# Generation Configuration Melodie

**ANALYSE ORGANIQUE**

## Références

* “AF Generation Config Melodie.docx”
* “ExpressionBesoinGénérationConfiguration.doc”
* “AO Gestion Licence Melodie.doc”

Table des matières

[Generation Configuration Melodie 1](#_Toc454524489)

[Références 1](#_Toc454524490)

[gestion interface 2](#_Toc454524491)

[Principe général 2](#_Toc454524492)

[Classement des objets de l'IHM 2](#_Toc454524493)

[Saisies utilisateurs 3](#_Toc454524494)

[Fonctionnement général 3](#_Toc454524495)

[Objets utilisés 3](#_Toc454524496)

[TabControl (composé de TabPage) 3](#_Toc454524497)

[TextBox 5](#_Toc454524498)

[Radio Button 5](#_Toc454524499)

[FolderBrowserDialog 6](#_Toc454524500)

[ComboBox 6](#_Toc454524501)

[Button 7](#_Toc454524502)

[Accès fichier de configuration 8](#_Toc454524503)

[Structure fichier de configuration 8](#_Toc454524504)

[Fonctions accès fichier de configuration 9](#_Toc454524505)

[Chargement classeur Excel 10](#_Toc454524506)

[Structure Dossier configuration 10](#_Toc454524507)

[Traitement en background 10](#_Toc454524508)

[La classe AccesDossierExcel 12](#_Toc454524509)

[Principe et appels 12](#_Toc454524510)

[Propriétés 12](#_Toc454524511)

[Méthodes 12](#_Toc454524512)

[Chargement Base de Données 16](#_Toc454524513)

[Structure Dossier configuration 16](#_Toc454524514)

[Traitement en background 16](#_Toc454524515)

[Comparaisons d’éléments et de configuration 18](#_Toc454524516)

[Comparaison de configuration 18](#_Toc454524517)

[Comparaison d’éléments 18](#_Toc454524518)

[Les classes et méthodes du dossier de configuration de la Base de Données 18](#_Toc454524519)

[Les constructeurs 18](#_Toc454524520)

[Les méthodes de comparaison 21](#_Toc454524521)

[Traitement des erreurs 25](#_Toc454524522)

[Traitement en background 25](#_Toc454524523)

[Creation des elements 26](#_Toc454524524)

[Hiérarchie des éléments 26](#_Toc454524525)

[Hiérarchie des ajouts : 26](#_Toc454524526)

[Liste des éléments 28](#_Toc454524527)

[Bouton MàJ libellés 29](#_Toc454524528)

[Objets concernés 29](#_Toc454524529)

[Principe 29](#_Toc454524530)

[Création base de donnees 29](#_Toc454524531)

[Principe 29](#_Toc454524532)

[Etat des lieux code existant 29](#_Toc454524533)

[contrôle coherence configuration 30](#_Toc454524534)

[Traitement des modeles de modes de marche 43](#_Toc454524535)

[Chargement Base de Données 44](#_Toc454524536)

[Structure Dossier configuration 44](#_Toc454524537)

[Annexes 45](#_Toc454524538)

[Diagramme de classe 45](#_Toc454524539)

## gestion interface

L’application permet à l’utilisateur de générer une configuration Mélodie Net à partir d’un classeur de configuration Excel. L’utilisateur peut aussi mettre à jour des éléments individuels de la configuration.

### Principe général

#### Classement des objets de l'IHM

Les objets de l'interface sont répartis selon leur usage

1. Sélection / Connexion base de donnée (Zone "générale").
2. Sélection/Ouverture du classeur Excel contenant la configuration à traiter (Zone "générale").
3. Création/MAJ des éléments de la configuration à proprement parler (Zones "Modes de marche" et "Process"

#### Saisies utilisateurs

Seules les chaines de caractères permettant de définir les connexions indispensables (base de données et classeur de configuration) sont à saisir par l'utilisateur (utilisation de TextBox). Le contenu de ces TextBox est conservé dans un fichier au format XML et reproposés à chaque démarrage du programme. Les autres données sont déduites du contenu du classeur de configuration et sont sélectionnées à l'aide de ComboBox.

#### Fonctionnement général

Au démarrage, le programme commence par lire le contenu des TextBox dans le fichier XML.

Remarque :

Lors du clic sur certains boutons de l'IHM

* concernant la base de donnée (Connecter, Créer, Purger) : appel de la fonction d'écriture des 4 composantes de l'accès à la base de données dans le fichier XML
* concernant le classeur Excel (Ouvrir) : appel de la fonction d'écriture du chemin d'accès au classeur Excel dans le fichier XML

Ensuite le fonctionnement du programme ne dépend plus que des actions utilisateur : le programme fonctionne donc sous le modèle évènementiel. Les règles générales suivantes sont applicables

* les objets de la troisièmes catégorie ne sont utilisables que si une base de données est connectée et un classeur de configuration est chargé.
  + le code des événements "Clic" des boutons de la première catégorie positionnent "rbConnecte" à vrai.
  + le code de l'événement "clic" du bouton ouvrir positionne "rbCharge" à vrai.
  + le code de l'événement "Changed" de chacun des 2 boutons radio (rbConnecte et rbCharge) teste l'état de l'autre bouton. Si les 2 boutons sont à vrai, les listes des comboBox sont initialisées et les boutons passent à l'état "Enabled"
* La zone "Process"est organisée d'une manière hiérarchique (voir § "Hiérarchie des éléments") : la sélection d'une valeur dans une comboBox (événement IndexChanged) déclenche la mmise à jour des listes déroulantes des comboBox de rang inférieur dans la hiérarchie.

### Objets utilisés

#### TabControl (composé de TabPage)

Utilisation :

* Séparation des deux fonctionnalités de l’application en deux onglets

Evènement :

* Simple navigation entre deux onglets

Etat initial :

* Enable et visible

|  |  |  |
| --- | --- | --- |
| Nom TabControl | Utilisation | Remarque |
| tcTableauDeControl | Totalité du tableau, contenant deux onglets |  |

|  |  |  |
| --- | --- | --- |
| Nom TabPage (onglet du TabControl) | Utilisation | Remarque |
| tpGeneration | Regroupe toute la partie de génération de configuration de l’application |  |
| tpComparaison | Regroupe toute la partie de comparaison de l’application |  |

#### TextBox

Utilisation :

* Uniquement dans la zone générale
* Saisir les paramètres de connexions à une base de données
* Saisir le chemin d'accès au classeur Excel

Evènement :

* Pas d'action : les changements sont pris en compte sur les boutons

Etat initial :

* Ecrire les paramètres par défaut dans les textbox (lireClefsServeur, lireClefClasseurExcel)

|  |  |  |
| --- | --- | --- |
| **Nom Textbox** | **Utilisation** | **Remarque** |
| tbServeur | Textbox Serveur |  |
| tbBaseDeDonnees | Textbox Base de données |  |
| tbUtilisateur | Textbox Utilisateur |  |
| tbPassword | Textbox Mot de Passe | Texte caché avec des \* |
| tbClasseurConfiguration | Textbox Dossier Excel | Rempli automatiquement avec parcourir |

#### Radio Button

Utilisation :

* Donner l'état de la connexion à la base de donnée ou classeur Excel
* Donner l’état du chargement de la Base de Données

Evènement :

* Aucun : juste une indication à l'utilisateur

Etat Initial :

* disabled
* non coché

Objets :

* rbConnecte
* rbCharge
* rbChargeeBDD

#### FolderBrowserDialog

Utilisation :

* Chercher le dossier Excel dans l'arborescence

Evènement :

* Récupérer le chemin sur un fichier sélectionné au format .xls

Initialisation :

* Paramètre de recherche : .xls

#### ComboBox

Utilisation :

* Sélection d’un élément dans le classeur Excel pour ajout
* Sélection d’un élément déjà existant pour mise à jour ou ajout de ses descendants
* Sélection d’un élément dans la Base de Donnée pour comparaison

Evènement :

* Sur changement récupérer la clef de l’objet à ajouter ou à mettre à jour ou à comparer
* Initialiser les comboBox de rang inférieur dans la hiérachie (par exemple combobox "Opération" et "Paramètre famille" quand un process est sélectionné)
* le bouton associé devient enabled
* Méthode associée : cbXxxxIndexChanged pour la comboBox cbXxxx

Initialisation :

* disabled

|  |  |
| --- | --- |
| **Nom Combobox** | **Responsable initialisation** |
| cbModeleModesMarche | btnOuvrir |
| cbProcess | btnOuvrir |
| cbOpération | cbProcess |
| cbParametreFamille | cbProcess |
| cbParametreSystem | cbProcess |
| cbTracabiliteSystem | cbProcess |
| cbConsigneSystem | cbProcess |
| cbMesureSystem | cbProcess |
| cbParametre | cbOperation |
| cbTracabilite | cbOperation |
| cbConsigne | cbOperation |
| cbMesure | cbOperation |
| cbLigne | cbProcess |
| cbStation | cbLigne |
| cbFicheReference | cbLigne |
| cbFicheProduit | cbLigne |
| cbFichierTexte | cbLigne |
| cbFichePalette | cbLigne |
| cbProcessBDD | btnCharger |
| cbLigneBDD | cbProcessBDD |
| cbModeleDeModesDeMarcheBDD | btnCharger |

#### Button

Utilisation :

* Gérer les options de la base de données (Purge, Initialisation)
* Gérer l'accès au classeur de configuration
* Ajouter une configuration complète
* Ajouter un élément a la configuration existante
* Charger une Base de Donnée
* Comparer des objets
* Comparer une configuration

Evènement :

* Exécuter l’action sur un On click.

