

```

/*****
 *   Logiciel portageMM2CPP de portage de code ModelMaker en langage C++   *
 *   Copyright INRA, février 2006                                         *
 *****/

```

```

/*****
 * Licence :

```

Le logiciel portageMM2CPP est une réécriture en langage C++ du modèle écrit dans le fichier ModelMaker "modele2.mod" (situé dans le répertoire "leSourceModelMaker"). Voir informations dans ../laDocumentation/help.

Ce logiciel est régi par la licence CeCILL soumise au droit français et respectant les principes de diffusion des logiciels libres. Vous pouvez utiliser, modifier et/ou redistribuer ce programme sous les conditions de la licence CeCILL telle que diffusée par le CEA, le CNRS et l'INRIA sur le site "<http://www.cecill.info>".

En contrepartie de l'accessibilité au code source et des droits de copie, de modification et de redistribution accordés par cette licence, il n'est offert aux utilisateurs qu'une garantie limitée. Pour les mêmes raisons, seule une responsabilité restreinte pèse sur l'auteur du programme, le titulaire des droits patrimoniaux et les concédants successifs.

A cet égard l'attention de l'utilisateur est attirée sur les risques associés au chargement, à l'utilisation, à la modification et/ou au développement et à la reproduction du logiciel par l'utilisateur étant donné sa spécificité de logiciel libre, qui peut le rendre complexe à manipuler et qui le réserve donc à des développeurs et des professionnels avertis possédant des connaissances informatiques approfondies. Les utilisateurs sont donc invités à charger et tester l'adéquation du logiciel à leurs besoins dans des conditions permettant d'assurer la sécurité de leurs systèmes et ou de leurs données et, plus généralement, à l'utiliser et l'exploiter dans les mêmes conditions de sécurité.

Le fait que vous puissiez accéder à cet en-tête signifie que vous avez pris connaissance de la licence CeCILL, et que vous en avez accepté les termes.

Les fichiers Licence\_CeCILL\_V2-fr.txt et Licence\_CeCILL\_V2-en.txt du répertoire ../LICENCE fournissent le texte de la licence, dans sa version 2, en langue française et en langue anglaise. (ces 2 fichiers sont aussi directement dans le répertoire courant)

```

*****/
/*****

