RAPPORT

IRSN
INSTITUT
DE RADIOPROTECTION
ET DE SÛRETÉ NUCLÉAIRE

*Faire avancer la sûreté nucléaire*

# Réalisation d'une interface graphique pour Evariste

## *Développement du module de calcul en C++*

Gaëtan Bayle des Courchamps*

Rapport FT/DSR/SAGR/2016/ N°xxx
Rapport EPS2/REP900/NS2-xxx

*Société Alliance Services Plus

IRSN
INSTITUT
DE RADIOPROTECTION
ET DE SÛRETÉ NUCLÉAIRE

Pôle sûreté des installations et des systèmes nucléaires
Service des Accidents Graves

INSTITUT
DE RADIOPROTECTION
ET DE SÛRETÉ NUCLÉAIRE

**Pôle sûreté des installations et des systèmes nucléaires**
Service des Accidents Graves
B.P. 17,
92262 Fontenay-aux-Roses Cedex

## Identification

Type de document : Rapport technique (conception et implémentation)

## Auteurs

| Personne | Fonction | Organisme | Visa | Date du visa |
|---|---|---|---|---|
| Gaëtan BAYLE | Ingénieur études et réalisations | AS+ Groupe EOLEN | | |

## Vérification

| Personne | Fonction | Organisme | Visa | Date du visa |
|---|---|---|---|---|
| Béatrice BRELIVET | Directeur du pôle HPC | AS+ Groupe EOLEN | | |

## Approbation

| Personne | Fonction | Organisme | Visa | Date du visa |
|---|---|---|---|---|
| Béatrice BRELIVET | Directeur du pôle HPC | AS+ Groupe EOLEN | | |

## Diffusion

| Personne | Fonction | Organisme | Action | Visa |
|---|---|---|---|---|
| V. TANCHOUX | Responsable technique Consultation | IRSN | Décision | |
| J. DENIS | Responsable technique Consultation | IRSN | Décision | |

## Liste des versions et révisions

| Version / Révision | Date | Objet |
|---|---|---|
| 0.1 | 19/12/2016 | Création d'après notes. |
| 0.2 | 20/12/2016 | Refonte pour la nouvelle implémentation. |
| 0.3 | 21/12/2016 | Ligne de commande de l'exécutable. |

Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*
**Siège social : 31 av. de la division Leclerc – 92260 Fontenay-aux-Roses - Standard +33 (0)1 58 35 88 88 - RCS Nanterre B 440 546 018**

# Fiche descriptive du rapport
## *Report description sheet*

**Titre**

**Réalisation d'une interface graphique pour Evariste**

**Sous-titre**

**Développement du module de calcul en C++**

**Title**

**Sub title**

**Auteur/author(s) :** Gaëtan Bayle des Courchamps

| | | | |
|---|---|---|---|
| **Type de document :** *Document type :* | **RT** | **Date de diffusion :** *Distribution date :* | |
| **Référence archivage :** | | **E-mail :** | |
| **Elément EOTP** | | | |

| | |
|---|---|
| **Mots-clés (Max. 5) :** | EPS2, KANT, Windows 7, x64 |
| **Key-words (Max. 5):** | |

**RÉSUMÉ**

Ce document est un élément de la livraison finale du projet «Réalisation d'une interface graphique pour Evariste».

Il concerne principalement les développeurs de cette interface graphique qui veulent utiliser le code de calcul sous-jacent.

Il décrit les choix d'implémentation, l'utilisation du code pris comme référence, et la structure du code adoptée pour le moteur de calcul.

Ce document a été mis à jour le 2017/09/21 pour correspondre à la révision 279 du code sur le serveur SVN du projet EVARISTE.

**ABSTRACT**

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

# SOMMAIRE

1 INTRODUCTION ..... 9

**1.1 CONVENTIONS ..... 9**

**1.2 CODE SOURCE ..... 9**

**1.3 SIGLES ET ACRONYMES ..... 10**

**1.4 RÉFÉRENCES ..... 10**

2 MOTEUR DE CALCUL ..... 10

**2.1 COMPILATION ET INSTALLATION ..... 10**

**2.2 LIGNE DE COMMANDE ..... 11**

**2.3 CHOIX ET TECHNIQUES D'IMPLÉMENTATION ..... 11**

2.3.1 Structure du code ..... 11

2.3.2 Idée générale ..... 12

2.3.3 Remplacement des MFC par la bibliothèque standard C++ ..... 12

**2.4 PORTAGES VERS LINUX ..... 13**

2.4.1 Création d'un Makefile ..... 13

2.4.1.1 Contenu du Makefile ..... 13

2.4.1.2 Compilation du moteur avec le Makefile ..... 14

2.4.2 Principales modifications ..... 14

2.4.3 Particularités du portage de MER ..... 15

2.4.3.1 Sélection du code source ..... 15

2.4.3.2 Dépendances CVODE et Xerces ..... 16

2.4.3.3 Compilation du code ..... 16

2.4.3.4 Tests et comportement ..... 16

3 EVARISTE : ÉTAT INITIAL DU CODE ..... 17

**3.1 UTILISATION DE L'IHM ..... 17**

3.1.1 Remarques préliminaires ..... 17

3.1.2 Calcul des rejets ..... 17

3.1.3 Dispersion ..... 17

3.1.3.1 Onglet : «Données météorologiques» ..... 17

3.1.3.2 Onglet : «Paramètres d'exécution de pX» ..... 17

3.1.3.3 Onglet «Résultats demandés» ..... 18

3.1.3.4 Démarrage effectif du calcul ..... 18

3.1.4 Conséquences radiologiques ..... 19

3.1.4.1 Fenêtre «Gestion des conséquences radiologiques» ..... 19

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

3.1.4.2 Fichiers utilisés et produits ..... 19

**3.2 FICHIERS DE L'IHM ..... 19**

3.2.1 Base de registres ..... 19

3.2.2 Fichiers .eps2 ..... 21

**3.3 INSTALLATIONS ET RÉPERTOIRES DE BASE_MERCOR ET EVARISTE ..... 21**

3.3.1 BASE_MERCOR ..... 21

3.3.2 Les fichiers statiques pour PX ..... 22

3.3.2.1 Le fichier de filiation ..... 22

3.3.2.2 Le fichier de radio-éléments ..... 23

3.3.3 Les fichiers traités par pX ..... 23

3.3.4 EVARISTE ..... 23

3.3.5 Base de registres ..... 24

**3.4 STRUCTURE DU CODE SOURCE INITIAL ..... 26**

3.4.1 BASE_MERCOR ..... 26

3.4.2 EVARISTE ..... 26

3.4.2.1 Le code source ..... 26

3.4.2.2 Points de repère pour le débogage ..... 26

3.4.3 MER ..... 26

4 UTILISATION DES EXÉCUTABLES MER, PX, CONSX ..... 27

**4.1 PROLOGUE : MER, PX, CONSX POUR LES NULS ..... 27**

**4.2 MER.EXE ..... 28**

4.2.1 Ligne de commande de MER ..... 28

4.2.2 Format de fichier d'entrée de MER (JDD) ..... 28

4.2.3 Format des fichiers de sortie de MER ..... 29

4.2.3.1 Valeur de groupfilter TermeSource ..... 29

4.2.3.2 valeurs de groupfilter autres que TermeSource ..... 29

4.2.3.3 Valeurs invalides de Groupfilter ..... 30

4.2.3.4 Irrégularités dans les formats de sortie de MER ..... 30

**4.3 PX.EXE ..... 30**

4.3.1 Ligne de commande pour pX ..... 30

4.3.2 Les fichiers statiques traités par pX ..... 30

4.3.2.1 Le fichier de filiation ..... 30

4.3.2.2 Le fichier de radio-éléments ..... 30

4.3.3 Les fichiers traités par pX ..... 30

4.3.3.1 Utilisation de NetCDF par pX ..... 30

4.3.3.2 Fichier météo (entrée) ..... 32

4.3.3.3 Fichier des résultats demandés (entrée) ..... 32

4.3.3.4 Fichier des sources (entrée) ..... 32

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

4.3.3.5 Les résultats (sortie) ..... 33
**4.4 CONSX ..... 33**
4.4.1 Ligne de commande pour ConsX ..... 33
4.4.2 Les fichiers statiques ..... 34
4.4.2.1 Solubilités ..... 34
4.4.2.2 Coefficients de dose ..... 34
4.4.3 Fichiers d'entrée ..... 34
4.4.3.1 Le fichier de paramètres ..... 34
4.4.3.2 Le fichier d'entrée ..... 34
4.4.3.3 Les fichiers statiques ..... 35
4.4.4 Fichier de sortie ..... 35
4.4.4.1 Format de fichier pour ConsX ..... 35

5 DÉTAIL DES RÉVISIONS SVN ..... 36
**5.1 PORTAGE DE MER POUR LINUX ..... 36**
**5.2 CODAGE DE L'IHM ET DU MOTEUR DE CALCUL ..... 36**

