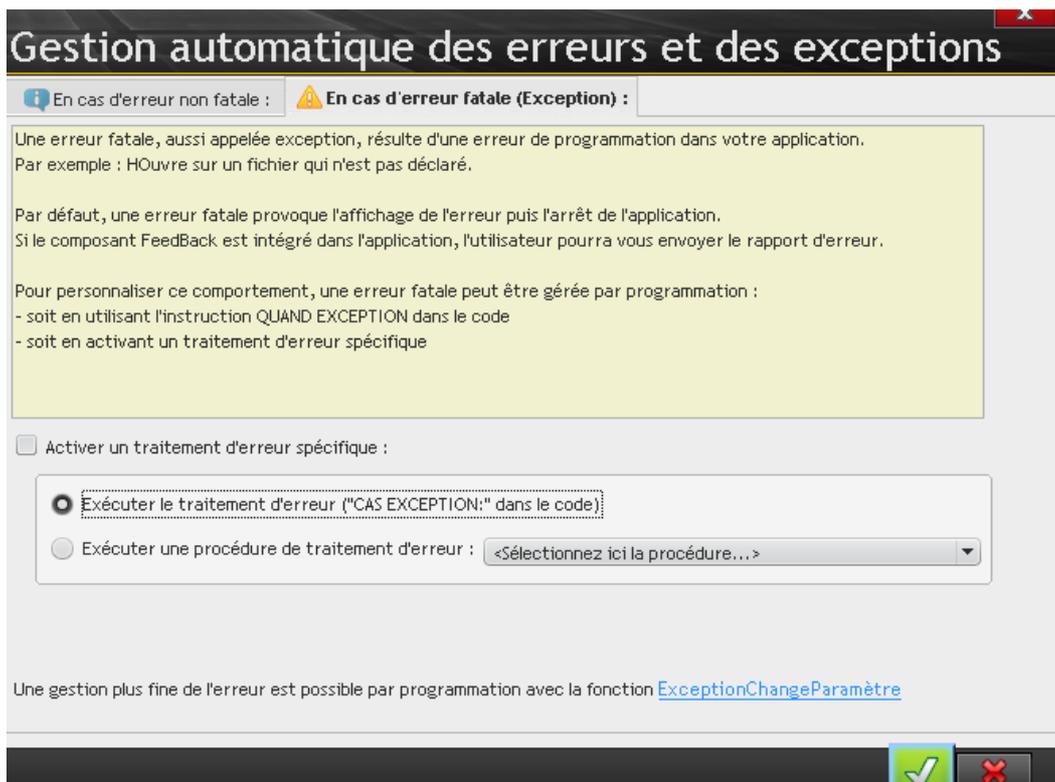
WebDev : Erreur et Exception

Vous pourrez alors indiquer plusieurs "stratégies" comme déclencher une procédure de traitement, afficher un message à l'utilisateur, etc.



### WebDev : Gestion des erreurs

En cas d'exception, étant bloquant par définition, les options sont moins fournies soit vous utilisez du code de traitement via "CAS EXCEPTION :" soit vous déclenchez le code d'une procédure.



WebDev : Gestion des exceptions



### Remarque : La gestion des erreurs ou exception s'imbrique

Il faut savoir qu'une erreur non traitée par le bloc ou la partie l'ayant déclenchée sera alors traitée au niveau plus haut et ainsi de suite.

**Ainsi vous pouvez implémenter une gestion des erreurs au niveau du code du projet pour être sûr qu'elles soient traitées.**



### Conseil : Pensez à écrire un traitement

Il est évident que des erreurs surviendront (un programme qui fonctionne est un programme bogué qui s'ignore), pensez donc à récupérer les erreurs et exceptions pour les traiter.