Etat initial :

* Boutons de la zone générale : enabled
* Boutons des zones "Mode de Marche" et zone "Process" : disabled
* Bouton « Charger » de l’onglet « Comparaison » : enabled
* Autres boutons de l’onglet « Comparaison » : disabled

|  |  |  |
| --- | --- | --- |
| **Nom Bouton** | **Méthode** | **Responsable état enabled** |
| btnConnecter | Classe Requete methode connect |  |
| btnCreer | CreerBaseMelodieNet |  |
| btnPurger | ? |  |
| btnParcourir |  |  |
| btnOuvrir | DossierConfiguration | btnParcourir|tbClasseurConfiguration |
| btnGenererConfig | GenererConfigMelodie | btnOuvrir & btnConnecter |
| btnModeMarche | MajModesMarche | btnOuvrir & btnConnecter |
| btnProcess | AjouterProcess | btnOuvrir & btnConnecter |
| btnOperation | AjouterOperation | cbProcess |
| btnParametreFamille | AjouterParametreFamille | cbProcess |
| btnParametreSystem | AjouterParametreSystem | cbProcess |
| btnTracabiliteSystem | AjouterTracabiliteSystem | cbProcess |
| btnConsigneSystem | AjouterConsigneSystem | cbProcess |
| btnMesureSystem | AjouterMesureSystem | cbProcess |
| btnParametre | AjouterParametre | cbOperation |
| btnTracabilite | AjouterTracabilite | cbOperation |
| btnConsigne | AjouterConsigne | cbOperation |
| btnMesure | AjouterMesure | cbOperation |
| btnLigne | AjouterLigne | cbOperation |
| btnFicheService | MajFicheService | cbLigne |
| btnFicheSuivi | MajFicheSuivi | cbLigne |
| btnDefautsProcess | MajDefautsProcess | cbLigne |
| btnDefautsProduits | MajDefautsProduits | cbLigne |
| btnStation | AjouterStation | cbLigne |
| btnFicheReference | MajFicheReference | cbLigne |
| btnFicheProduit | MajFicheProduit | cbLigne |
| btnFichierTexte | MajFichierTexte | cbLigne |
| btnFichePalette | MajFichePalette | cbLigne |
| btnQuitter | FermerClasseurExcel |  |
| btnCharger | btnChargerClick (charge la base de données) | btnConnecter |
| btnComparerProcess | btnComparerProcessClick (compare le Process de la BDD sélectionné avec TOUS les Process du classeur Excel chargé) | cbProcessBDD |
| btnComparerLigne | btnComparerLigneClick (compare la Ligne de la BDD sélectionnée avec TOUTES les Lignes du classeur Excel chargé) | cbLigneBDD |
| btnComparerModesMarche | btnComparerModesMarcheClick (compare le Mode de Marche de la BDD sélectionné avec TOUS les Modes de Marche du classeur Excel chargé) | cbModeleDeModesDeMarcheBDD |
| btnComparerConfiguration | btnComparerConfigurationClick (compare la configuration de la BDD chargée avec la configuration du classeur Excel chargé) | btnCharger |
| btnInitialiserProfils | btnInitialiserProfilsClick() (initialise les utilisateurs d’une base de données) |  |
| btnNettoyerGlobalCust | btnNettoyerGlobalCustClick() (efface les « valeurs fantômes » de la table GLOBAL\_CUST) |  |

### Accès fichier de configuration

#### Structure fichier de configuration

2 sections

<accesServeur> avec 4 clefs

serveur

baseDeDonnées

utilisateur

password

<accesClasseur Excel> avec 1 clef

cheminAccesExcel

#### Fonctions accès fichier de configuration

1 classe AccesFichierXML.cs avec les chaines en propriétés publiques

* string serveur
* string base
* string utilisateur
* string password
* string cheminAccesExcel

2 méthodes d'accès en écriture ( à appeler après avoir mis à jour les propriétés ci-dessus)

ecrireClefsServeur()

ecrireClefClasseurExcel()

pas de méthode en écriture, le constructeur suffit, car la lecture du fichier de configuration n'est nécessaire qu'au lancement du programme.

## Chargement classeur Excel

### Structure Dossier configuration

Le contenu du classeur Excel est stocké dans la classe DossierConfiguration. C'est le constructeur de cette classe qui appelle les constructeurs des classes de représentation des objets de MelodieNet hiérachiquement inférieur.

Ces constructeurs, quand ils doivent accéder au contenu du classeur Excel font appel aux méthodes de la classe AccesClasseurExcel. C'est pourquoi le constructeur du DossierConfiguration crée un objet de la classe AccesClasseurExcel.

Le schéma suivant représente la hiérachie des classes du DossierConfiguration. (Ce schéma est disponible en UML dans les annexes)

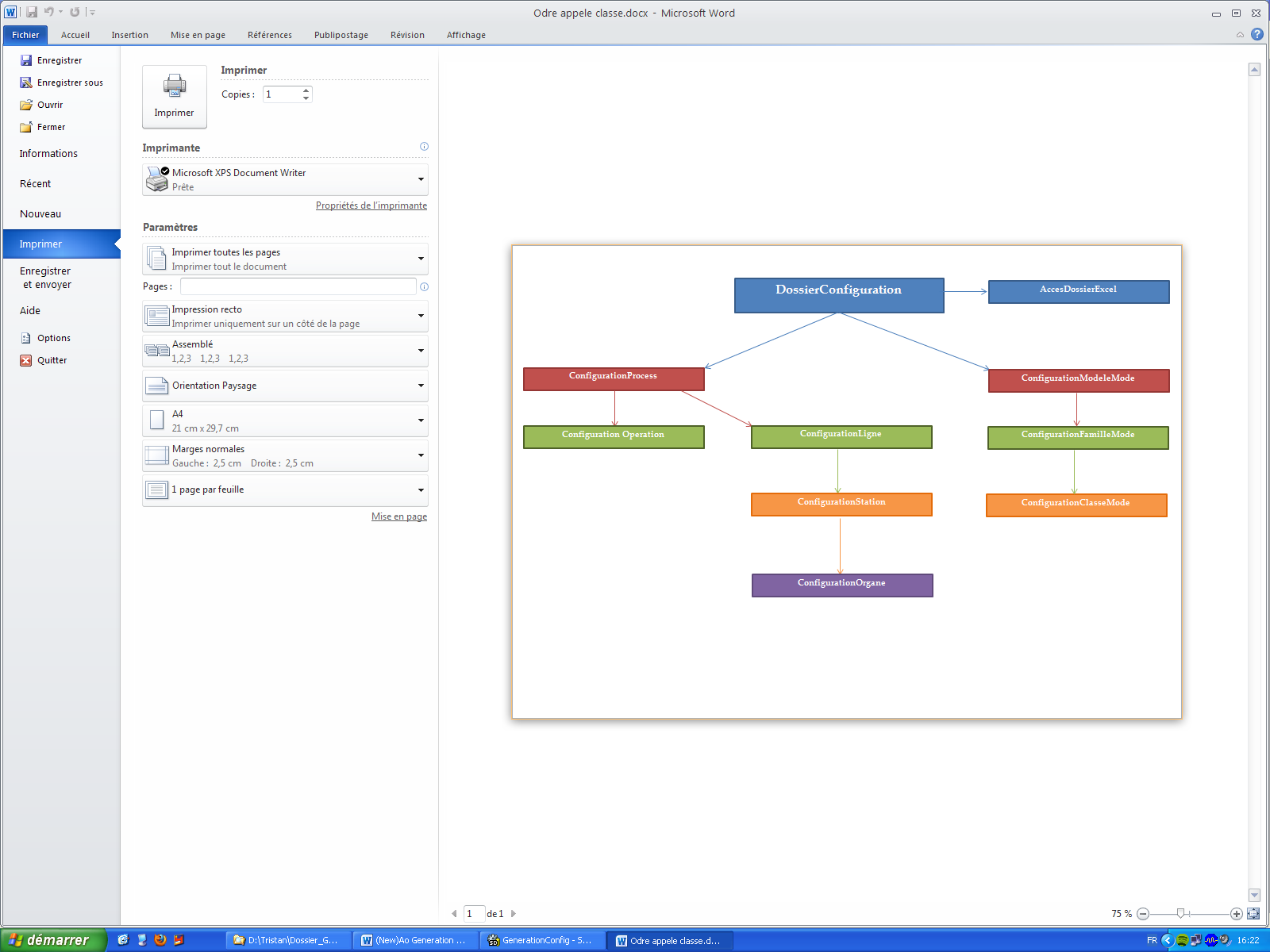
### Traitement en background

Dans la méthode " btnOuvrirClick", le BackGroundWorker "bgwChargementDossier" est lancé.