6 INSTALLATION/UTILISATION DE MPI ..... 36
**6.1 INTRODUCTION À MPI ..... 36**
6.1.1 Contenu d'une implémentation MPI ..... 36
6.1.2 Contraintes matérielles ..... 37
6.1.3 Contraintes de configuration ..... 37
**6.2 MISE EN RÉSEAU DE PCS SOUS WINDOWS 7 ..... 37**
6.2.1 Configurer les adresses IP ..... 37
6.2.2 Configurer les connexions réseau pour avoir un groupe de travail ..... 37
6.2.3 Créer et partager des lecteurs réseau ..... 38
**6.3 INSTALLATION DE INTEL MPI SOUS WINDOWS 7 ..... 39**
6.3.1 Documentation ..... 39
6.3.2 Installation sous Windows 7 ..... 39
6.3.3 Test de compilation ..... 40
6.3.4 Enregistrement d'un mot de passe avec WMPIRegister ..... 40
6.3.5 Exécution par wmpiexec ..... 40
6.3.6 Installation en cluster ..... 41
6.3.7 Configuration avec WMPICONFIG ..... 41
**6.4 DÉBOGAGE DE PROGRAMMES MPI AVEC VC2010 ..... 41**

7 ANNEXES ..... 42
**7.1 TORTOISE SVN ..... 42**

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

**7.2 GIT SVN ..... 42**
7.2.1 Git SVN ..... 42
7.2.2 Git bundle ..... 43
**7.3 VISUAL C++ 2010 : NETTOYAGE DE PROJETS ..... 43**
7.3.1 Fichiers de Visual Studio 2010 ..... 43
7.3.2 Produits de compilation ..... 43
7.3.3 Fichiers pour Visual Studio 6.0 ..... 44
7.3.4 Problème de cohérence de GUIDs ..... 44
7.3.5 Débogage avec Visual Studio ..... 45
**7.4 INSTALLATION DE NETCDF POUR WINDOWS ..... 45**
7.4.1 Compilation de NetCDF avec Visual Studio ..... 45
7.4.2 Compilation de NetCDF-cxx avec Visual Studio ..... 47

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

# 1 INTRODUCTION

Ce document est un élément de la livraison finale du projet «Réalisation d'une interface graphique pour Evariste».

Il concerne principalement les développeurs de cette interface graphique qui veulent utiliser le code de calcul sous-jacent.

Il décrit les choix d'implémentation, l'utilisation du code pris comme référence, et la structure du code adoptée pour le moteur de calcul.

Ce document a été mis à jour le 2017/09/21 pour correspondre à la révision 279 du code sur le serveur SVN du projet EVARISTE.

## 1.1 CONVENTIONS

Les conventions typographiques sont les suivantes :

- *noms de fichiers en italique* ;
- `commandes et fonctions en police de largeur fixe` ;
- des noms de répertoires ou des variables systèmes seront encadrés de %[1], comme *%PATH%*, %MER%, *%KANT%* ;
- des paramètres de compilation seront préfixés par $, comme *$Configuration* ou *$Platform*[2].
- La notation *{a,b}{c,d}*, à l'instar des développements en algèbre, désignera les quatre chaînes *ac*, *ad*, *bc*, *bd* ; on utilisera beaucoup cette notation pour des noms de répertoires comme *$Platform/$Configuration* avec par exemple *{Win32,x64}/{Release,Debug}* ;
- A l'instar de ce qui se fait dans la plupart des shells Windows ou Unix, «*» désigne n'importe quel chaîne de caractères. On généralise cela pour que par exemple «*»/ désigne n'importe quel répertoire, et «*/*.h» désigne n'importe quel fichier de n'importe quel sous-répertoire, terminé par «*.h».

## 1.2 CODE SOURCE

Notons %SVN% l'adresse actuelle de la machine qui héberge les développements SVN du BEPAG.

- Actuellement : %SVN%=https://a12w1015.proton.intra.irsn.fr:443/svn

Les dépôts SVN concernés sont :

- Pour le projet Evariste : *SVN_EVARISTE=%SVN%/EVARISTE1300/trunk ;*
- Pour le projet MER : *SVN_MER=%SVN%/MER1300/trunk* ;
- Pour le projet *BASE_MERCOR : SVN_MERCOR=%SVN%/BASE_MERCOR/trunk* ;

Les copies locales des dépôts SVN seront nommées :

- EVARISTE_SRC pour SVN_EVARISTE
- MER_SRC pour SVN_MER

Pour l'heure, aucune branche de développement particulière n'a été créée (le PC de développement gérant des branches locales grâce à git-svn) ;

On parlera de révision pour une branche accompagnée d'un numéro de version.

[1] A l'instar des variables dans les scripts .bat

[2] A l'instar des dialogues de configuration de Visual Studio qui utilisent *$(Platform)* et *$(Configuration)*.

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

### 1.3 SIGLES ET ACRONYMES

Voici quelques acronymes couramment rencontrés au BEPAG.

DSL : Documents de Spécification Logicielle

DFT : Documents Fonctionnels de Test

DCL : Documents de Conception Logicielle

MUL : Manuel d'Utilisation du Logiciel

VS, VS 6.0, VS 10, VS2010 : Visual Studio, Visual Studio 6.0, Visual Studio 10, Visual Studio 2010

VC++ 6.0, VC++ 2010 : Visual C++ 6.0, Visual C++ 2010

### 1.4 RÉFÉRENCES

- [MCR] – Documentation Mercor – «*%SVN%/BASE_MERCOR/trunk/documentation/NS2-09-12 - Documentation MERCOR.doc*», consulté à la révision SVN 26 ;
- [MPX] – Manuel utilisateur de PX - «*%SVN%\EVARISTE1300\trunk\documentation\MUL\docs pX-consX\nt_MUT_px1.1.doc*», consulté à la révision SVN 203 ;
- [MCX] – Manuel utilisateur de ConsX - «*%SVN%/EVARISTE1300/trunk/documentation/MUL/docs pX-consX/docs consx/consx 1.0 Manuel Utilisateur 08-66.doc*», consulté à la révision SVN 203 ;
- [RMH] – *Recursive Make Considered Harmful* – Peter Miller - AUUGN Journal of AUUG - 1998 - pp 14-25; disponible en ligne sur http://aegis.sourceforge.net/auug87.pdf.

## 2 MOTEUR DE CALCUL

### 2.1 COMPILATION ET INSTALLATION

- Les directives les plus pertinentes pour la compilation se trouvent dans la documentation produite par Doxygen. Elle glane ses informations dans :
  - Les emplacements cités par le fichier *%SVN_EVARISTE%\projet\src\calcul\Doxyfile*
  - Les sources *%SVN_EVARISTE%\projet\src\calcul* ;
  - Les pages associées *%SVN_EVARISTE%\documentation\DCL\CALCUL_EPS2*, notamment *Evariste-install.md* ;
- La documentation est lisible dans les fichiers de texte **.md*, mais est plus facilement consultable après génération des pages HTML correspondantes :
  - *cd "%SVN_EVARISTE%\projet\src\calcul"*
  - *doxygen*
- Dans le répertoire *html_doc/* produit, ouvrir « *index.html*». Cette page, produite à partir de *architecture.md*, donne les informations d'architecture principale. La rubrique la rubrique «*Pages associées*» mène vers les pages associées aux autres fichiers **.md* :
  - «*IHM Evariste : Architecture*» : concerne l'interface graphique implémentée en Java
  - «*Evariste : compilation/installation*» : contient les directives d'installation et de compilation, principalement pour le moteur développé en C++ ;
  - «*Moteur de calcul Evariste : spécifications*» : décrit le langage de commande du moteur. Utilisé par l'IHM et par ceux qui voudraient utiliser le moteur dans un script.
  - «*Les modules C++ PX et ConsX*» : notes sur la structure (reprise de l'existant) des classes C++ qui permettent l'utilisation des programmes PX et ConsX ;
- Une documentation propre à l'interface graphique en Java est produite par l'outil Javadoc à partir des sources de l'IHM :

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- La façon de compiler le projet Java pour l'IHM est aussi décrite dans le fichier *Evariste-install.m*d mentionné plus haut.

## 2.2 LIGNE DE COMMANDE

L'exécutable produit s'appelle CALCUL_EPS2.exe et interprète ses arguments ainsi :

- D'abord en lisant le contenu de la ligne de commande ;
- Ensuite en lisant un fichier d'entrée, si cela a été spécifié sur la ligne de commande ;

Une option ou un argument est n'importe quelle chaîne de caractères non vide délimitée par des espaces, des tabulations ou des fins de lignes.

L'implémentation privilégie les options dont le nom commence par un tiret.

Les commandes et options sont détaillées dans le fichier Evariste-specs.md ; elles sont assez nombreuses, et quelques-unes parmi celles implémentées ressemblent à :

- `-wait` : entre dans une boucle infinie jusqu'à ce qu'un débogueur l'en sorte ; cette option est naturellement réservée à la phase de développement ;
- `-input <fichier>` : continuer la lecture des arguments en lisant <fichier> ; si <fichier> vaut « stdin », c'est l'entrée standard qui est lue ; cette option ne peut être indiquée qu'une seule fois.
- `-merxml <fichier>` : chemin du fichier XML de configuration de MER ;
- `-merjdd <fichier>` : chemin du fichier XML des données à prendre en compte au prochain calcul par MER ;
- `-mer-sorties-set <n> <fichier1> <filtre1> …` : noms des fichiers que MER doit produire, avec le filtre à utiliser pour cela ;
- `-mer-calculer` : démarre immédiatement un calcul MER en utilisant les arguments fournis auparavant ;

Certaines options (en particulier celles qui démarrent un calcul) s'exécutent immédiatement.