Voici un exemple de code accessible dans le code du projet de Arts et culture :

```
CAS EXCEPTION:
HFermeConnexion(gcnxBDD)
HTransactionAnnule()
Erreur("Une erreur fatale a été détectée. Merci de
réessayer ultérieurement. Si le problème persiste, veuillez
écrire à dsi.dsi-da@ac-toulouse.fr",
ExceptionInfo(errComplet), "", "Arrêt de la procédure.")
SI gbModeDebug ALORS
fichierInfoErreur est une chaîne =
ComplèteRep(fRepDonnées())+"Exception_"+DateHeureSys()
+".txt"
fSauveTexte(fichierInfoErreur, ExceptionInfo(errComplet))
fichierDump est une chaîne = ComplèteRep(fRepDonnées())
+"Exception_Dump_"+DateHeureSys()+".wdump"
```

```

dbgSauveDumpDébogage (fichierDump)
fichiersJointS est un tableau de chaînes =
[fichierInfoErreur,fichierDump, gsFichierAudit]
EnvoiEmailErreur (ExceptionInfo (errComplet), fichiersJointS)
FIN

```

Une procédure de gestion des erreurs simple déclenché par le projet en cas d'erreur permet d'annuler la transaction et de prévenir l'utilisateur qu'une erreur s'est produite :

```

PROCEDURE GestionErreur()
HTransactionAnnule()
Info("Une erreur est survenue. Merci de réessayer
ultérieurement. Si le problème persiste, veuillez écrire à
dsi.dsi-da@ac-toulouse.fr", ExceptionInfo (errComplet), "",
"Arrêt de la procédure.")

```

## 2. Audit

Il existe 2 types d'audit : statique et dynamique.

Un audit statique permet de corriger les erreurs les plus fréquentes : code mort, détection des éléments orphelins, erreurs d'orthographe, nettoyage

Un audit dynamique va permettre d'analyser les variables et leur état au cours d'un parcours du site Web, il nécessite donc d'effectuer une "simulation" d'utilisation du site Web et le compte rendu de cet audit vous permettra de voir l'état des variables et l'état d'exécution des données pour déceler des incohérences dans le projet.

### *Lancer un audit*

Pour lancer un audit statique ou dynamique, cela se passe depuis la page d'accueil du projet, au niveau du tableau de bord, dans la partie haute à droite :



*WebDev : Audit*

L'audit statique génère un rapport qui vous permettra de corriger tout un ensemble de problèmes vous permettant d'optimiser votre application.



WebDev : Rapport audit statique

L'audit dynamique, comme son nom l'indique, se fait en lançant un "go minutieux" du projet qui va inspecter les variables, contrôler l'état de l'application, etc dans le but de remonter les incohérences. Il nécessitera donc une analyse plus approfondie des résultats pour déceler les erreurs éventuelles.



WebDev : Lancement audit dynamique

Date	Heure	Événement
17/07/2013	15:20:05	Dump mémoire automatique en fin de test minutieux

### WebDev : Rapport audit dynamique



#### Remarque : Audit dynamique : historique

Vous pouvez aussi accéder aux précédents audits dynamiques effectués. Un bon découpage serait de tester chaque fonctionnalité de l'application, une à une, pour analyser et consolider les résultats que l'on présenterait aux développeurs en vue d'une correction.



#### Conseil : Bonnes pratiques : effectuer le test de l'audit dynamique à une personne n'ayant pas écrit le code et ne pas sous-estimer la durée des tests de débogage

Il semble plus efficace et cohérent de **faire effectuer le test soit par une personne extérieure au projet soit par un développeur n'ayant pas contribué au code de la fonctionnalité contrôlée** par l'audit dynamique.

**D'autre part, la détection et la correction des incohérences prennent du temps, il faut donc s'assurer d'avoir "chiffré" convenablement la durée liée à ces tâches.**



#### Remarque : Audit d'infrastructure : un service de collecte

L'audit d'infrastructure s'appuie sur un outil tiers chargé de récupérer une ensemble de données via des "collecteurs" et un serveur spécifique WebDev appelé SILO. Cela permet de récupérer un ensemble d'indicateurs se situant à mi-chemin entre nagios et piwik. Cet audit est plus pertinent sur de grosses organisations qui ferait un usage massif des outils de PC SOFT.

#### Audit dynamique : lancement par programmation

On peut déclencher un audit dynamique par programmation. On pourrait imaginer implémenter une fonction automatique de déclenchement dans l'application en mode debug pour envoyer automatiquement le rapport (par courriel par exemple) aux développeurs dès que l'on détecte un bug dans l'application.