L'appel au constructeur est fait dans la méthode "bgwChargementDossier\_DoWork".

A la fin du chargement, le traitement du retour est quant à lui traité dans "bgwChargementDossierRunWorkerCompleted". C'est depuis cette méthode qu'est lancé "ControlerCoherenceConfiguration". Le traitement des retours y est également fait.

Enchainement des classes



### La classe AccesDossierExcel

#### Principe et appels

La classe AccesDossierExcel est appelée par toutes les classes du dossier de configuration.

Elle permet d’aller chercher des informations dans le classeur de configuration

#### Propriétés

Dans la classe AccesDossierExcel nous trouvons toutes les propriétés relatives à l’ouverture et le parcours du classeur de configuration.

Sont présents :

Le chemin du dossier  
L’application Excel  
Le classeur de configuration  
L’onglet courant

La liste des champs et des onglets

La liste des erreurs

#### Méthodes

Les méthodes sont listées dans le tableau suivant :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Méthode | Description | Paramètres | Sortie | Tests associés |
| ListerOnglet | Liste les onglets du dossier Excel | Aucuns | 0 si aucunes erreurs  1 si la procédure a recontré des erreurs | **ContenuListOnglet** |
| ListerChampsNommes | Liste les champs només dans l’onglet passé en paramètre | -Le nom de l’onglet dont on désire lister les champs nommés | 0 si aucunes erreurs  1 si la procédure a recontré des erreurs | **ContenuListChamps** |
| SelectionnerOnglet | Selectionne l’onglet dans ongletCourant | -Le nom de l’onglet à selectionner | Aucunes | **ContenuOngletSelectionne** |
| TraiterPlageParametres | Récupère les valeurs d’une plage paramètre | -Nom de la plage paramètre  La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuParamètresOperations**  **NombreParametresAjoutesOK**  **ChampsNumeriquesInvalides** |
| TraiterPlageTracabilité | Récupère les valeurs d’une plage de tracabilités | -Nom de la plage de tracabilités  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuTracabilitéOperations**  **NombreTracabilitésAjoutesOK**  **ChampsNumeriquesInvalides** |
| TraiterPlageConsignes | Récupère les valeurs d’une plage de consignes | -Nom de la plage de consignes  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuConsignesOperations NombreConsignesAjoutesOK**  **ConsigneInfOuSupErroné**  **ChampsNumeriquesInvalides** |
| TraiterPlageParametresFamille | Récupère les valeurs d’une plage de Paramètres famille | -Nom de la plage de paramètre famille  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuParametresFamille**  **NombreParametresFamilleAjoutesOK**  **ChampsNumeriquesInvalides** |
| TraiterPlageServeurOPC | Récupère les valeurs d’une plage de serveurs OPC | -Nom de la plage des serveurs OPC  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuServeursOPC**  **ChampsNumeriquesInvalides** |
| TraiterPlageGroupeOPC | Récupère les valeurs d’une plage de  groupe OPC | -Nom de la plage des groupe OPC  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuGroupeOPC**  **ChampsNumeriquesInvalides** |
| TraiterPlageMesure | Récupère les valeurs d’une plage de  Mesures | -Nom de la plage de mesures  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuMesuresOperations**  **NombreMesuresAjoutesOK**  **ChampsNumeriquesInvalides** |
| LireValeurChamp | Recupère la valeur d’un champ | -Le nom du champ dont on désire la valeur.  -La valeur par défaut de se champ si il n’est pas référencé | La valeur du champ si il est défini, la valeur par defaut si on ne l’a pas trouvé | **ContenueLireValeurChamps**  **ChampInvalideLireValeurChamps**  **ChampNullLireValeurChamps** |
| FermerClasseur | Ferme le classeur utilisé | Aucuns | Aucunes | **Aucuns** |
| TesterAppartenanceRange | Verifie qu’une cellule appartien à une plage de données | Une cellule  Une plage de données | Vrai si la cellule appartient a la plage  Faux sinon | **MethodeAppartenanceRange** |
| TesterAppartenanceMesureConsigne | Vérifie si les consignes figurants dans les mesures sont bien définis | -Une liste de mesures  -Une liste de consignes | Vrai si les consignes sont bien définies  Faux sinon | **MethodeAppartenanceMesuresConsignes** |
| TesterExistenceServeurGroupeOPC | Vérifie si les serveurs figurants dans le Groupe OPC sont bien définis | -La liste de serveurs.  -La liste des groupe OPC | Vrai si les serveurs sont bien définis  Faux sinon | **MethodeExistanceServeurGroupeOPC** |
| TraiterPlageListeOP | Récupère les valeurs d’une plage de  Liste d’OP | -Nom de la plage de la liste d’OP  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuPlageListOP**  **ChampsNumeriquesInvalides** |
| TraiterPlageDefauts | Récupère les valeurs d’une plage de  Défaut(Cartons jaunes,oranges,defauts procédés …) | -Nom de la plage des defauts  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuePlageDefaut**  **NombreDefautsAjoutesOK**  **ChampsNumeriquesInvalides** |
| TraiterPlageExceptions | Récupère les valeurs d’une plage d’exception | -Nom de la plage des exception  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuePlageExceptions**  **NombreExceptionsAjoutesOK**  **ChampsNumeriquesInvalides** |
| TraiterPlageEchangeVariables | Récupère les valeurs d’une plage dans les fiches (Entetes,Parametres Systematiques, Paramètres) | -Nom de la plage d’Echange Variable  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuEchangeVariables**  **NombreEchangeVariablesAjoutesOK**  **ChampsNumeriquesInvalides** |
| TraiterPlageVariablesFichierTexte | Récupère les valeurs d’une plage de Variables d’un onglet Texte | -Nom de la plage de Variables  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuVariablesFichierTexte**  **NombreVariablesTexteFichierAjoutesOK**  **ChampsNumeriquesInvalides** |
| TraiterPlageModes | Récupère les valeurs d’une plage de modes | -Nom de la plage de modes  -La liste où ajouter les données | Nombre de lignes ajoutés ou 0 si erreurs | **ContenuModes**  **NombreModesAjoutesOK**  **ChampsNumeriquesInvalides** |

## Chargement Base de Données

### Structure Dossier configuration

Le dossier Configuration de la Base de Données est quasi-similaire à celui du classeur Excel chargé. En effet, le dossier configuration de la Base de Données reprend les mêmes classes que celui du classeur Excel, mis à part que l’intitulé de ces classes se termine par « BDD » de manière à les distinguer. Dans ces classes, nous ne retrouverons que deux méthodes : le constructeur, ainsi qu’une méthode qui servira à la comparaison des éléments concernés.

### Traitement en background

Dans la méthode « btnChargerClick », le BackGroundWorker « bgwChargementBDD » est lancé. Sa fonction est de traiter le chargement de la Base de Données en arrière-plan. Pour cela, on fait appel au constructeur de la classe « DossierConfigurationBDD » dans la méthode « bgwChargementBDD\_DoWork », du BackGroundWorker cité précédemment. Dans cette méthode, nous veillons également à bloquer les éléments de l’IHM de façon à ce que l’utilisateur ne puisse effectuer d’action pendant le chargement de la Base de Données. A la fin du chargement, le BackGroundWorker fait appel à la méthode « bgwChargementBDDRunWorkerCompleted » ; qui a pour fonction de réactiver les éléments de l’IHM.

Hiérarchie succincte des classes du package ConfigurationMelodieNetBDD

ConfigurationLigneBDD

ConfigurationStationBDD

ConfigurationClasseModesBDD

ConfigurationModeleModesBDD

ConfigurationProcessBDD

ConfigurationFamilleModeBDD

ConfigurationOperatioBDD

DossierConfigurationBDD

## Comparaisons d’éléments et de configuration

La fonction de l’onglet « Comparaison » est d’effectuer des comparaisons entre des éléments précis de la Base de Données par rapport à ceux d’un classeur Excel chargé, ou bien entre la configuration de la Base de Données et celle du classeur Excel en question.