L'idée est de pouvoir chaîner les opérations et de fournir une certaine souplesse à l'IHM.

## 2.3 CHOIX ET TECHNIQUES D'IMPLÉMENTATION

### *2.3.1 STRUCTURE DU CODE*

Le code propre au moteur de calcul est traité par un nouveau projet *CALCUL_EPS2* :

- Le fichier *CALCUL_EPS2.vcxproj* se trouve dans le même répertoire *workspace/* que le projet *IHM_EPS2* correspondant à l'interface graphique existante ;
- Le code source se trouve dans *src/calcul*, frère du répertoire *src/gui* ; sa structure est similaire :
  - *impl/* : fichiers d'implémentation des classes, fonctions et traitements ;
  - *inc/* : fichiers de déclaration des classes et objets
- Le code source est parti d'une copie de celui de *src/gui* ; toutefois il abandonne presque totalement l'utilisation de cet ancien code, pour n'en réimplémenter les fonctions utiles que dans deux fichiers principaux :
  - *main.cpp* : interprète les entrées et la ligne de commande ;
  - *calcul.cpp* : implémente une classe `CCALCUL` chargée d'appeler effectivement les modules `MER`, `PX`, `ConsX` ;
- Les modules correspondant aux logiciels tiers sont :

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- `CInterfaceMER` : implémentée à la place de l'ancienne classe `CCalcuRejetsEPS2` (qui appelait rejet.dll), cette classe appelle MER.exe et est basée sur la classe `EXE` du logiciel Kant, après renommage ;
- `pX` : le module existant est conservé ;
- `consX` : le module existant est conservé.

### *2.3.2 IDÉE GÉNÉRALE*

- Les fichiers implémentant des éléments purement graphiques ont été supprimés de calcul/ ;
- Les fichiers *calcul.cpp* ou *main.cpp* ont petit à petit hérité des fonctions (hors graphiques) qui étaient dévolues aux anciens *IHM_EPS2.cpp* et *EPS2Doc.cpp*, dont les objets (trop liés aux MFCs) ne sont plus utilisés) ;
- Le fichier de la classe `LectureIni` est conservé pour lire les fichiers de configuration : son portage sous Linux est élémentaire.
- Les arguments de la ligne de commande sont interprétés par `main()` ; l'implémentation devant être compilable à la fois sous Windows et Linux, on n'est pas passé par le `getopt()` propre à Linux, ni par des fonctions propres à Windows, mais par une classe auxiliaire `c_arg_provider` implémentée pour l'occasion et permettant de traiter des séries de commandes complexes.
- Les actions spécifiées dans les options de la ligne de commande pourront aussi être fournies via un fichier d'entrée (évite de répéter les options, et facilite l'écriture de fichiers de test) ;

### *2.3.3 REMPLACEMENT DES MFC PAR LA BIBLIOTHÈQUE STANDARD C++*

Quelques aspects mineurs de MER, pX, consX ont demandé des modifications pour passer des MFC de Microsoft à la bibliothèque standard du C++.

Le développement se déroule principalement avec Visual Studio 2010. Il faut donc utiliser un standard C++ antérieur (ici C++ 98).

- *calcul_common.h* : ce fichier remplace le *stdafx.h* et *afxtempl.h* de Microsoft, pour fournir les en-têtes système les plus utilisées comme `<vector> <string> <map>` … :
  - permet d'utiliser directement les remplacements `vector`, `string`, `map`
  - évite StdAfx que Microsoft interprête à sa façon.
  - `#include <vector> //remplace CArray`
  - `#include <map>    //remplace CMap`
  - `#include <string>  //remplace CString`
  - `#include <sstream> //remplace CString par stringstream`
  - définit `FALSE`, `TRUE`, `BOOL`

Remplacement de `CString` par `std::string` et `std::stringstream` :

- par `string` dans les cas simples
- par `stringstream` pour permettre des effets proches de CString::Format
- par rien ... quand le code n'a pas d'utilité pour un moteur de calcul.

Remplacement de `CMap` par `std::map`

- `CMap<key_t,key_t,val_t,val_t>` ==> `map<key_t,val_t>`
- `POSITION` ==> `map<...>::iterator`
- `pos = cm.GetStartPosition()` ==> `pos = cm.begin()`

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- `while(pos!=NULL)` ==> `while(pos!=cm.end())`
- `cm.GetNextAssoc(pos,key,val)` ==>
  - `cm.Lookup(key,val)` ==> `pos = cm.find(key); val=pos->second;` //si 'key' est absent, `pos` vaut `cm.end()`
  - `key=pos->first; val=pos->second; ++pos;`

Remplacement pour `CArray` par `std::vector` ou `std::array`

- `CArray<val_t,val_t>` ==> `vector<val_t>`
- `ca.GetSize()` ==> `ca.size()`
- `ca.SetAt(i,v)` ==> `ca[i] = v`
- `v=ca.GetAt(i)` ==> `v=ca[i]`
- `ca.Add(v)` ==> `ca.push_back(v)`

Remplacement de `CFile` par `FILE*`

- `cf.Open(...)` ==> `pf = fopen(...)`
- `cf.Close()` ==> `fclose(pf)`
- `cf.Write(buf,len)` ==> `fwrite(buf,1,len,pf)`
- Un remplacement au moins aussi simple est possible avec `istream`, `ostream`, `iostream` en fonction des circonstances ;

Techniques de remplacement `STARTUPINFO`, `PROCESS_INFORMATION`, etc :

- Ce code est lié à l'interface graphique.
- Les sections ayant recours à ces identifiant se content en fait d'un équivalent de l'appel `system(...)`

Autres remplacements:

- `DEBUG_NEW` : seulement nécessaire en débogage des MFC. Supprimé pour l'instant.

## 2.4 PORTAGES VERS LINUX

### *2.4.1 CRÉATION D'UN MAKEFILE*

#### 2.4.1.1 Contenu du Makefile

La structure retenue pour le *Makefile* de MER a été fortement inspirée de celle suggérée dans [RMH].

- Un fichier *Makefile* principal pour accéder aux outils de compilation et aux bibliothèques ;
- Un fichier auxiliaire *sources.mk* (le nom importe peu) produit en extrayant les noms des fichiers *.cpp* mentionnés dans le fichier de projet Visual C++ 2010 :

  Commande:
  ```
  cat ../workspace/MER/MER.vcxproj | \
  sed '/cpp/ !d; s/\\/\//g; s/<ClCompile Include="//; s/" \/>//' | \
  sed '/^<Import/ d; s/^ *//; s|^../../src/||1; /StdAfx.cpp/ d;' >sources.mk
  ```

  La commande d'extraction se décompose comme suit:
  - lister les lignes du du fichier `.vcxproj` ;
  - ne garder que celles contenant un .cpp, y remplacer `\` par `/`, extraire le chemin ;

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- enlever les faux positifs (`<Import`); les espaces et `../../src` initiaux; Négliger `StdAfx.cpp`
- Ecrire les noms dans `sources.mk`

- Après exécution de la commande, une mise en forme manuelle élémentaire du fichier lui permet d'être compris par `make`.

### 2.4.1.2 Compilation du moteur avec le Makefile

La version Linux du moteur a été compilée dans l'environnement suivant, que l'on supposera installé (sur la machine FARUX) :

- NetCDF 4.4.1 : des instructions d'installation sont dans *Evariste-install.md*
- NetCDF-cxx-4.2 : des instructions d'installation sont dans *Evariste-install.md*
- g++4.4.7 : celui proposé par le système

Le Makefile utilise plusieurs variables d'environnement qu'il faut définir :

- Définir les variables d'environnement pour NetCDF :
  - `NETCDF_ROOT=…`
  - `export NETCDF_CFLAGS=-I$NETCDF_ROOT/include`
  - `export NETCDF_LDFLAGS=-L$NETCDF_ROOT/lib`
  - `export NETCDF_LIBS=-lnetcdf`
- Définir les variables d'environnement pour NetCDF-cxx :
  - `NETCDF_CXX_ROOT=…`
  - `export NETCDF_CXX_CFLAGS=-I$NETCDF_CXX_ROOT/include`
  - `export NETCDF_CXX_LDFLAGS=-L$NETCDF_CXX_ROOT/lib`
  - `export NETCDF_CXX_LIBS=-lnetcdf_c++`

Passons aux commandes de compilation proprement dites :

- `cd %EVARISTE_SRC%/projet/src`
- `make`

Cela doit produire un fichier CALCUL_EPS2.exe dans le répertoire courant

### *2.4.2 PRINCIPALES MODIFICATIONS*

Ce paragraphe présente des points communs aux portages de MER et du moteur de calcul.

Afin que les changements ne modifient pas le comportement sous Windows, les modifications incompatibles avec Windows sont encadrées par une directive de préprocesseur #ifdef _WIN32.