**En effet, l'intérêt de l'audit dynamique est que les développeurs pourront alors voir l'état des variables au fur et à mesure (état chronologique) avant la survenue de l'erreur.**

```
//Pour déclencher un audit dynamique
```

```
//L'audit générera un fichier .waudit que l'on peut ensuite
analyser sous WebDev
//en faisant ouvrir avec ou via un glisser/déposer
directement dans l'interface
dbgActiveAudit("nomFichierAudit")
```

## B. Déploiement d'un site



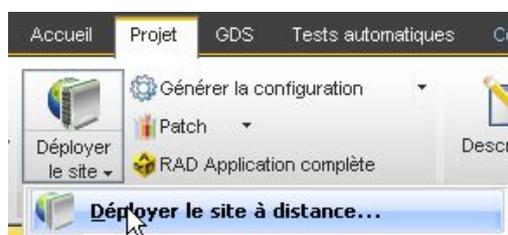
**Conseil : Vérifiez la connexion à la BDD avant déploiement !**

Avant tout déploiement, vérifiez dans le code du projet que vous avez bien sélectionné la base de données voulues. Si cette information se trouve en dur dans le code, pensez à le changer pour basculer vers la bonne connexion.

Par exemple, pour ArtsetCulture :

```
//VARIABLE A METTRE A VRAI SUR LA PROD
//ET FAUX SI EN TEST : détermine le choix de la bdd (test
ou réelle) et l'envoi de mails à la création d'un projet
(DSI-DA/pmietlicki =>
l'enseignant+établissement/pmietlicki)
gbEnProd est un booléen = Vrai
SI EnModeTest() ALORS
gbEnProd = Faux // <-- à commenter (tout simplement) pour
utiliser la base réelle
FIN
```

Pour déployer un site, il suffit d'aller dans l'onglet "Projet" puis "Déployer le site" :



WebDev : Déployer le site

La première étape consiste à construire la bibliothèque (une archive) de déploiement, l'outil va détecter automatiquement les éléments à importer, vous pouvez directement valider via la coche du milieu

Suite à la construction de l'archive, le déploiement à proprement dit va commencer. Une fenêtre s'affiche pour vous demander où vous souhaitez le déployer, l'option général étant sur un serveur d'applications WebDev, vous sélectionnez donc "Déployer le site WebDev sur un Serveur D'application WebDev distant".

Ensuite, il va vous falloir indiquer les différentes informations d'accès au serveur (nom dns ou adresse IP), le compte de déploiement WebDev (créé au préalable sur le serveur) et le compte FTP WebDev (compte ayant les droits d'envoyer l'archive précédemment créé sur un serveur FTP intermédiaire installé sur le serveur d'applications WebDev). Toutes ces informations se trouvent dans le keypass de dsi-da disponible dans :

G:\DA\2-coffre\_fort\Prod\_machines\_autres.kdbx

Mettre dans "Adresse du serveur", l'adresse IP du serveur : celle de production est,

à l'heure où j'écris ces lignes, 172.29.240.15 (au 17/07/2013)

Compte WebDev : le compte administrateur de l'application sur le serveur d'applications WebDev. Normalement, nous en avons 1 par application (bonnes pratiques) -> par exemple, PROD\_ARTSETCULTURE (voir les mots de passe dans le keypass dans Prod\_machines\_autres)

Compte FTP : pour déployer les fichiers qui seront ensuite installés par le serveur d'applications (voir nom utilisateur et mots de passe dans le keypass dans Prod\_machines\_autres) -> par exemple, dsi-da



*Attention : Mettre l'adresse IP si déploiement en production et le nom DNS si déploiement en test*

**C'est l'adresse IP qu'il faut indiquer pour la production** car la configuration du virtual host a été faite par adresse IP sur le frontal au lieu du nom DNS (contrainte technique équipe système).

**Pour la base de test**, n'étant pas configuré depuis un frontal via virtualhost, **il faut indiquer le nom dns (prodwd.in.ac-toulouse.fr)**.

Valider les paramètres entrés en cliquant sur "Tester les paramètres" puis "Suivant"

### *Nom du site de déploiement*

Nom de déploiement du site : en fonction du type de déploiement, par exemple "test\_artsetculture" ou "Art\_et\_culture", c'est ce qui se trouve à la fin de l'URL c'est à dire le chemin de l'application sur le serveur.

Par exemple, in fine, on y accédera via [http://prodwd.in.ac-toulouse.fr/test\\_artsetculture](http://prodwd.in.ac-toulouse.fr/test_artsetculture) (si accès direct interne) ou <https://artsetculture.ac-toulouse.fr> (si accès via reverse proxy).