### 

Il est possible de comparerPour cela, le lle-ci La méthode de comparaison d’une classe va appeler, de manière arborescente, les méthodes de comparaisons des classes qui dépendent d’elle, ainsi s’effectue la comparaison.

Le

### Comparaison d’éléments

Il est possible de comparer des Process, des Lignes et encore des Modes de Marche de la Base de Données, par rapport aux mêmes éléments d’un classeur Excel. Pour comparer ces éléments, il faut les sélectionner dans les listes déroulantes prévues à cet effet, et cliquer sur le bouton de comparaison qui leur correspond. Lorsque l’utilisateur effectue ce procédé, la méthode « comparer*N* », pour une comparaison de l’élément *N* de la Base de Données avec le même élément du classeur Excel, est appelée et effectue la comparaison.

Cette méthode retourne 0 si les éléments comparés sont identiques ou -1 s’ils diffèrent.

## Les classes et méthodes du dossier de configuration de la Base de Données

### Les constructeurs

Le principe des constructeurs des classes composant le package de configuration de la Base de Données est de parcourir la table mère (celle que l’on veut instancier), puis, en suivant l’arborescence qu’elle implique, instancier et parcourir ces classes fille, et ainsi de suite. Cet algorithme est répété tant que l’arborescence des classes filles n’est pas terminé.

On instancie chaque classe avec une clé qui leur est unique (un identifiant), qui sera ensuite utilisé, dans des requêtes SQL, pour récupérer les vraies propriétés des classes dans la Base de Données.

Dans chacune des classe, une liste d’erreur est passée en paramètre. Cette liste est définie dans DossierConfigurationBDD et sera donc remplie par les classes. Lorsqu’elles rencontreront une erreur, elles génèreront un message d’erreur et l’ajouteront à la liste passée en paramètre.

|  |  |  |
| --- | --- | --- |
| Classe | Constructeur | Paramètres |
| ConfigurationCelluleBDD | ConfigurationCelluleBDD | int numCellule, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationCibleReferenceBDD | ConfigurationCibleReferenceBDD | int numeroCible, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationClasseModesBDD | ConfigurationClasseModesBDD | int idClasse, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationConsigneBDD | ConfigurationConsigneBDD | int idConsigne, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationDefautProcedeBDD | ConfigurationDefautProcedeBDD | int code, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationDefautProduitBDD | ConfigurationDefautProduitBDD | int code, cdeGravite, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationEchangeSuiviBDD | ConfigurationEchangeSuiviBDD | int idEchange, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationEchangeVariableBDD | ConfigurationEchangeVariableBDD | int idEchange, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationFamilleModesBDD | ConfigurationFamilleModesBDD | int idFamille, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationFicheHorodatageBDD | ConfigurationFicheHorodatageBDD | int numFicheHorodatage, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationFichePaletteBDD | ConfigurationFichePaletteBDD | int numFichePalette, int langue1, int langue2, Requetes requete, DossierConfiguration configurationExcel |
| ConfigurationFicheProduitBDD | ConfigurationFicheProduitBDD | int numFicheProduit, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationFicheReferenceBDD | ConfigurationFicheReferenceBDD | int numFicheReference, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationFicheVisuDynBDD | ConfigurationFicheVisuDynBDD | int numFicheVisuDyn, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationFichierTexteBDD | ConfigurationFichierTexteBDD | int numFichierTexte, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationFonctionsServiceBDD | ConfigurationFonctionsServiceBDD | int numFonctionService, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationGroupeOPCBDD | ConfigurationGroupeOPCBDD | int numGroupeOPC, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationLigneBDD | ConfigurationLigneBDD | int numLigne, int numProcess, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationMesureBDD | ConfigurationMesureBDD | int idVar, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationModeleModesBDD | ConfigurationModeleModesBDD | int numero, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationModeMarcheBDD | ConfigurationModeMarcheBDD | int idMode, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationOperationBDD | ConfigurationOperationBDD | int numOperation, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationParametreBDD | ConfigurationParametreBDD | int idVar, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationParametreFamilleBDD | ConfigurationParametreFamilleBDD | int idVar, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationProcessBDD | ConfigurationProcessBDD | int numero, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationServeurOPCBDD | ConfigurationServeurOPCBDD | int numServeurOPC, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationStationBDD | ConfigurationStationBDD | int numStation, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationSuiviEquipementBDD | ConfigurationSuiviEquipementBDD | int numLigne, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationTracabiliteBDD | ConfigurationTracabiliteBDD | int idVar, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationVariableFichierTexteBDD | ConfigurationVariableFichierTexteBDD | int idVariableFichierTexte, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| ConfigurationVisuDynBDD | ConfigurationVisuDynBDD | int idVisuDyn, int langue1, int langue2, Requetes requete, List<String> listeErreursSQL |
| DossierConfigurationBDD | ConfigurationBDD | Requetes requete, DossierConfiguratio configurationExcel |

### Les méthodes de comparaison

Le principe des méthodes de comparaison des lasses composant le package de configuration de la Base de Données est de parcourir le classeur Excel à la recherche de l’élément qui a fait appel à sa méthode de comparaison, puis de le comparer avec l’élément trouvé si la recherche est un succès. Si cette première étape est un succès, le programme recherche si l’arborescence de la classe mère de la Base de Données existe dans le classeur, puis il effectue l’inverse. Ces vérifications sont faites grâce à un champ non nul de chaque classe.