Changements faits pour Linux (avec compilateur g++ 4.4.7) :

- Les fonctions et symboles qui existent sous Windows et pas sous Linux sont regroupés dans un fichier «*linux_port.h*» ;
- Comme Linux prend en compte la casse des noms de fichiers, certaines majuscules ont été corrigées dans les directives `#include` de certains fichiers. C'est notamment le cas de `Math.h` et `ACoeur.h`, remplacés par `math.h` et `Acoeur.h` ;
- Des prototypes de fonctions semblent connus par défaut de Visual Studio, mais pas de gcc : ainsi, dans *util/{LoiLogNormale.cpp,LoiNormale,cpp,Maths.cpp,Number.cpp}*, il a fallu inclure `<math.h>` `<cstdio>`, `<cstdlib>` pour les fonctions `printf()` `abs()` `atoi()` `atof()` `exit()` ;
- Un fichier demandait aussi `<cvode>` ;
- Le compilateur g++ refuse les syntaxes ambigües pour l'instanciation de templates. Ainsi, dans les instanciations de template imbriquées, des « >> » finaux ont été remplacés par la terminaison non-ambigüe «> >», comme dans *CalculerCsqRadio.cpp* ;
- Le fichier *linux_port.h* définit les fonctions et symboles suivants :
  - équivalent inline passable de `GetTickCount()`
  - `__declspec()` pour le supprimer (cf *sys_utils.h*).

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- g++4.4.7 refuse d'utiliser des références comme itérateurs. En particulier, la syntaxe «`const_iterator& = fonction()` » cause une erreur « `invalid initialization of non-const reference` », qui disparaît si on utilise simplement «`const_iterator = fonction()`» ;
- g++4.4.7 refuse les déclarations «`classe::membre()`» à l'intérieur de la déclaration de «`classe`»:
  les prototypes «`classe::membre()`» ont donc été remplacés par «`membre()`», comme dans le fichier *CalculerCoeffDepotAux_Persan.h*

### *2.4.3 PARTICULARITÉS DU PORTAGE DE MER*

#### 2.4.3.1 Sélection du code source

Le dépôt SVN du code MER occupe plusieurs dizaines de GB, dont seules quelques centaines de MB sont utiles à la production de l'exécutable (voir la remarque sur les checkout partiels de SVN au §7.1).

En l'occurrence, il a suffi de copier quelques portions du répertoire projet/ vers Farux pour servir de base au portage :

- *projet*/ : 4.47 GB à l'origine
  - *installation*/: 173 MB --> viré (c'est pour InstallShield)
  - *modele*/: 2MB
  - *Outils*/ : 158 MB pour Xerces et CVODE; 40 MB après nettoyage
    - XERCES
      - Build/ : compilation propre à Windows, supprimée.
    - cvode-2.7.0/
      - Release/ :Compilation propre à Windows, suprrimée
  - *Specification*/ : 3.89 GB supprimés avec empressement
  - *src*/ : 5.11 MB
  - *UnitTest*/ : 10.8 MB
  - *workspace*/ : contient quelques fichiers .cpp qui rendent ce répertoire utile hors de VS2010 ;
    - *REJET*/
      - *MerResult.h Main.cpp ... EPS2.h EPS2.cpp*;
      - **.vcx** : fichiers supprimés car propres à VS2010 ;
      - *conf*/ : fichiers utiles aux tests (mais pas à la compilation) ;
    - *MER_EXE*/
      - *parser*/
        - *ParserFacadeJDD.{cpp,.h}* ;
      - *MER_EXE.cpp* ;
    - *MER*/
      - **.vcx** : supprimé car propre à Visual Studio ;
      - *StdAfx.cpp* : inutile hors de Visual Studio ;
    - *bin*/<*arch*>/<*config*>/ : fichiers compilés inutiles hors de Windows ;
    - *conf*/ : fichiers XML et XSLT utiles pour les tests ;
  - *dll*/<*arch*>/<*config*>/ : fichiers compilés inutiles hors de Windows ;
  - *lib*/<*arch*>/<*config*>/ : fichiers compilés inutiles hors de Windows ;

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- *documentation/* : 199MB inutiles pour la compilation ;
- *conf/* : nombreux fichiers XML, sans doute utiles pour tester ;
- *bin/<arch>/<config>/* : fichiers compilés inutiles hors de Windows ;

#### 2.4.3.2 Dépendances CVODE et Xerces

Les deux bibliothèques CVode et Xerces ont été recompilées sans aucun problème sur la machine Farux, avec le compilateur gcc-4.4.7 du système.

Le code source utilisé est celui présent dans le sous-répertoire *Outils/* du code de MER (à la révision 1966).

##### 2.4.3.2.1 Compilation de Xerces

Parmi les fichiers copiés, certains sont des scripts shell dont les droits d'exécution ont pu être perdus par l'opération d'archivage. Pour remettre ces droits aux bons fichiers sans y passer trop de temps, on les accorde à tout fichier reconnu comme exécutable par l'utilitaire standard `file` :

```
cd [MER]/projet/Outils/xerces-3.1.1/xerces-c-3.1.1
find . -type f | while read f; do { file "$f" | grep -q executable; } && chmod a+x "$f" ; done
XERCES_ROOT=..... #Au choix
./configure --prefix=$XERCES_ROOT
make -j 4
make install
```

##### 2.4.3.2.2 Compilation de CVODE

Comme pour Xerces, les fichiers archivés dans le dépôt SVN avaient parfois perdu leurs droits d'exécution, que l'on a donc remis avec la même astuce :

```
cd [MER]/projet/Outils/cvode-2.7.0/cvode
find . -type f | while read f; do { file "$f" | grep -q executable; } && chmod a+x "$f" ; done
CVODE_ROOT=..... #Au choix
./configure –prefix=$CVODE_ROOT
make -j 4
make install
```

#### 2.4.3.3 Compilation du code

Nous avons en fait deux fichiers Makefile pour compiler MER.

Le premier produit une bibliothèque statique *MER.a* (équivalente au *rejet.lib* de produit sous Windows)

```
cd [MER]/projet/src
make clean #Le Makefile est celui produit à partir de MER.vcxproj
make –j8
```

Le second produit le MER.exe

```
cd [MER]/projet/workspace/MER_EXE
make clean
make
```

NOTE : le premier *Makefile* dépend du fichier *sources.mk* lui-même produit à partir de *MER.vcxproj* comme décrit dans §2.4.1 ; en particulier, si *MER.vcxproj* change (notamment si des fichiers sont ajoutés ou retirés dans le projet), il faudra mettre à jour *sources.mk* ;

#### 2.4.3.4 Tests et comportement

- Dans les fichiers *.mer* produits, les nombres indéfinis, apparaissent comme `-0.00IND000000e+000` sous Windows, et `-nan` sous Linux

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- Le script de test *diff_grille_txt.pl* est un script Perl, destiné à comparer le fichier obtenu et la référence, à l'instar du projet *MER_Ecarts* pour Windows.

# 3 EVARISTE : ÉTAT INITIAL DU CODE

Le code examiné et discuté ici est celui de la révision %EVARISTE_SVN% :223

## 3.1 UTILISATION DE L'IHM

Utilisation de bout-en-bout faite à partir de l'exécutable IHM_EPS2 recompilé pour x64 et NetCDF 4.4.1

### *3.1.1 REMARQUES PRÉLIMINAIRES*

- Ne JAMAIS fermer une fenêtre de l'IHM (car le menu «fenêtre» ne sait pas toutes les rouvrir). Par contre on peut les réduire.
- Le calcul en configuration DEBUG est très lent avec MER, mais c'est le seul que l'on arrive à faire avec la version recompilée.

### *3.1.2 CALCUL DES REJETS*

C'est l'appel à MER … qui se fait en fait via la DLL rejet.dll.

Ce n'est qu'après ce calcul que le lancement du calcul pX/consX devient possible.

### *3.1.3 DISPERSION*

Ouvrir la fenêtre « Paramètres d'exécution de pX».

C'est quand on est dans cette fenêtre que l'icône « pX->consX» s'active (en fait, elle ressemble vaguement à une gomme, tout à droite de la barre d'outils).

#### 3.1.3.1 Onglet : «Données météorologiques»

C'est le second onglet, mais il est presque plus commode de le remplir en premier.

- «Dimensions du maillage» : valeurs X, Y, Z à donner en mètres.
  - Pour chacune des trois listes X,Y,Z, les valeurs sont à entrer par ordre croissant
- «Instants météo» : temps en s, à entrer par ordre croissant en faisant «Ajouter» pour chaque valeur.
- Bouton «Valider» : remplit les 2 listes en bas de la fenêtre :
  - Première liste : vitesse du vent :
    - Une ligne par quadruplet (Instant,X,Y,Z) :
    - Les champs de vitesse U,V,W sont initialisés par défaut (pour un vent de 2m/s dans une direction à 45° de l'axe des abscisses).
    - Les valeurs U,V,W sont éditables ;
    - Les valeurs Instant, X, Y, Z ne le sont pas ;
  - Seconde liste : stabilité pluviométrique :
    - Une ligne par triplet (Instant,X,Y)
    - Coefficients Rain, Stabilité, HCLA initialisés par défaut.
    - Les champs Rain, Stabilité, HCLA sont-éditables ;
    - Les champs Instants, X, Y ne le sont pas.