Choisissez le nom du profil voulu (test\_artsetculture ou prod\_artsetculture), ce que vous souhaitez, cela permet d'enregistrer les paramètres saisis précédemment pour ne pas avoir à tout ressaisir la prochaine fois.

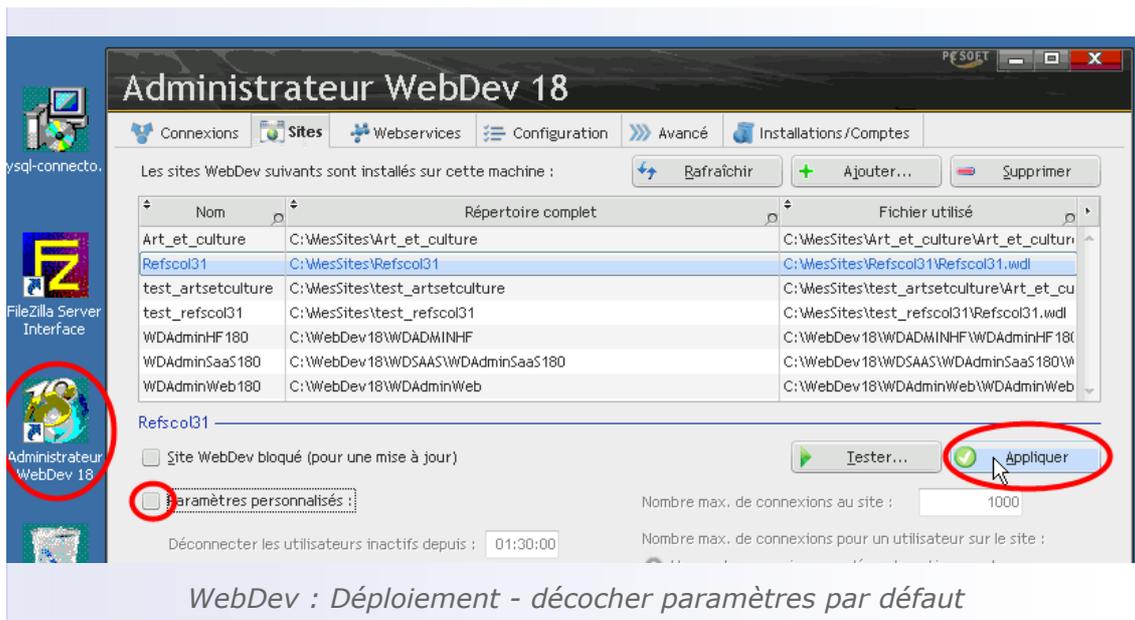
**Laissez les paramètres par défaut pour la suite et valider en cliquant sur la coche du milieu.**



*Attention : Suite au déploiement, se connecter en RDP sur la machine prodwd et ouvrir l'administrateur WebDev*

Comme WebDev indique des paramètres par défaut de déploiement (nombre de connexion, durée d'une connexion, etc) qui ne sont pas forcément pertinentes. Il faut donc aller dans l'Administrateur WebDev sur le serveur d'applications (via connexion RDP), dans "Administrateur WebDev 18" puis décocher "Paramètres personnalisés".

Un "bug existe" qui le remet par défaut dès qu'on effectue un déploiement alors qu'on veut garder les paramètres par défaut du serveur.



## C. GDS

Comme indiqué précédemment, le GDS apporte des fonctionnalités dont il serait dommage de se passer même s'il est loin d'être aussi bien qu'un subversion ou qu'un github.

Il n'est aussi pas exempt de bugs et nous indiquons ici une méthode pour "débloquer" l'accès à une ressource qui serait bloqué en édition par un membre du projet qui serait parti en vacances ou parti à Nouméa.