|  |  |  |  |
| --- | --- | --- | --- |
| Classe | Méthode de Comparaison | Paramètres | Remarques |
| ConfigurationCelluleBDD | ComparerCelluleExcelBDD | ConfigurationCellule celluleExcel, List<String> listeErreursComparaison |  |
| ConfigurationCibleReferenceBDD | ComparerCibleReferenceExcelBDD | ConfigurationCibleReference cibleExcel, List<String> listeErreursComparaison |  |
| ConfigurationClasseModesBDD | ComparerClasseModesExcelBDD | ConfigurationClasseModes classeExcel,List<String> listeErreursComparaison |  |
| ConfigurationConsigneBDD | ComparerConsigneExcelBDD | ConfigurationConsigne consigneExcel ,List<String> listeErreursComparaison |  |
| ConfigurationDefautProcedeBDD | ComparerDefautProcedeExcelBDD | ConfigurationDefaut defautExcel, List<String> listeErreursComparaison |  |
| ConfigurationDefautProduitBDD | ComparerDefautProduitExcelBDD | ConfigurationDefaut defautExcel, List<String> listeErreursComparaison |  |
| ConfigurationEchangeSuiviBDD | ComparerEchangeSuiviExcelBDD | ConfigurationEchengeSuivi echangesuiviExcel, int numVariable, List<String> listeErreursComparaison | \*\*le numVariable permet de savoir d’où vient le suivi d’échange |
| ConfigurationEchangeVariableBDD | ComparerEchangeVariableExcelBDD | ConfigurationEchangeVariable echangeVariableExcel, int numObjet, int typeObjet, List<String> listeErreursComparaison | \*le numObjet permet de savoir d’où vient la variable d’échange  \*\*\*le typeObjet permet de savoir la nature de la variable d’échange |
| ConfigurationExceptionBDD | ComparerExceptionExcelBDD | ConfigurationException exceptionExcel, List<String> listeErreursComparaison |  |
| ConfigurationFamilleModesBDD | ComparerFamilleModesExcelBDD | ConfigurationFamilleModes familleExcel,List<String> listeErreursComparaison |  |
| ConfigurationFicheHorodatageBDD | ComparerFicheHorodatageExcelBDD | ConfigurationFicheHorodatage ficheHorodatageExcel, List<String> listeErreursComparaison |  |
| ConfigurationFichePaletteBDD | ComparerFichePaletteExcelBDD | ConfigurationFichePalette, fichePaletteExcel, List<String> listeErreursComparaison |  |
| ConfigurationFicheProduitBDD | ComparerFicheProduitExcelBDD | ConfiguratioFicheProduit ficheProduitExcel, List<String> listeErreursComparaison |  |
| ConfigurationFicheReferenceBDD | ComparerFicheReferenceExcelBDD | ConfigurationFicheReference ficheReferenceExcel, List<String> listeErreursComparaison |  |
| ConfigurationFicheVisuDynBDD | ComparerFicheVisuDynExcelBDD | ConfigurationFicheVisuDyn ficheVisuDynEcvel, List<String> listeErreursComparaison |  |
| ConfigurationFichierTexteBDD | ComparerFichierTexteExcelBDD | ConfigurationFichierTexte fichierTexteExcel, List<String> listeErreursComparaison |  |
| ConfigurationFonctionsServiceBDD | ComparerFonctionsServiceExcelBDD | ConfigurationFonctionsService fonctionsServiceExcel, List<String> listeErreursComparaison |  |
| ConfigurationGroupeOPCBDD | ComparerGroupeOPCExcelBDD | ConfigurationGroupeOPC groupeOPCExcel, List<String> listeErreursComparaison |  |
| ConfigurationLigneBDD | ComparerLigneExcelBDD | int numLigne, ConfigurationLigne ligneExcel, List<String> listeErreursComparaison |  |
| ConfigurationMesureBDD | ComparerMesureExcelBDD | ConfigurationMesure mesureExcel, List<String> listeErreursComparaison |  |
| ConfigurationModeleModesBDD | ComparerModelModesExcelBDD | int numero, DossierConfiguration configurationExcel, List<String> listeErreursComparaison |  |
| ConfigurationModeMarcheBDD | ComparerModeMarcheExcelBDD | ConfigurationModeMarche modeMarcheExcel, List<String> listeErreursComparaison |  |
| ConfigurationOperationBDD | ComparerOperationExcelBDD | ConfigurationOperation operationExcel, List<String> listeErreursComparaison |  |
| ConfigurationParametreBDD | ComparerParametreExcelBDD | ConfigurationParametre parametreExcel, List<String> listeErreursComparaison |  |
| ConfigurationParametreFamilleBDD | ComparerParametreFamilleExcelBDD | ConfigurationParametreFamille parametreFamilleExcel, List<String> listeErreursComparaison |  |
| ConfigurationProcessBDD | ComparerProcessExcelBDD | int numProcess, DossierConfiguration configurationExcel, List<String> listeErreursComparaison |  |
| ConfigurationServeurOPCBDD | ComparerServeurOPCExcelBDD | ConfigurationServeurOPC serveurOPCExcel, List<String> listeErreursComparaison |  |
| ConfigurationStationBDD | ComparerStationExcelBDD | ConfigurationStation stationExcel, List<String> listeErreursComparaison |  |
| ConfigurationSuiviEquipementBDD | ComparerSuiviEquipementExcelBDD | ConfigurationSuiviEquipement suiviEquipemetExcel, List<String> listeErreursComparaison |  |
| ConfigurationTracabiliteBDD | ComparerTracabiliteExcelBDD | ConfigurationTracabilite tracabiliteExcel, List<String> listeErreursComparaison |  |
| ConfigurationVariableFichierTexteBDD | Pas de méthode de comparaison car, comme rien ne rend une variable de fichier texte unique, on doit faire la comparaison dans « ConfigurationFichierTexteBDD » |  |  |
| ConfigurationVisuDynBDD | ComparerVisuDynExcelBDD | ConfigurerVisuDyn visuDynExcel, List<String> listeErreursComparaison |  |
| DossierConfigurationBDD | ComparerDossierConfigurationExcelBDD | DossierConfiguration dossierconfigurationExcel, List<String> listeErreursComparaison |  |

Ajouté récemment

\*Remarque : Concernant le paramètre numObjet de la méthode de comparaison de la classe « EchangeVariableBDD », il s’agit d’un repère qui nous permet de savoir d’où vient la variable d’échange courante, pour pouvoir générer des messages d’erreur précis (1 🡪 Fiche VisuDyn, 2 🡪 Fiche Produit, 3 🡪 Fiche Palette, 4 🡪 Cible Référence, 5 🡪 Fiche Horodatage).

\*\*Remarque : Concernant le paramètre numVariable de la méthode de comparaison de la classe « EchangeSuiviBDD », il s’agit d’un repère qui nous permet de savoir d’où vient le suivi d’échange courant, pour pouvoir générer des messages d’erreurs précis (1 🡪 Modope, 2 🡪 Procede, 3 🡪 Modcell).

\*\*\*Remarque : Concernant le paramètre typeObjet de la méthode de comparaison de la classe « EchangeVariableBDD », il s’agit d’un repère qui nous permet de connaître le type la variable d’échange courante, pour pouvoir générer des messages d’erreur précis (1 🡪   
Variable système, 3 🡪 Paramètre, 4 🡪 Paramètre systématique, 5 🡪 Traçabilité, 6 🡪 Traçabilité systématique, 7 🡪 Mesure, 8 🡪 Mesure systématique, 9 🡪 Consigne inférieure, 10 🡪 Consigne supérieure, 12 🡪 VisuDyn, 13 🡪 Horodatage, cf. le document « 9-ParametrageEchangePLC.doc »).

### Traitement des erreurs

Les erreurs seront stockées sous forme de liste d’erreur. Il s’agira de chaines de caractères retournées à l’utilisateur, indiquant où est ce que le test de comparaison a trouvé une différence entre le classeur Excel et la Base de Données, ou bien ou est-ce qu’une exception a été interceptée.

A chaque fois qu’une erreur est trouvée par le programme, il l’ajoute à une liste d’erreur. A la fin de l’exécution, si le nombre d’erreur de cette liste est inférieur à 20, ces dernières sont directement affichées à l’écran, sinon, les 20 premières erreurs seront affichées à l’écran, puis la totalité de la liste des erreurs rencontrées sera écrite dans un fichier texte.

### Traitement en background

L’appel aux méthodes de comparaisons se fera en arrière-plan, grâce à des « BackGround Workers ». Ces derniers seront lancés après un clic sur un bouton de comparaison.

Dans la méthode « bgwComparaison*XXXXX*\_DoWork », du back ground worker de la classe *XXXX*, le programme récupèrera l’identifiant de l’entité sélectionnée dans la liste déroulante en question pour le passer en paramètre de la méthode de comparaison appelée. Ainsi se lance la comparaison à partir de l’entité sélectionnée, c’est-à-dire que si l’on décide de comparer une ligne, seul les classes filles de cette ligne seront prises dans la comparaison. Si, en revanche, on compare toute la configuration, la totalité des classes seront comparées.

## Creation des elements

### Hiérarchie des éléments

Un utilisateur peut, s’il le souhaite, ajouter où mettre à jour comme bon lui semble des opérations, lignes, process, stations …

Pour ajouter un élément il faut que ses ascendants soient sélectionnés et déjà présent dans la base de données.

Si l’utilisateur veut ajouter une fiche produit, il devra sélectionner le process et la ligne correspondant à cette fiche.

Si la ligne n’est pas présente dans la base de donnée il ne pourra pas continuer il devra donc ajouter la ligne qui se chargera de tous ces descendants.

### Hiérarchie des ajouts :

La hiérarchie des ajouts se présente comme ceci :

### Liste des éléments

|  |  |  |
| --- | --- | --- |
| Methode | Information DossierConfiguration | Ecriture BDD |
| GenererConfiguration | Toutes les informations | Ecriture dans toutes les tables |
| MajModesMarche | Descriptif, Famille,  Classe, Mode | T0\_MDLMA  T1\_FAMILLE  T2\_CLASSE  T3\_MODE |
| AjouterProcess | Configuration systématique, Opération,  Ligne | T0\_PROCESS  T1\_LIGNE  T1\_OPERATION |
| AjouterOperation | Paramètres, traçabilités, Mesures, consignes | T1\_OPERATION |
| AjouterLigne | Lignes, Fiches, Paramètre | T1\_LIGNE  T2\_STATION |
| AjouterStation | Paramètres, Défauts,  Organes | T2\_STATION  T2\_DEFAUTS |
| AjouterParametreFamille | Paramètre Famille | T2\_VAR |
| AjouterParametre | Paramètre | T2\_VAR |
| AjouterTracabilite | Tracabilité | T2\_VAR |
| AjouterConsigne | Consigne | T2\_VAR |
| AjouterMesure | Mesure | T3\_ASSOCONS  T2\_VAR |
| AjouterParametreSystem | Paramètre | T2\_VAR |
| AjouterTracabiliteSystem | Traçabilité | T2\_VAR |
| AjouterConsigneSystem | Consignes | T2\_VAR |
| AjouterMesureSystem | Mesure | T3\_ASSOCONS  T2\_VAR |
| MajFicheSuivi | Modopes et procedés | T2\_FICHE  T2\_ECHANGE |
| MajFicheService | Horodatage et Sync | T2\_FICHE  T2\_ECHANGE |
| MajDefautProcess | Code, libellé, gravité | T2\_DEFAUT  T3\_PRCD |
| MajDefautProduit | Code, libellé, gravité | T2\_DEFAUT |
| MajFicheReference | Numéro fiche et cibles | T2\_FICHE  T2\_ECHANGE |
| MajFicheProduit | Paramètres, Fichier texte, numéro fiche | T2\_FICHE  T3\_TXTFIC  T4\_TXTVAR |
| MajFichierTexte | Variable fichier texte, Paramètres | T3\_TXTFIC  T4\_TXTVAR |
| MajFichePalette | Consignes, mesures, paramètres, entete, systématique , numéro fiche | T2\_FICHE  T2\_ECHANGE |