#### 3.1.3.2 Onglet : «Paramètres d'exécution de pX»

- Date Calcul : conditions à remplir :

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

    - Date début : aucune condition particulière, sinon que tous les champs soient remplis ;
    - Date fin : doit être postérieur à Date début ;
    - Date fin : ne doit pas suivre Date début de plus de s que le plus grand «Instant» indiqué dans l'onglet «Données météorologiques» ;
- Mode : sélectionner «Standard» pour initialiser certains champs de l'onglet par défaut ; ne sert à rien d'autre.
- Modèle physique :
    - Modèle de diffusion : je me contente de ce qui est mis par le mode «Standard» ;
    - Facteur de battement : idem ;
    - Coeff. Diffusivité horizontale : idem ;
    - Coeff Diffusivité verticale : idem ;
- Pas de temps :
    - Pas de temps ADR :
        - Prendre une valeur inférieure au pas de temps utilisé pour le calcul des rejets ;
        - TODO : autre condition ?
    - Fréq. d'appauvrissement des bouffées : j'utilise la valeur mise par «Standard» ;
    - Fréq. de filiation des bouffées : idem ;
    - Fréq. du calcul de dépôt : idem ;
    - Fréq. du calcul des valeurs intégrées : idem;
- Précision du calcul
    - Formule de gauss : j'utilise ce qui est mis par «Standard»
    - Sueil gauss x/sigma : idem ;
- Aide facultative : ne pas toucher, ça fait planter Evariste.

#### 3.1.3.3 Onglet «Résultats demandés»

- Distances demandées : entrer une liste croissante de valeurs, en m (elles ne doivent pas sortir de la zone couverte par les données météo) ;
- Angles demandés : entrer une liste croissante, en degrés ;
- Instants demandés : les entrer par ordre croissant, en s (ils ne doivent pas sortir de la zone couverte par les données météo) ;
- Clic sur le bouton «Générer maillage» ;
- Le bouton devient «Maillage généré» ;

#### 3.1.3.4 Démarrage effectif du calcul

- Clic sur l'icône «calcul pX-consX» (celle qui a la tête de gomme) ;
- Un calcul MER est (re) lancé
- Les fichiers sont créés dans Px<version>/datafile/
- *px.bat* : créé à l'issue du calcul MER, contient la commande d'invocation de l'exécutable px ;
- *masource.nc* : créé à l'issue du calcul MER
- *mameteo.nc* : créé à l'issue du calcul MER (un fichier *mameteo.cdl* de l'installateur semble inutilisé) ;
- *monformat.nc* : créé à l'issue du calcul MER
- *mesres.nc* : créé à l'issue du calcul pX

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

#### *3.1.4 CONSÉQUENCES RADIOLOGIQUES*

Le calcul s'exécute dans la foulée de pX : les paramètres demandés sont les mêmes.

### 3.1.4.1 Fenêtre «Gestion des conséquences radiologiques»

- Champ «Type» :
- Champ Organe :
- Champ «Voie d'exposition» :
- Champ «Instantanées ou intégrées» :
- Champ «Cibles» :
- Champ «Type sommation isotopes» :
- Isotope :
- Cadre «Choix courbe» :
    - Champ «Titre de graphique» : ne PAS le remplir (concerne la bibliothèque AxGC) sous peine de plantage
    - Champ «Instant»
    - Champ «Distance»
    - Champ «Angle»

### 3.1.4.2 Fichiers utilisés et produits

Dans %CONSX% :

- *consx.bat* : écrasé avant le début du calcul consx ;

Les fichiers concernés sont dans %CONSX_HOME% :

- *mesparams.txt* : liste des calculs demandés ; attention, les chemins (notamment celui de pX) sont lus depuis la base de registres (cf §3.2.1);
- *mesparams_incertitude.txt* : produit, que le calcul pX ait été fait avec incertitude ou non d'ailleurs. C'est *consx.bat* qui décide d'appeler consx avec *mesparams.txt* ou *mesparams_incertitude.txt* ;
- *mesres_consx.nc* : fichier de résultats pour consX ; voir §4.4.4.1 ;

## 3.2 FICHIERS DE L'IHM

#### *3.2.1 BASE DE REGISTRES*

Les données sont stockées dans «HKEY_CURRENT_USER\Software\EVARISTE Application» :

- IHM_EPS2
    - CONSX
        - CHAR
            - "NMAXCHARCONSX"=dword:00000040
        - CIBLE
            - "UNAN"="1an", "ADULTE"="adulte"
        - EXPOSITION
            - "PANACHE"="Panache", "EFFICACE"="Efficace", "DEPOT"="Depot", "INHALATION"="Inhalation", "ACTDEP"="Activite"
        - INSTORINTEG:
            - "INTEG"="Integ", "INST"="Inst"

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- IO: "NETCDF"=dword:00000001, "TXT"=dword:00000000
- ORGANE: "THYROIDE"="Thyroide", "EFFICACE"="Efficace", "SEIN"="Sein", "POUMON"="Poumon", "OS"="Os"
- PATH: "CONSX_PARAM_PATH"="data\\mesparams.txt", "PX_RES_PATH"="D:\\evariste\\\\Px-1.1.0.139\\dataFile\\mesres.nc", "CONSX_BAT_PATH"="D:\\evariste\\\\ConsX\\consx.bat", "CONSX_DEF_PARAM_PATH"="data\\consx.defparam", "CONSX_RES_PATH"="data\\mesres_consx.nc", "CONSX_PARAM_INCER_PATH"="data\\mesparams_incertitude.txt", "HOME_PYTHON"="D:\\evariste\\\\ConsX\\Python24\\", "HOME_CONSX"="D:\\evariste\\\\ConsX\\Python24\\Lib\\site-packages\\consx-1.0.3_r1062-py2.4.egg\\consx\\", "CONSX_CONSOLE_PATH"="D:\\evariste\\\\ConsX\\consx.log"

- DERNIER_DOCUMENT : "TYPE_DERNIER_FICHIER"=dword:00000000, "DERNIER_FICHIER"="….eps2"
- FICHIER_DEFAUT
- MER
- PX
  - CHAR
    - "NMAXCHAR"=dword:00000040
  - DEFAULTCONFIG
    - "FREQINT"=dword:00000001
    - …
  - IO
    - "NETCDF"=dword:00000001
    - …
  - MODELE
    - "BRIGGS_RURAL"=dword:00000001
  - PATH
    - "HOME_PX"="D:\\evariste\\\\Px-1.2.1.223\\"
    - "PX_CONSOLE_PATH"="console.txt"
    - "PX_EXE_PATH"=".\\bin\\px.exe"
    - "PX_REF_PATH"="dataFile\\Ref_BEPAG.pxs"
    - "PX_BAT_PATH"="px.bat"
    - "PX_PARAM_PATH"="mesparams.txt"
    - "PX_RES_PATH"="dataFile\\mesres.nc"
    - "PX_TS_PATH"="dataFile\\masource.nc"
    - "PX_METEO_PATH"="dataFile\\mameteo.nc"
    - "PX_FAMILLES_PATH"="dataFile\\familles_BEPAG.pxs"
    - "PX_DEF_PARAM_PATH"=".def_param"
    - "PX_MONFORMAT_PATH"="dataFile\\monformat.nc"
  - RUNMODE

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- "DB"=dword:00000000, "RV"=dword:00000002, "RS"=dword:00000003 "ALL"=dword:00000001

- Recent File List:
  - "file1"="…"
  - File2=…
- Settings

### *3.2.2 FICHIERS .EPS2*

A synchroniser avec le fichier .xml de MER pour certaines sections.

- Version : VERSION4
- [VARIABLES_FPA]: cf .xml d'entrée de MER
  - <Nom> <nb_modalites>
- [PARAMETRES_INCERTAINS] : cf .xml d'entrée de MER. Le format de chaque ligne est
  - Nom <val1> <val2> <NumeroLoi> <coef1> … <coef4>
  - Le numéro de loi est donné par le tableau LoisDefaut défini dans Loi.cpp ;
  - <val1> et <val2> sont des valeurs remplies par l'utilisateur dans l'IHM
  - <coef1> … <coef4> sont présents quelle que soit la loi : si elle utilise moins de paramètres, ils viennent en dernier.
  - ➔ on n'a pas vraiment à se préoccuper de cette section, qui est écrite après un tirage.
- [PARAMETRES_CALCUL]
- [TERMES_SOURCES]
- [TIRAGES]
- [CONSOLE]
- [CDATARES]
- [WINDOW_POS]
- [SPLITTER_POS]
- [PARAMPX]
- [RESPX] : distances, angles, instants transcrits dans
- [METEOPX]

## 3.3 INSTALLATIONS ET RÉPERTOIRES DE BASE_MERCOR ET EVARISTE

### *3.3.1 BASE_MERCOR*

- *bin/{BASE_MERCOR.exe,netcdf.dll}* : l'un compilé du projet, l'autre récupéré comme on peut.
- *ConsX/*
  - *Consx.bat* : script générique de lancement, non-modifié (chemin en dur vers ConsX/) ;
  - *consx.log* : log du dernier calcul … eh oui.
  - *Python24/Lib/site-packages/*
    - *consx-1.0.3_r1062-py2.4.egg/* : MODIFIE lors d'un calcul
      - *consX.defparam*
      - *consX.userparam*

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- *ICRP38.RAD*
- *mesparams.txt* : écrit pendant le calcul
- *mesparams.txt_save.txt* : écrit pendant le calcul
- *mesparams_incertitude.txt* : écrit pendant le calcul
- *mesres_consx.nc* : résultats du calcul consx

▪ *dosecoef-1.0.3-r1274\dosecoef\data* : ne semble pas modifié lors d'un calcul

- *CoefPX.pxs*
- *CoefPX_BEPAG.pxs*
- *CoefPX-ref_DD.pxs*
- *SolubDefault.pxs*
- *SolubDefault_BEPAG.pxs*
- *SolubDefault-ref_DD.pxs*

o *Parametres_BASE_MERCOR/*

▪ *BASE_MERCOR.ini* : chemins (en dur) de pX, consX, etc. (TODO : spec ?)