```

```
*
* Fichier      : genericiteConfigPortageMM2CPP.h
*
* Auteur(s)    : Nathalie Rousse, Nathalie.Rousse@toulouse.inra.fr
*                de l'INRA - Institut National de la Recherche Agronomique -
*                (département MIA, UMR AGIR, http://www.modelia.org).
*
* Description  :
*
* Ce fichier fait partie du code source C++ du logiciel portageMM2CPP.
*
* Il concerne la configuration de la partie générique qui "se veut"
* indépendante du modèle traité (déclarations des classes), il est associé
* aux fichiers genericitePortageMM2CPP.h et genericitePortageMM2CPP.cpp.
* Il contient des constantes de configuration qu'il est intéressant d'isoler
* ici parce que susceptibles d'être touchées pour les besoins de scénarios
* de test du logiciel.
*
*****
* Historique  :
*
* 26/11/07, Nathalie Rousse : création du fichier.
*
* Le fichier est créé par déplacement de texte qui se trouvait jusque là
* dans le fichier genericitePortageMM2CPP.h.
*
* (création dans le cadre de la modification m_repertoires : réorganisation
* des répertoires pour une organisation mieux adaptée, plus souple).
*
*****/

/* modeDemploi
* Instruction s'adressant à quelqu'un qui est en train de construire son propre logiciel C++ relativement à son propre
* modèle ModelMaker : dans tout le code de ce fichier, consulter et suivre les consignes/indications marquées du label "
* modeDemploi". Voir aussi "Mode d'emploi" dans le fichier "help".
*/

/* modeDemploi
*
* Ce fichier n'est pas sensé être retouché A PRIORI, pas plus que genericitePortageMM2CPP.h et genericitePortageMM2CPP
*.cpp.
*
* Cependant, leur code est susceptible de bouger au titre d'évolutions, qui peuvent être nécessaires pour différentes
* raisons (voir les commentaires marqués du label "modeDemploiEvolutions" dans tout le code de ce fichier (et de generici
* tePortageMM2CPP.h et genericitePortageMM2CPP.cpp).
*
```

```
* Il est possible que des scénarios de test impliquent de produire localement (dans l'espace même du scénario) un fichier genericiteConfigPortageMM2CPP.h remplaçant celui du logiciel testé.
```

```
*/
```

```
#ifndef _GENERICITE_CONFIG_H_INCLUS_
```

```
#define _GENERICITE_CONFIG_H_INCLUS_
```

```
/*  
*****
```

```
*/
```

```
*/ Constantes relatives à la classe fichierDentree
```

```
*/
```

```
*****/
```

```
/*
```

```
*/ Dimension maximale d'un nom de fichier
```

```
*/ Pour : nomFichier de la classe fichierDentree
```

```
*/
```

```
#define TAILLE_MAX_NOM_FICHIER 200
```

```
/*  
*****
```

```
*/
```

```
*/ Constantes relatives à la classe horlogeLocale
```

```
*/
```

```
*****/
```

```
#define INSTANT_INITIAL_DEFAULT 0.0 // instant initial par défaut : 0 seconde
```

```
/* modeDemploi_ConfDim
```

```
*/ Eventuellement, changer la valeur de INSTANT_INITIAL_DEFAULT si c'est souhaité
```

```
*/
```

```
*/ Exemple supplémentaire (nommé "bh") d'un modèle journalier :
```

```
*/ pas besoin de modifier la valeur de INSTANT_INITIAL_DEFAULT:
```

```
*/ #define INSTANT_INITIAL_DEFAULT 0.0 // instant initial par défaut : 0 seconde
```

```
*/
```

```
#define DELTA_T_DEFAULT 0.001 // pas de temps par défaut : 0.001 seconde
```

```
/* modeDemploi_ConfDim
```

```
*/ Eventuellement, changer la valeur de DELTA_T_DEFAULT si c'est souhaité
```

```
*/
```

```
*/ Exemple supplémentaire (nommé "bh") d'un modèle journalier :
```

```
*/ La valeur de DELTA_T_DEFAULT est modifiée (0.001 ie 0.001 seconde ne convenait pas) :
```

```
*/ #define DELTA_T_DEFAULT 1.0 // pas de temps par défaut : 1.0 seconde
```

```
*/
```

```
/*  
*****
```

```
*/
```

```
* Constantes relatives à la classe listeEntitesAvecDerivee
*
*****/

/* nombre maximum d'objets déclarés de type entityAvecDerivee */
#define NB_MAX_EntitesAvecDerivee 10000

/* modeDemploi_ConfDim
* Verifier que NB_MAX_EntitesAvecDerivee majore bien le nombre d'objets de type entityAvecDerivee (ou ses classes filles) qui sont déclarés dans la classe systemParticulier. Si ce n'est pas le cas, augmenter sa valeur pour que ça devienne vrai.
* Suggestion : comme les sous-classes associées aux entités ModelMaker (constant, parameter, variable, flow, compartment, controlled) sont filles soit de la classe entityInstantanee, soit de la classe entityAvecDerivee, soit de la classe entityInterpolee, il suffit de choisir NB_MAX_EntitesAvecDerivee supérieur au nombre d'éléments ModelMaker qui sont déclarés dans la classe systemParticulier.
*/

/*****
*
* Constantes relatives à la classe listeEntitesInterpolees
*
*****/

/* nombre maximum d'objets declares de type entityInterpolee */
#define NB_MAX_EntitesInterpolees 10000

/* modeDemploi_ConfDim
* Verifier que NB_MAX_EntitesInterpolees majore bien le nombre d'objets de type entityInterpolee (ou ses classes filles) qui sont déclarés dans la classe systemParticulier. Si ce n'est pas le cas, augmenter sa valeur pour que ça devienne vrai.
* Suggestion : comme les sous-classes associées aux entités ModelMaker (constant, parameter, variable, flow, compartment, controlled) sont filles soit de la classe entityInstantanee, soit de la classe entityAvecDerivee, soit de la classe entityInterpolee, il suffit de choisir NB_MAX_EntitesInterpolees supérieur au nombre d'éléments ModelMaker qui sont déclarés dans la classe systemParticulier.
*/

/*****
*
* Constantes relatives à la classe genericSystem
*
*****/

/*
* Valeur minimale du rapport pasDeLaSimulation/deltat.
* Cette constante servira à "ajuster" (en fait augmenter) pasDeLaSimulation
```

```
* de telle sorte qu'il y ait au moins PAS_SIM_SUR_PAS_TEMPS_MIN itérations
* deltat dans la durée pasDeLaSimulation
*/
#define PAS_SIM_SUR_PAS_TEMPS_MIN 1000 // est-ce suffisant ? (il n'y a aucun "filet")

/* modeDemploi_ConfDim
* S'assurer que PAS_SIM_SUR_PAS_TEMPS_MIN n'est pas trop faible par rapport à son système particulier (en fait, par rapport aux équations de "actualiserIncrement"). Si cela est besoin, augmenter sa valeur.
* Indications : Plus PAS_SIM_SUR_PAS_TEMPS_MIN est grande, plus les calculs seront précis, mais aussi plus l'exécution des simulations sera lente. Il s'agit de trouver le compromis entre la précision des calculs et la vitesse de calcul.
* ATTENTION : Il vaut mieux sur estimer que sous estimer cette valeur. Si PAS_SIM_SUR_PAS_TEMPS_MIN est trop faible, il y a risque d'ERREURS de calculs (notamment dans le calcul des dérivées). C'est un POINT TRES DELICAT car il n'est fait aucun controle dans le code. Il faut déceler un éventuel problème lors des tests (lancement et vérifications de simulations).
*/

#endif /* _GENERICITE_CONFIG_H_INCLUS_ */

/*****
*
* Fin du fichier : genericiteConfigPortageMM2CPP.h
*
*****/
```