## Bouton MàJ libellés

### Objets concernés

Process -> OP -> Variables (T2\_VAR et T2\_CONSIGN)

Ligne -> Stations -> Défauts (produits et procédés, cartons)

Modes de marches (doivent être pris en compte dans la mise à jour du modèle) (A vérifier)

### Principe

Ajouter un 1 bouton sur zone Process : mettre à jour libelles associés au process et aux lignes rattachées au process dans la configuration

Etat des lieux code existant

Dans Requete.cs

|  |  |  |
| --- | --- | --- |
| Méthode | Fonction | Utilisée/Testée |
| EffacerAssociesGLOBALCUST | Efface dans GLOBAL\_CUST les messages associés à un champ donné pour une table de la configuration | NON |
| majTableBDD | Exécute une mise à jour de la table (test existence de l'enregistrement puis INSERT ou UPDATE) | NON |
| lireMaxIdRefGLOBALCUST | Recherche la valeur maximum du champ ID\_REF de la table GLOBAL\_CUST indépendamment des champs OBJET | OUI |
| EffacerAssociesGLOBAL\_CUST | Efface dans GLOBAL\_CUST les messages associés à un champ donné pour une table de la configuration | NON |

## Création base de donnees

### Principe

Ajouter l’insertion des scripts de procédure stockée, vue et fonction rattaché à la base de donnée

### Etat des lieux code existant

Dans Requete.cs

|  |  |
| --- | --- |
| **Méthode** | **Fonction** |
| CreerBaseMelodieNet | Ajout de l’appel a la méthode d’insertion de script dans la base de donnée , avec les tests |
| AjoutDesScriptsSQL | Fait appel a la méthode d’ajout d’un script dans la base de donnée , avec mise en place de test , et de retourner d’erreur SQL si l’ajout est impossible (1) ou 0 si tous est ok |
| AjoutScript | Ajout dans la base de donnée le script dont le chemin dacée est place en paramètre |
| **btnCreerClick** | Rajout du traitement d’affichage des erreur SQL |

## contrôle coherence configuration

Chaque élément du dossier de configuration, pour lequel il existe une méthode AjouterXXX ou MAJXXX possède une méthode ControlerCoherenceXXX

La méthode ControlerCoherenceConfiguration est appelée après la construction du dossier de configuration dans la méthode "BgwChargementDossierRunWorkerCompleted". Elle gère une liste d'erreurs comme le classeur excel. Cette liste est une propriété de l'objet DossierConfiguration). et est passée comme paramètre en cascade à chacune de ces méthodes. Le dossier de configuration complet est également passé en paramètre à chacune de ces méthodes, pour y permettre les recherches nécessitées par certains contrôles.

Le principe général pour les contrôles de cohérence est le même dans quasiment dans toutes les classes , d’abord on vérifie l’unicité des paramètres de certaine sous classe comme numéro nom ou ordre d’affichage ( ex Unicité num process , Unicité du nom de l’opération etc…) .

Pour les sous classes on fera des test sur l’existence de certain paramètre local et vérifier que certains d’entre eux sont bien égaux a une valeur voulu (exemple vérifier existence libellé ou vérifier inf/sup=0 ou 1 ) Vous trouverai juste en dessous le schéma de dépendance des classes Pour effectuer les contrôles cohérence.

Remarque :

Les variables "systématiques" étant gérées par les mêmes classes que leur équivalents "ordinaires", leur méthodes de contrôles de cohérence seront appelées par 2 classes:

* depuis ControlerCoherenceProcess pour les variables "systématiques" (flèches rouges du schéma)
* depuis ControlerCoherenceOperation pour les variables "ordinaires" (flèche vertes du schéma)

**ConfigurationSuiviEquipement**

**ConfigurationCibleReference**

**ConfigurationFicheHorodatage**

**ConfigurationVariableFichierTexte**

**ConfigurationFichierTexte**

**ConfigurationFicheReference**

**ConfigurationFicheProduit**

**ConfigurationFonctionService**

**ConfigurationLigne**

**ConfigurationFichePalette**

**ConfigurationFicheVisuDyn**

**ConfigurationDefaut**

**ConfigurationVisuDyn**

**ConfigurationOrgane**

**ConfigurationParametreFamille**

**ConfigurationMesure**

**ConfigurationConsigne**

**ConfigurationTracabilité**

**ConfigurationParametre**

**ConfigurationOperation**

**ConfigurationProcess**

**AccesDossierExcel**

**DossierConfiguration**

**ConfigurationEchangeVariable**

**ConfigurationModeMarche**

**ConfigurationClasseMode**

**ConfigurationFamilleMode**

**ConfigurationModeleMode**

**ConfigurationStation**

Contrôle Cohérence DossierConfiguration :

* Vérifier unicité ModelesModesMarche :

Erreur : numéro Modèle

Pour chaque process appeler ControlerCoherenceModelesModesMarche

* Vérifier unicité Process :

Erreur : numéro Process

Pour chaque process appeler ControlerCoherenceProcess

Contrôle Cohérence Process :

* Vérifier Cohérence paramètres famille
  + **V**érifier unicité paramètres famille :

Erreur : nom du parametreFamille et le numéro du process

* + Vérifier unicité Ordre Affichage pour paramètres famille :

Erreur : numéro ordreAffichage, et le numéro du process

Pour chaque paramètre famille appeler ControlerCoherenceParametreFamille

* Vérifier Cohérence paramètres Systématique
  + **V**érifier unicité paramètres Systématique:

Erreur : nom du parametreFamille et le numéro du process

* + Vérifier unicité Ordre Affichage pour paramètres Systématique:

Erreur : numéro ordreAffichage, et le numéro du process

Pour chaque paramètre Systématique appeler ControlerCoherenceParametreSystématique

* Vérifier Cohérence Traçabilité Systématique
  + **V**érifier unicité Traçabilité Systématique:

Erreur : nom de la traçabilité et le numéro du process

* + Vérifier unicité Ordre Affichage pour Traçabilité Systématique:

Erreur : numéro ordreAffichage, et le numéro du process

Pour chaque paramètre Systématique appeler ControlerCoherenceTraçabiliteSystématique

* Vérifier Cohérence Consigne Systématique
  + **V**érifier unicité Consigne Systématique:

Erreur : nom de la consigne et le numéro du process

Pour chaque Consigne Systématique appeler ControlerCoherenceConsigneSystématique

Vérifier Cohérence Mesure Systématique

* + **V**érifier unicité Mesure Systématique:

Erreur : nom de la mesure et le numéro du process

* + Vérifier unicité Ordre Affichage pour Mesure Systématique:

Erreur : numéro ordreAffichage, et le numéro du process

Pour chaque Consigne Systématique appeler ControlerCoherenceMesureSystématique

* Vérifier Cohérence Opération
  + Vérifier unicité rang bit « Gamme »

Erreur : numéro opération, numéro process

* + **V**érifier unicité Opération

Erreur : numéro Opération et le numéro du process

Pour chaque opération appeler ControlerCoherenceOperation

* Vérifier Cohérence Ligne
  + **V**érifier unicité Ligne

Erreur : numéro ligne et le numéro du process

Pour chaque ligne appeler ControlerCoherenceLigne

* Vérifier Cohérence entre mesures et consignes
  + Vérifier que la Mesure consigne mini s’agit bien d’un consigne inferieur (=0) :

Erreur : nom de la consigne et le numéro du process

* + Vérifier que la Mesure consigne maxi s’agit bien d’un consigne supérieur (=1):

Erreur : nom de la consigne et le numéro du process

Contrôle Cohérence ConfigurationOperation :

* Vérifier Cohérence paramètres
  + **V**érifier unicité paramètres :

Erreur : nom du paramètre, le numéro de l’opération et le numéro du process

* + Vérifier unicité Ordre Affichage pour paramètres:

Erreur : numéro ordreAffichage, nom du paramètre, numéro de l’opération et le numéro du process

Pour chaque paramètre appeler ControlerCoherenceParametre

* Vérifier Cohérence Traçabilité
  + **V**érifier unicité Traçabilité:

Erreur : nom de la traçabilité, le numéro de l’opération et le numéro du process

* + Vérifier unicité Ordre Affichage pour Traçabilité Systématique:

Erreur : numéro ordreAffichage, nom de la traçabilité, numéro de l’opération et le numéro du process