▪ *mesSources.txt* : entrée de BASE_MERCOR

o *Parametres_CONSX/*

▪ *Consx.ini* : chemins (en dur) Python, PX_RES_PATH, consx.bat, …

▪ *mesparams_consx.txt* : description des tâches consX (cf §4.4.3.1 «Le fichier de paramètres»), changée en cours de calcul

o *Parametres_PX/*

▪ *mameteo.txt* : sert à produire entrée netcdf ?

▪ *mesparams_px.txt* : début/fin/tirages/loi/…

▪ *monformat.txt* : distances, angles, instants où produire les résultats

▪ *Px.ini* : chemins (en dur) de l'installation Px, non-modifiables

o *Px-1.1.0.139/*

▪ *mesparams.txt* : copie de mesparams_px.txt ?

▪ *param* : comme 'mesparams.txt' mais avec accents ? ignoré ?

▪ *console.txt* : écrasé à chaque nouveau calcul px ?

▪ *bin/{ncdump.exe,netcdf.dll,px.exe}*

▪ *dataFile/*

- *familles.pxs* : non-modifié pendant calcul
- *familles_BEPAG.pxs* : non-modifié pendant calcul
- *mameteo.nc* : transcription de mameteo.txt pendant le calcul
- *masources.nc* : transcription de mesSources.txt ( ?)
- mesres.nc : cf 4.3.2 «Les fichiers statiques traités par pX»

### *3.3.2 LES FICHIERS STATIQUES POUR PX*

Il s'agit là des fichiers que Evariste ne modifie pas

#### 3.3.2.1 Le fichier de filiation

Fichier de texte tabulé (familles_BEPAG.pxs), mentionné dans [MPX] §3.2.8 ; Structure non-documentée à cet endroit (où d'ailleurs ?) mais aisément reconstituable :

- Ligne 1 : nombre d'éléments décrits

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- Lignes suivantes : champs séparés par des blancs (espace ou tabulation) :
  - Index
  - Nom de l'élément
  - Index de chaque élément obtenu dans la chaîne de décomposition.
  - La valeur 0
- L'élément d'index 0 est nommé «Scalar» et sa décomposition est limitée à l'élément 0.

#### 3.3.2.2 Le fichier de radio-éléments

Fichier de texte tabulé (Ref_BEPAG.pxs) :

- Structure non-documentée dans [MPX] §3.2.8 et je ne vais pas me fatiguer à le faire.

### *3.3.3 LES FICHIERS TRAITÉS PAR PX*

- Utilisation de NetCDF par pX
  - *monformat.nc* : transcription de monformat.txt
  - *Ref.pxs* : inchangé lors du calcul (décroissance radioactive)
  - *Ref_BEPAG.pxs* : inchangé lors du calcul
- *RESULTATS_BASE_MERCOR/*
  - *FonctionsTransferts_<famille>.txt* : fichier XML par <Isotope>
- *TS/* : un répertoire par élément de mesSources.txt ?
  - *masource<n>/*
    - *data/* [Résultats fournis par ConsX ?]
      - *console.txt* : copie de celle de ConsX/
      - *mesparams.txt* : copie de celui de ConsX/
      - *mesres_consx.nc* : cf §4.4.4.1 «Format de fichier pour ConsX» ;
      - *mesres_consx.nc.cdl* : conversion du précédent par ncdump ?
    - *datafile/* [Entrées et sorties de pX ?]
      - *familles_BEPAG.pxs*
      - *mameteo.nc{,.cdl}* : meteo fournie à pX
      - *masource.nc{,.cdl}* : termes sources fournis à pX ?
      - *mesres.nc{,.cdl}* : diffusion calculée par pX
      - *monformat.nc{,.cdl}* : points et instants des résultats souhaités ;
      - *Ref_BEPAG.pxs* : données de décroissance radioactive ;
    - *console.txt* : copie de la console pX du calcul ?
    - *mesparams.txt* : copie de celui de Px-1.1.0.139

### *3.3.4 EVARISTE*

- *ConsX/* : voir §3.3.1 «BASE_MERCOR»
  - *Python24/*
  - *consx.bat*
  - *consx.log*
- *Evariste/*
  - **.dll* : les dépendances DLL
  - *rejet.dll*, *netcdf.dll*, *xerces-c_3_1.dll* : à remplacer en fonction de MER et BASE_MERCOR
  - *conf/*

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- */*.xml*: fichiers de configuration[3] pour MER.
- *Data/Default/*
  - *ACCDIM/* : Accidents de dimensionnement ?
  - *<accident>/*.eps2* : Format genre *.ini* à spécifier (cf 3.2.1 «Base de registres»)
    - Fichiers .eps2 : Variables FPA, Paramètres incertains, Paramètres Calcul, Termes Sources, Tirages, Console, CDataRes, Window_pos, Splitter_pos, ParamPx)
  - *AG/* : Accident Grave : idem. Il semble que les fichiers *.eps2 doivent avoir une section *[VARIABLES_FPA]* conforme au *.xml* d'initialisation de MER (sinon erreur).
- *lib/*
  - *rejet.lib* : bibliothèque produite par le projet EPS2 du code MER ;

- *Px-1.2.1.223/*
  - *Bin/*
    - *px.exe*, *ncdump.exe*, *netcdf.dll* : compilation de Px
    - *msvcr71.dll* : dépend du compilateur Visual Studio qui a produit l'exécutable
  - dataFile/
    - *familles.pxs*
    - *familles_BEPAG.pxs*
    - *mameteo.nc{,.cdl}* : meteo fournie à pX
    - *masource.nc{,.cdl}* : termes sources fournis à pX ?
    - *mesres.nc{,.cdl}* : diffusion calculée par pX
    - *Ref.pxs, Ref_BEPAG.pxs* : données de décroissance radioactive ;
  - *.def_param*, *.def_testparam* : argument --def_param_file dans *px.bat*
  - *console.txt* : argument --message_file dans *px.bat*
  - *mesparams.txt* : argument --param_file dans *px.bat*
  - *px.bat* : cd . ; px.exe … ;

### *3.3.5 BASE DE REGISTRES*

Les données sont stockées dans «HKEY_CURRENT_USER\Software\EVARISTE Application» :

* IHM_EPS2
  - o CONSX
    - CHAR
      - "NMAXCHARCONSX"=dword:00000040
    - CIBLE
      - "UNAN"="1an", "ADULTE"="adulte"
    - EXPOSITION
      - "PANACHE"="Panache", "EFFICACE"="Efficace", "DEPOT"="Depot", "INHALATION"="Inhalation", "ACTDEP"="Activite"

[3] Semblent utiliser «#...» pour les commentaires. Tant mieux si cela fonctionne, mais les specs XML seraient plus en faveur de <!-- … --> ou de <?ignore … ?>

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- INSTORINTEG:
  - "INTEG"="Integ", "INST"="Inst"
- IO: "NETCDF"=dword:00000001, "TXT"=dword:00000000
- ORGANE: "THYROIDE"="Thyroide", "EFFICACE"="Efficace", "SEIN"="Sein", "POUMON"="Poumon", "OS"="Os"
- PATH: "CONSX_PARAM_PATH"="data\\mesparams.txt", "PX_RES_PATH"="D:\\evariste\\\\Px-1.1.0.139\\dataFile\\mesres.nc", "CONSX_BAT_PATH"="D:\\evariste\\\\ConsX\\consx.bat", "CONSX_DEF_PARAM_PATH"="data\\consx.defparam", "CONSX_RES_PATH"="data\\mesres_consx.nc", "CONSX_PARAM_INCER_PATH"="data\\mesparams_incertitude.txt", "HOME_PYTHON"="D:\\evariste\\\\ConsX\\Python24\\", "HOME_CONSX"="D:\\evariste\\\\ConsX\\Python24\\Lib\\site-packages\\consx-1.0.3_r1062-py2.4.egg\\consx\\", "CONSX_CONSOLE_PATH"="D:\\evariste\\\\ConsX\\consx.log"

o DERNIER_DOCUMENT : "TYPE_DERNIER_FICHIER"=dword:00000000, "DERNIER_FICHIER"="….eps2"

o FICHIER_DEFAUT

o MER

o PX

- CHAR
  - "NMAXCHAR"=dword:00000040
- DEFAULTCONFIG
  - "FREQINT"=dword:00000001
  - …
- IO
  - "NETCDF"=dword:00000001
  - …
- MODELE
  - "BRIGGS_RURAL"=dword:00000001
- PATH
  - "HOME_PX"="D:\\evariste\\\\Px-1.2.1.223\\"
  - "PX_CONSOLE_PATH"="console.txt"
  - "PX_EXE_PATH"=".\\bin\\px.exe"
  - "PX_REF_PATH"="dataFile\\Ref_BEPAG.pxs"
  - "PX_BAT_PATH"="px.bat"
  - "PX_PARAM_PATH"="mesparams.txt"
  - "PX_RES_PATH"="dataFile\\mesres.nc"
  - "PX_TS_PATH"="dataFile\\masource.nc"
  - "PX_METEO_PATH"="dataFile\\mameteo.nc"
  - "PX_FAMILLES_PATH"="dataFile\\familles_BEPAG.pxs"
  - "PX_DEF_PARAM_PATH"=".def_param"
  - "PX_MONFORMAT_PATH"="dataFile\\monformat.nc"
- RUNMODE

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- "DB"=dword:00000000, "RV"=dword:00000002, "RS"=dword:00000003 "ALL"=dword:00000001

o Recent File List:

- "file1"="…"
- File2=…

- Settings

## 3.4 STRUCTURE DU CODE SOURCE INITIAL

### *3.4.1 BASE_MERCOR*

Les projets du répertoire *Outils/* de ce dépôt sont directement utilisés par Evariste, au point qu'ils les référencent directement par des liens symboliques (ou plus exactement des raccourcis Windows).