Il est possible, lorsque l'on travaille à plusieurs sur un projet, qu'un de vos collaborateurs bloquent des pages ou ressources du projet puis parte en congé... Dans ce cas, **vous allez devoir débloquer ces ressources pour pouvoir travailler dessus.**

Cependant, avec le GDS (Gestionnaire De Sources) de WebDev, ce n'est pas forcément intuitif. En fait, il va falloir dans **"Administrateur du GDS", le nom est trompeur** puisqu'il nous fait croire que l'on est déjà connecté en tant qu'administrateur mais ce n'est pas forcément le cas, qui se trouve dans "GDS" -> "Gérer" :



WebDev : Administrateur du gestionnaire de sources

Puis connectez vous en tant qu'**administrateur de la base de données.**

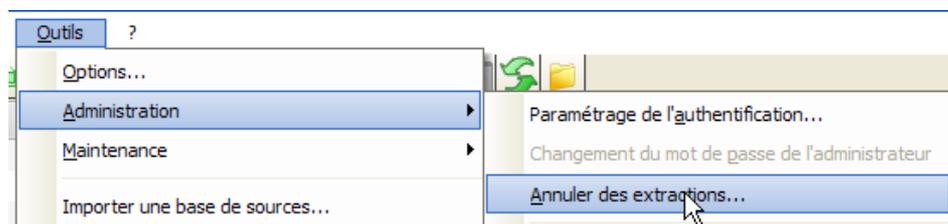
Pour se connecter à la base du GDS en tant qu'administrateur, il suffit d'utiliser l'option **"Fichier .. Connexion à une base de sources"** de l'administrateur du GDS.

Dans la fenêtre qui s'ouvre :

- Indiquez les caractéristiques de la base de sources utilisée.
- Spécifiez le nom et le mot de passe de l'administrateur de la base :
- Si vous utilisez une base GDS au format HyperFileSQL Classic, par défaut l'administrateur est "Admin" (sans mot de passe).
- Si vous utilisez une base GDS au format HyperFileSQL Client/Serveur, par défaut l'administrateur correspond à l'utilisateur de type "Administrateur".

**Remarque : La connexion à la base de données du GDS en tant qu'administrateur n'est pas la même chose que "Devenir maître sur le projet". Pour réaliser des manipulations d'administration, il est nécessaire de se connecter en tant qu'administrateur.**

Vous allez alors avoir accès au menu "**Outils**"->"**Administration**", l'**option qui nous intéresse ici est "Annuler des extractions"** (pensez à sélectionner, au préalable dans l'arborescence, le projet pour lequel vous souhaitez le faire) :



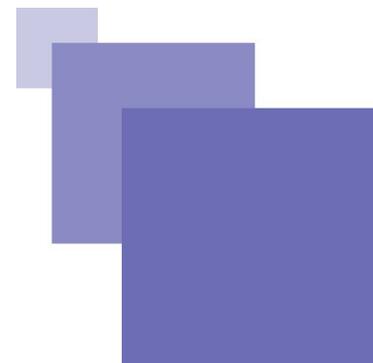
*WebDev : GDS - Annuler des extractions*



**Remarque : Administrateur de la base de données HyperfileSQL et non l'administrateur du GDS**

Pour accéder à ce fameux menu, il faut se connecter avec le compte administrateur du SGBD hyperfile qui héberge la base de données du GDS et non celui du compte administrateur du GDS.

# Conclusion



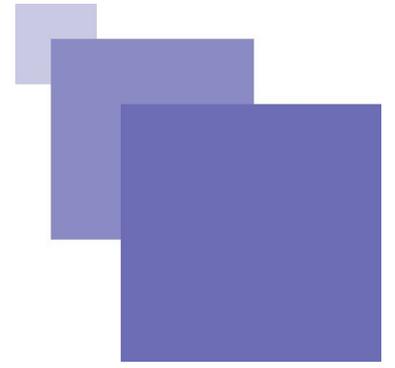
Ce module vous aura présenté les différentes fonctionnalités de WebDev, les bases et les fonctions pour pouvoir réaliser rapidement son premier site Web. L'interface, complexe au premier abord, est très riche et il vous faudra manipuler l'outil pour pouvoir "fouiller" un peu et le maîtriser.

Sachez qu'une multitude d'assistants sont présents et vous permettront de découvrir à votre rythme ce que vous pouvez faire avec WebDev.

**N'hésitez pas à solliciter le support PC SOFT**, accessible depuis l'aide (onglet "Accueil", partie "Aide en ligne", bouton "Aide" puis "Support technique") si vous rencontrez un problème ou un bug. Dans le cas d'un bug important, vous devrez préparer un projet permettant de reproduire le bug qu'il faudra leur envoyer au format ZIP.

Bonne continuation et bon courage.

# Index



*webdev, initiation, wlangage..... p.8*  
*perfectionnement,*