Pour chaque Traçabilité appeler ControlerCoherenceTraçabilite

* Vérifier Cohérence Consigne
  + **V**érifier unicité Consigne:

Erreur : nom de la Consigne, le numéro de l’opération et le numéro du process

Pour chaque Consigne appeler ControlerCoherenceConsigne

* Vérifier Cohérence Mesure
  + **V**érifier unicité Mesure:

Erreur : nom de la Mesure, le numéro de l’opération et le numéro du process

* + Vérifier unicité Ordre Affichage pour Mesure:

Erreur : numéro ordreAffichage, nom de la Mesure, numéro de l’opération et le numéro du process

Pour chaque Mesure appeler

ControlerCoherenceMesure

* Vérifier Cohérence entre mesures et consignes
  + Vérifier que la Mesure consigne mini s’agit bien d’un consigne inferieur (=0) :

Erreur : nom de la consigne et le numéro du process

* + Vérifier que la Mesure consigne maxi s’agit bien d’un consigne supérieur (=1):

Erreur : nom de la consigne et le numéro du process

* + Vérifier que le rang bit « Gamme » soit supérieur à 0:

Erreur : numéro opération, numéro process

Contrôle Cohérence ConfigurationTracabilité,  ConfigurationParametre, ConfigurationParametreFamille , ConfigurationMesure :

* Vérifier existence libellé1 :

Erreur : nom du paramètre (mesure parametreFamille ou traçabilité), (le numéro de l’opération si tester à partir d’une opération) et le numéro du process (probleme)

* Vérifier existence Ordre Affichage ( !=0) :

Erreur : nom du paramètre (mesure consigne ou traçabilité), (le numéro de l’opération si tester à partir d’une opération) et le numéro du process

Contrôle Cohérence ConfigurationConsigne :

* Vérifier existence libellé1 :

Erreur : nom de la consigne, (le numéro de l’opération si tester à partir d’une opération) et le numéro du process

* Vérifier inf/sup=0 ou 1 :

Erreur : nom de la consigne, (le numéro de l’opération si tester à partir d’une opération) et le numéro du process

Contrôle Cohérence ConfigurationLigne :

* Vérifier Unicité de la station :

Erreur : numéro de la station, numéro de la ligne et le numéro du process

Pour chaque Station appeler ControlerCoherenceStation

* Vérifier Unicité de la FicheReference :

Erreur : numéro de la FicheReference, numéro de la ligne et le numéro du process

Pour chaque fiche Reference appeler ControlerCoherenceFicheReference

* Vérifier Unicité de la FichePalette :

Erreur : numéro de la FichePalette, numéro de la ligne et le numéro du process

Pour chaque fiche Palette appeler ControlerCoherenceFichePalette

* Vérifier Unicité de la ficheProduit :

Erreur : numéro de la FicheProduit, numéro de la ligne et le numéro du process

Pour chaque fiche Produit appeler ControlerCoherenceFichePruduit

* Vérifier existence numComman:

Erreur : numéro de la ligne et le numéro du process

* Vérifier existence de l’unité d’archivage:

Erreur : numéro de la ligne et le numéro du process

* Vérifier existence de la durée de maintien:

Erreur : numéro de la ligne et le numéro du process

* Vérifier existence de l’activation du journal :

Erreur : numéro de la ligne et le numéro du process

* Vérifier existence de l’activation de sanction :

Erreur : numéro de la ligne et le numéro du process

* SI existence de la fonctionService :

appeler ControlerCoherenceFonctionService

* Si existence du SuiviEquipement:

appeler ControlerCoherenceSuiviEquipement

* Si existence de la ficheVisuDyn :

appeler ControlerCoherenceFicheVisuDyn

ControleCohérence ConfigurationFonctionService :

* Vérifier unicité de la syncs : ( Excel devrai le faire auto car pas de num sync)

Erreur : numéro de la fonction service, numéro de la ligne et le numéro du process

Pour chaque syncs appeler ControleCoherenceEchangeVariable

* Vérifier unicité de la fiche Horodatage :

Erreur : numéro de la fonction service, numéro de la ligne et le numéro du process

Pour chaque fiche horodatages appeler ControleCoherenceFicheHorodatage

Contrôle cohérence ConfigurationFicheHorodatage

Pour chaque fiche horodatage appeler ControleCoherenceEchangeVariable

Contrôle cohérence ConfigurationSuiviEquipement :

* Vérifier unicité fiche Modope:

Erreur : numéro de la fiche Modope, numéro de l’equipementSuivi, numéro de la ligne et le numéro du process

Pour chaque modopes appeler ControleCoherenceEchangeSuivi

* Vérifier unicité fiche procède:

Erreur : numéro de la fiche procède, numéro de l’equipementSuivi, numéro de la ligne et le numéro du process

Pour chaque procèdes appeler ControleCoherenceEchangeSuivi

Contrôle cohérence ConfigurationFicheVisuDyn :

* Vérifier unicité VisuDyn:

Erreur : numéro de la fiche VisuDyn, numéro de la ligne et le numéro du process

Pour chaque visuDyns apeller ControleCoherenceEchangeVariable

Contrôle cohérence ConfigurationFichePalette :

* Vérifier unicité entête:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque entête appeler ControleCoherenceEchangeVariable

* Vérifier unicité paramètre Systématique:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque paramètre Systématique appeler ControleCoherenceEchangeVariable

* Vérifier unicité paramètre:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque paramètre appeler ControleCoherenceEchangeVariable

* Vérifier unicité traçabilité Systématique:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque traçabilité Systématique appeler ControleCoherenceEchangeVariable

* Vérifier unicité traçabilité:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque traçabilité appeler ControleCoherenceEchangeVariable

* Vérifier unicité consigne Inferieures :

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque consigne Inferieures appeler ControleCoherenceEchangeVariable

* Vérifier unicité consigne Supérieure:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque consigne Supérieure appeler ControleCoherenceEchangeVariable

* Vérifier unicité mesure Systematique:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque mesure Systématique appeler ControleCoherenceEchangeVariable

* Vérifier unicité mesure:

Erreur : numéro de la fiche Palette, numéro de la ligne et le numéro du process

Pour chaque mesure appeler ControleCoherenceEchangeVariable

Contrôle cohérence ConfigurationFicheProduit :

* Vérifier unicité entête:

Erreur : numéro de la fiche Produit, numéro de la ligne et le numéro du process

Pour chaque entête appeler ControleCoherenceEchangeVariable

* Vérifier unicité traçabilité Systématique:

Erreur : numéro de la fiche Produit, numéro de la ligne et le numéro du process

Pour chaque traçabilité Systématique appeler ControleCoherenceEchangeVariable

* Vérifier unicité traçabilité:

Erreur : numéro de la fiche Produit, numéro de la ligne et le numéro du process

Pour chaque traçabilité appeler ControleCoherenceEchangeVariable

* Vérifier unicité consigne Inferieures :

Erreur : numéro de la fiche Pal Produit ette, numéro de la ligne et le numéro du process

Pour chaque consigne Inferieures appeler ControleCoherenceEchangeVariable

* Vérifier unicité consigne Supérieure:

Erreur : numéro de la fiche Produit, numéro de la ligne et le numéro du process

Pour chaque consigne Supérieure appeler ControleCoherenceEchangeVariable

* Vérifier unicité mesure Systematique:

Erreur : numéro de la fiche Produit, numéro de la ligne et le numéro du process

Pour chaque mesure Systématique appeler ControleCoherenceEchangeVariable

* Vérifier unicité mesure:

Erreur : numéro de la fiche Produit, numéro de la ligne et le numéro du process

Pour chaque mesure appeler ControleCoherenceEchangeVariable

Contrôle cohérence ConfigurationFicheReference :

* Vérifier unicité cible Reference:

Erreur : numéro de la fiche Reference, numéro de la ligne et le numéro du process

Pour chaque cible référence appeler ControleCoherenceCibleReference

Contrôle cohérence ConfigurationCibleReference :

* Vérifier unicité entête:

Erreur : numéro de la fiche Reference, numéro de la ligne et le numéro du process

Pour chaque entête appeler ControleCoherenceEchangeVariable

* Vérifier unicité paramètre Systématique:

Erreur : numéro de la fiche Reference, numéro de la ligne et le numéro du process

Pour chaque paramètre Systématique appeler ControleCoherenceEchangeVariable

* Vérifier unicité paramètre:

Erreur : numéro de la fiche Reference, numéro de la ligne et le numéro du process

Pour chaque paramètre appeler ControleCoherenceEchangeVariable

Contrôle cohérence ConfigurationStation :

* Vérifier Unicité Cartons Jaunes :