- *librairie-3.6.1.lib* (il s'agit en fait de NetCDF)
- *netcdf-3.6.1_lib* (il s'agit en fait de l'interface C++ netcdf-cxx)

### *3.4.2 EVARISTE*

#### 3.4.2.1 Le code source

On ne considère ici que le répertoire *trunk/*, i.e. la branche principale, du dépôt SVN :

- *bin* : exécutables *IHM_EPS2.exe*, sans séparation par architecture.
- *dll* : *DLL rejet.dll*, sans séparation par architecture (cause des problèmes…).
- documentation: voir surtout MUL
- *lib/{Debug,Release}/* : fichiers produits par le répertoire Outils (attention : pas de $(Platform)
  - *librairie-3.6.1.lib*
  - *netcdf-3.6.1.lib*
- *Outils/* : liens vers le répertoire *Outils* de BASE_MERCOR
- *Projet*
  - *installation/* : diverses dépendances
  - *src/* : les sources en C++ des modules *px*, *consx*, *util*
    - *util/inc* : les fichiers *netcdf.h*, *ncvalues.h*, *netcdfcpp.h* DOIVENT correspondre à ceux du projet *netcdf-3.6.1_lib*
  - *UnitTest/* : vide
  - *ValidTest/* : vide
  - *workspace/* : l'espace pour les projets Visual Studio
    - *conf/* : les fichiers XML pour MER
    - *Release/*, *Debug/* : ces répertoires ne devraient pas être versionnés.

#### 3.4.2.2 Points de repère pour le débogage

Evariste est pour l'heure une application graphique MFC. Un bon point d'entrée est la fonction `CIHM_EPS2App::InitInstance` (appelé par le `WinMain`)

### *3.4.3 MER*

- *bin/{Win32,x64}/{Debux,Release}/MER_EXE.exe*
- *conf/* : fichiers de configuration pour l'exécutable *MER_EXE.exe*

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- *dll/$(Platform)/$(Configuration)/xerces-c_3_1.dll*
- *documentation/{DCL,DFT,DSL,MUL}*
- *gestion_projet/* : vide
- *lib/$(Platform)/$(Configuration)/* : apparemment copiés à la main ?
    - *CVODE-2.7.0.lib*
    - *MER.lib*
    - *rejet.lib*
    - *xerces-c_3_1.lib*
    - *cppunit.lib* #Seulement pour Win32 actuellement
    - *TestRunner.lib* #Seulement pour Win32 actuellement
- *projet/*
    - *Outils/* : codes sources (inchangés) des bibliothèques tierces
        - *cvode-2.7.0/* : code source original
            - *CVODE*.{sln,vcxproj} : solution et projet VS2010
        - *xerces-3.1.1/xerces-c-3.1.1* : code source original de Xerces
            - *projects/Win32/* : créé pour compiler Xerces avec VS
                - *lib/{Debug,Release}* : vide
                - *VC10/xerces-all/*
                    - *xerces-all.sln* : solution VS2010 pour Xerces ; configurations {ICU,Static}{Debug, Release} inutiles ?
                    - *XercesLib/*
                        - *XercesLib.vcxproj.**: projet VS2010
    - *Specification*
    - *src/*
        - *Makefile*, *sources.mk* : fichiers pour compiler sous Linux (SVN 1969)
        - *main.cpp*, */*.cpp */*.h …* : sources de MER
    - *ValidTest/* : répertoire de validation : on en reparle plus loin
    - *workspace/*
        - *EPS2/*.vcxproj** : produit *rejet.dll*
        - *MER/*.{sln,vcxproj}* : produit *MER.lib*
        - *MER_EXE/*.{sln,vcxproj}* : produit MER_EXE.exe
        - *REJET/* : semble obsolète (référence à des fichiers .cpp inexistants)
            - *conf/* : que font ces fichiers XML ici ?

# 4 UTILISATION DES EXÉCUTABLES MER, PX, CONSX

Cette section donne les lignes de commande à utiliser sous Windows pour lancer chacun des exécutables, et le format des fichiers attendus.

## 4.1 PROLOGUE : MER, PX, CONSX POUR LES NULS

Ce paragraphe définit quelques termes de métier pour les purs développeurs informatiques étrangers au monde des risques nucléaires :

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- Activité : valeur en désintégrations/s, i.e. en Becquerels ; c'est typiquement ce que produit MER, en-dehors du fichier de terme source : chaque instant est associé à un débit exprimé en Becquerels.
- Terme source : intégrale d'une fonction temporelle donnant des Becquerels. Le résultat est un nombre de désintégrations. En général on le calcule sur des phases représentatives de l'accident étudié : 24h, phases A, B, C, D …
- Espèce : c'est un ensemble d'isotopes. Il est défini par la balise <Espece> du fichier XML de configuration de MER ;
- Famille : synonyme de Espèce ;

## 4.2 MER.EXE

### *4.2.1 LIGNE DE COMMANDE DE MER*

```
"MER\MER_EXE.exe" --dataset:"MER\in\XXXX.xml" --configXML:"MER\mer1300.xml"
```

### *4.2.2 FORMAT DE FICHIER D'ENTRÉE DE MER (JDD)*

Les programmes comme KANT ou BASE_MERCOR fournissent un fichier de Jeu De Données (JDD) à MER.exe, dont le format est :

```
<?xml version="1.0" encoding="utf-8"?>
<MER version="1.0">
     <!--Sorties-->
     <Sortie>
          <FilterLogger filename="xxxxx.mer" groupfilter="TermeSource"
workingdirectory="MER\out\"/>
          <FilterLogger filename="xxxxx.mer" groupfilter="xxxxx"
workingdirectory="MER\out\"/>
          ...
     </Sortie>
     <!--Data-->
     <Data>
          <FPAs>
               <FPA name="DR_B" value="2"/>
               <FPA name="BR_B" value="1"/>
          ...
          </FPAs>
          <PIs>
               <PI name="PAR_SPHASE_B" defaultvalue="0.5"/>
               <PI name="PAR_SPHASE_D" defaultvalue="0.625"/>
               ...
          </PIs>
          <VPs>
               <VP name="PEIs" value="5"/>
               ...
          </VPs>
     </Data>
```

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

</MER>

Les calculs demandés à MER par Evariste sont plus complexes et demandent un certain nombre de pas de temps, indiqués dans les balises <VP>.

La balise <FilterLogger> sert à définir les fichiers de sortie de MER, détaillés dans la section qui suit.

#### *4.2.3 FORMAT DES FICHIERS DE SORTIE DE MER*

En l'absence de manuel explicite de MER, on a recoupé le fonctionnement actuel du code et l'expertise de Vincent Tanchoux sur le sujet.

- MER produit un fichier de sortie par balise <FilterLogger> indiquée dans le nœud <Sortie> du JDD.
- La balise <FilterLogger> comporte plusieurs attributs XML :
  - filename : le nom sous lequel sera enregistré le fichier ;
  - groupfilter: la catégorie de données à mettre dans le fichier de sortie ; le format du fichier de sortie dépend de cette valeur.
  - workingdirectory : le répertoire où doit être produit le fichier. C'est un chemin relatif à partir du répertoire courant (celui d'où l'on a invoqué MER) ;

#### 4.2.3.1 Valeur de groupfilter TermeSource

- Texte tabulé, champs séparés par des espaces ou des tabulations ;
- Les champs 1-5 sont donnent la date de production de la ligne et sont ignorés en pratique ;
- Les champs 1-6 sont séparés par des espaces ;
- Les champs suivants sont séparés par des tabulations
- Ligne 1 :
  - Champ : texte «AFFICHAGE DES TERMES SOURCES» ;
- Lignes suivantes :
  - Champ 6 : nom de la source (texte sans espace);
  - Champ 7 : Valeur du terme source : nombre décimal double précision, en notation scientifique (représente un nombre de désintégrations sur la durée considérée)
- Dernière ligne :
  - Champ : texte « AFFICHAGE DES RESULTATS DE MERCOR» ;

#### 4.2.3.2 Valeurs de groupfilter autres que TermeSource

- Texte tabulé, champs séparés par des espaces ou des tabulations ;
- Les champs 1-5 sont donnent la date de production de la ligne et sont ignorés en pratique ;
- Les champs 1-6 sont séparés par des espaces ;
- Les champs suivants sont séparés par des tabulations
- Ligne 1 :
  - Champ 6 : nom de la grandeur représentée (i.e. valeur donnée à groupfilter)
- Ligne2 : Noms des sous-grandeurs représentées :
  - Champ 6 : « Temps »
  - Champ 7-fin : Nom de chaque sous-grandeur représentée ; c'est le titre de colonne pour toutes les valeurs indiquées dans les lignes ultérieures ;
- Lignes 3-fin :

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- Champ 6 : temps en s, sous forme de flottant double précision en notation scientifique ;
- Champs 7-fin : valeur correspondant au titre de colonne, sous forme d'un flottant double précision en notation scientifique ;

#### 4.2.3.3 Valeurs invalides de Groupfilter

Les valeurs «Data_Ru02s» et «Data_RuO4g» produisent des fichiers vides.