Erreur : numéro du carton Jaunes, numéro de la station, numéro de la ligne et le numéro du process

Pour chaque Carton Jaune appeler ControlerCoherenceDefaut

* Vérifier Unicité Carton Oranges :

Erreur : numéro du Carton Orange, numéro de la station, numéro de la ligne et le numéro du process

Pour chaque Carton Orange appeler ControlerCoherenceDefaut

* Vérifier Unicité DefautProduit :

Erreur : numéro du defautProduit, numéro de la station, numéro de la ligne et le numéro du process

Pour chaque Défaut Produit appeler ControlerCoherenceDefaut

* Vérifier Unicité DefautProcédé :

Erreur : numéro du DefautProcédé, numéro de la station, numéro de la ligne et le numéro du process

Pour chaque Défaut Procédé appeler ControlerCoherenceDefaut

* Vérifier existence RangBitGamme :

Erreur : numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence ModelModeMarche :

Erreur : numéro de la station, numéro de la ligne et le numéro du process

Si Organe à surveiller existe

* + - * Vérifier Unicité Ordre Affichage

Erreur : numéro Ordre Affichage, numéro de la station, numéro de la ligne et le numéro du process

Pour chaque Organe à surveiller appeler ControlerCoherenceOrganes

Si VisuDyn existe

Pour chaque Organe à surveiller appeler ControlerCoherenceVisuDyn

Contrôle cohérence ConfigurationDefaut :

* Vérifier existence gravité

Erreur : numéro du Défaut, numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence code

Erreur : libelle du Défaut, numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence libelle

Erreur : numéro du Défaut, numéro de la station, numéro de la ligne et le numéro du process

Contrôle cohérence ConfigurationOrgane :

* Vérifier existence DescriptifLangue1

Erreur : numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence IncrémentStandard

Erreur : numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence alerteBasse

Erreur : numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence alerteHaute

Erreur : numéro de la station, numéro de la ligne et le numéro du process

* Vérifier alerteBasse<alerteHaute

Erreur : numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence ordreaffichage

Erreur : numéro de la station, numéro de la ligne et le numéro du process

Contrôle cohérence VisuDyn :

* Vérifier existence nom VisuDyn

Erreur : numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence type

Erreur : nom de la visuDyn, numéro de la station, numéro de la ligne et le numéro du process

* Vérifier type=1ou2

Erreur : nom de la visuDyn, numéro de la station, numéro de la ligne et le numéro du process

* Vérifier existence descriptif langue 1

Erreur : nom de la visuDyn, numéro de la station, numéro de la ligne et le numéro du process

Contrôle cohérence EchangeVariable

* Vérifier existence varibalePLC

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* Vérifier existence protocole

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si protocole =OPC
      * Vérifier Existence Item

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence numero GroupeOPC

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence GroupeOPC dans onglet général

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si protocole =Texte
      * Vérifier existence NomFichier

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence NumeroLigne

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - Sinon protocole =applicom
      * Vérifier existence Canal

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence EquipementDefini

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence Adresse

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + Vérifier type de l’objet

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 1 ;2 ; 11 ;13
      * Vérifier dans T0\_VARPREDEF existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 9 ;10
      * Vérifier dans liste consigne process et opération existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 12
      * Vérifier dans la liste des VisuDyn existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 3 ;4 ;5 ;6 ;7 ;8
      * Vérifier dans la liste des process et opération existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

Contrôle cohérence EchangeSuivi

* Vérifier existence varibalePLC

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* Vérifier existence numStation

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* Vérifier existence varibalePLC

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* Vérifier existence protocole

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si protocole =OPC
      * Vérifier Existence Item

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence numéro GroupeOPC

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence GroupeOPC dans onglet général

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si protocole =Texte
      * Vérifier existence NomFichier

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence NumeroLigne

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - Sinon protocole =applicom
      * Vérifier existence Canal

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence EquipementDefini

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + - * Vérifier existence Adresse

Erreur : nom protocole numéro de la fiche, numéro de la ligne et le numéro du process

* + Vérifier type de l’objet

Erreur : numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 1 ;2 ; 11 ;13
      * Vérifier dans T0\_VARPREDEF existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 9 ;10
      * Vérifier dans liste consigne process et opération existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 12
      * Vérifier dans la liste des VisuDyn existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

* + - Si le type est 3 ;4 ;5 ;6 ;7 ;8
      * Vérifier dans la liste des process et opération existence

Erreur : type, numéro de la fiche, numéro de la ligne et le numéro du process

Contrôle Cohérence ModeDeMarche :

* Vérifier existence libellé 1

Erreur : numéro du modelé

* Vérifier unicité positionAffichageFamille

Erreur : numéro du modelé

Pour chaque mode de marche appeler ControlerCoherenceFamilleModes

Contrôle Cohérence FamilleModes :

* Vérifier existence description langue 1

Erreur : numéro du modèle, numéro famille

* Vérifier existence couleur

Erreur : numéro du modèle, numéro famille

* Vérifier existence libellé 1

Erreur : numéro du modèle, numéro famille

Pour chaque famille modes appeler ControlerCoherenceClasseMode

Contrôle cohérence ClasseMode :

* Vérifier existence libellé 1

Erreur : numéro du modèle, numéro famille, numéro classe

Pour chaque classe mode appeler ControlerCoherenceModeMarche

Contrôle cohérence ModeMarche :

* Vérifier existence libellé 1

Erreur : numéro du modèle, numéro famille, numéro classe, numéro mode

* Vérifier existence attribuRendement

Erreur : numéro du modèle, numéro famille, numéro classe, numéro mode

## Traitement des modeles de modes de marche

Attention pour les modèles de modes de marche, il ne faut pas effacer le modèle car des enregistrements peuvent exister dans la table de production T1\_MODMARCH (les codes de mode de marche)

**En fait, la structure hiérarchique d'un modèle de modes de marche rend difficile une fonctionnalité de mise à jour sans perte. Or, le risque vis à vis des données de production reste limité car ces dernières ne sont destinées qu'à faciliter la maintenance. Le programme fonctionnera donc par effacement du modèle existant et recréation.**

*Si le modèle n'existe pas on le crée*

*Si le modèle existe*

*Les Familles ne peuvent être accéder par un numéro (auto incrémenté et absent du dossier Excel)*

*Pour un modèle donné (ID\_MDLMA) la seule identification peut être faite par le libelle.*

*Recherche dans GLOBAL\_CUST des libellés et des id\_ref pour OBJET = T1\_FAMILLE et ID\_LANG = numéro langue 1 (ExecuterSELECT)*

*Pour chaque libellé ainsi trouvé, comparaison avec le libellé1*

*Si égalité, la famille existe*

* *Récupérer l'ID\_REF (dans le dataReader)*
* *Mettre à jour GLOBAL\_CUST pour le libellé2 (majTableBDD pour cet id\_ref et OBJET = T1\_FAMILLE)*
* *Mettre à jour T1\_FAMILLE pour T1\_FAMILLE\_LIB = id\_ref*
* *Mettre à jour les classes*

*Si non la famille n'existe pas*

* *Créer l'enregistrement dans T1\_FAMILLE*
* *Mettre à jour les classes (en fait cela impliquera une création)*

*Les Classes ne peuvent être accéder par un numéro (auto incrémenté et absent du dossier Excel)*

*Pour une famille donnée (ID\_FAMILLE) la seule identification peut être faite par le libelle.*

*Recherche dans GLOBAL\_CUST des libellés et des id\_ref pour OBJET = T2\_CLASSE et ID\_LANG = numéro langue 1 (ExecuterSELECT)*

*Pour chaque libellé ainsi trouvé, comparaison avec le libellé1*

*Si égalité, la classe existe*

* *Récupérer l'ID\_REF (dans le dataReader)*
* *Mettre à jour GLOBAL\_CUST pour le libellé2 (majTableBDD pour cet id\_ref et OBJET = T2\_CLASSE)*
* *Rechercher ID\_CLASSE pour T2\_CLASSE\_LIB = ID\_REF*
* *Mettre à jour T2\_CLASSE pour T2\_CLASSE\_LIB = id\_ref*
* *Mettre à jour les modes*

*Si non la classe n’existe pas*

* *Créer l'enregistrement dans T2\_FAMILLE*
* *Mettre à jour les modes (en fait cela impliquera une création)*

*Modes*

*Les modes sont accessibles par leur code, mais le lien avec la classe peut-être perdu*

*Faire mise à jour sur le code mode, Y compris GLOBAL\_CUST*

## Chargement Base de Données

### Structure Dossier configuration

Le contenu de la Base de Données est stocké dans la classe DossierConfigurationBDD. ….

# Annexes

## Diagramme de classe