Techniquement (discussion avec Vincent, 2017/01/02), ces valeurs correspondent à des modèles invalides, mais qui pourraient être réintégrés à MER dans le futur.

#### 4.2.3.4 Irrégularités dans les formats de sortie de MER

- D'anciennes versions de MER omettent la ligne de titre.
- Les fichiers de rejets d'isotopes (ceux qui sont transmis à pX) mentionnent un pas de temps de plus que les fichiers d'activité. C'est le premier pas de temps.
- Des temps sont dupliqués à un endroit :
  - Sur un fichier hors-isotopes, la ligne du temps répété contient des valeurs numériques invalides pour les valeurs d'activité instantanée ;
  - Sur un fichier de rejet d'isotopes, la ligne du temps répété est strictement identique à la première ligne pour ce pas de temps.

### 4.3 PX.EXE

#### *4.3.1 LIGNE DE COMMANDE POUR PX*

D'après *[MPX] §5 «Exécution du logiciel»*:

```
bin/px --runmode [RS|RV|DB|ALL]   //Verbosité: standard/bavard(lent)/debug/tout
     --def_param_file fileIn1     //Fichier de. paramètres. Défaut .def_param
     --param_file fileIn2   //Fichier de paramètres. Défaut .param   (pour nous: mesparams.txt)
     --message_file fileOut    //Fichier de log. Défaut: stdout
```

#### *4.3.2 LES FICHIERS STATIQUES TRAITÉS PAR PX*

Il s'agit là des fichiers que Evariste ne modifie pas

#### 4.3.2.1 Le fichier de filiation

Fichier de texte tabulé (familles_BEPAG.pxs), mentionné dans [MPX] §3.2.8 ; Structure non-documentée à cet endroit (où d'ailleurs ?) mais aisément reconstituable :

- Ligne 1 : nombre d'éléments décrits
- Lignes suivantes : champs séparés par des blancs (espace ou tabulation) :
  - Index
  - Nom de l'élément
  - Index de chaque élément obtenu dans la chaîne de décomposition.
  - 0
- L'élément d'index 0 est nommé «Scalar» et sa décomposition est limitée à l'élément 0.

#### 4.3.2.2 Le fichier de radio-éléments

Fichier de texte tabulé (Ref_BEPAG.pxs) :

- Structure non-documentée dans [MPX] §3.2.8 et je ne vais pas me fatiguer à le faire.

#### *4.3.3 LES FICHIERS TRAITÉS PAR PX*

#### 4.3.3.1 Utilisation de NetCDF par pX

Le programme pX, tout comme ConsX, utilise des fichiers dont la structure est basée sur le format NetCDF-3.

Un fichier au format NetCDF 3 repose sur les notions suivantes :

- fichier NetCDF = dimensions, variables, attributs

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- dimension: nom, valeur (entier>0 désignant le nombre d'éléments dans cette dimension)
- variable: nom, liste de dimensions (aucune pour un scalaire)
- types possibles: *double*, *float*, *int*, *char* ; pour les chaînes il faut ajouter une dimension.
- attribut : associé à une variable ou à tout le fichier (attribut global)

Le programme pX utilise la spécification du format NetCDF-3 de la façon qui suit :

- attribut global «*Convention*» = «*IRSN V1.0*«
  - toute variable a un attribut «<*variable*>*_long_name*» et un attribut «<*variable*>*_units*»
  - variable «*Timeref*»: origine des temps «*jj/mm/aaaa hh:mm:ss*» (donc un tableau de caractères)
  - dimension «*NCharDate*»: nombre de caractères pour l'origine des temps
  - variable «*NPointsSet*»: nombre des ensembles de points (maillages fixes ou mobiles)
- Pour décrire un maillage ou ensemble de points quelconque d'index <*ep*>:
  - Attribut global: *PointsSet_long_name*<*ep*> : nom de l'ensemble de points
  - Attribut global: *PointsSet_Type*<*ep*> : 0 pour dire que c'est un maillage
  - Attribut global: *PointsSet_IsMobile*<*ep*> : 0 pour fixe, 1 pour mobile
- Pour décrire un maillage fixe d'index <*ep*>:
  - Dimensions: *DimI*<*ep*>, *DimJ*<*ep*>, *DimK*<*ep*> : nombre de points sur chaque coordonnée
  - *CoordX0*<*ep*> : position X, pour tout *DimI*<*ep*>**DimJ*<*ep*>**DimK*<*ep*>
  - *CoordY0*<*ep*>, *CoordZ0*<*ep*>: idem pour positions Y et Z
- Pour décrire un maillage mobile d'index <*ep*>: comme maillage, mais avec
  - dimension *NTimePointsSet*<*ep*> : nombre d'instants à considérer
  - variable *TimePointsSet*<*ep*> : liste des instants, en s, au nombre de *NTimePointsSet*<*ep*>
  - variable *CoordX0*<*ep*> : une valeur pour tout *NTimePointsSet*<*ep*>**DimK*<*ep*>**DimJ*<*ep*>**DimI*<*ep*>
  - variables *CoordY0*<*ep*>, *CoordZ0*<*ep*> : idem
- Pour décrire une liste de points fixes <ep>:
  - dimension *DimNPts*<*ep*> : nombre de points
  - dimension *NMaxChar* : nombre de caractères pour les noms des points
  - variable *NamePts*<*ep*> : pour tout *DimNPts*<*ep*>**NMaxChar*
  - variable *CoordX0*<*ep*> : pour tout *DimNPts*<*ep*>
  - variables *CoordY0*<*ep*>, *CoordZ0*<*ep*> : idem
- Pour décrire une liste de points mobiles <ep>: comme ensemble de points fixe, mais avec
  - dimension *NTimePointsSet*<*ep*> : nombre d'instants
  - variable *TimePointsSet*<*ep*> : pour tout *NTimePointsSet*<*ep*>
  - variable *CoordX0*<*ep*> : pour tout *NTimePointsSet*<*ep*>**DimNPts*<*ep*>
  - variable *CoordY0*<*ep*>, *CoordZ0*<*ep*> : idem
- Pour décrire des données météo: comme ensemble de points, avec noms un poil différents:
  - dimension *NTimeMeteo* : nombre d'instants de données météo
  - dimensions *DimI*<*ep*>, ... *DimK*<*ep*> : comme pour un maillage normal
  - variable *TimeMeteo* : pour tout *NTimeMeteo*
  - variables *VitU*, *VitV*, *VitW* : pour tout *NTimeMeteo***DimK*<*ep*>*...**DimI*<*ep*>

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- variables *Stab*, *Rain*, *HCLA* : pour tout *NTimeMeteo*DimJ<ep>*DimI<ep>*
- variables *CoordX0<ep>,...,CoordZ0<ep>* : pour *DimK<ep>,...,DimI<ep>* comme maillage fixe normal.
- attribut global *Meteo_long_name<ep>* : comme pour un maillage
- attribut global *Meteo_RefPointsSet* : valeur de l'indice *<ep>* (il y a UNE météo par fichier)
- attributs globaux *PointsSet_Type<ep>*, *PointsSet_IsMobile<ep>* : =0,0 car maillage fixe obligatoire.

- Pour décrire les sources [MPX §6.1.2.4]:
  - dépend du temps
  - est défini sur un maillage réduit à un seul point

#### 4.3.3.2 Fichier météo (entrée)

C'est un fichier NetCDF. Son format est celui décrit dans §4.3.3.1 pour des données météo.

#### 4.3.3.3 Fichier des résultats demandés (entrée)

C'est un fichier NetCDF qui suit les spécifications de §4.3.3.1, et de [MPX] §6.1.2.5 ; Ce fichier

- décrit quels résultats *{r}* sont demandés. *{r}* peut valoir :
  - 1 pour l'activité instantanée ;
  - 2 pour l'activité intégrée ;
  - 3 pour le dépôt ;
  - 4 pour demander les résultats 1, 2 et 3 ;
- donne des ensembles de points *{ep}* où ces résultats sont demandés ;
- associe un résultat demandé *{r}* à l'ensemble des points *{ep}* où il est demandé.

L'association entre un résultat *{r}* et un ensemble de points *{ep}* se fait ainsi :

- variable *NResult* : nombre de résultats définis dans le fichier ;
- dimension *NTimeResult{r}* : nombre d'instants auxquels le résultat r est demandé
- variable *ResultTime{r}* : liste des *NTimeResult{r}* instants (en s) où *{r}* est demandé.
- Attribut global *Result_long_name{r}* : le nom du résultat *{r}*;
- Attribut global *Result_RefPointsSet{r}* : l'index *{ep}* de l'ensemble des points concernés par ce résultat;

Pour chaque index *{ep}* mentionné dans les variables *Result_RefPointsSet{r}*, le fichier décrit l'ensemble de points *{ep}* comme en §4.3.3.1, en donnant :

- Les dimensions associées à l'ensemble de points *{ep}* ;
- Les variables associées à l'ensemble de points *{ep}* ;
- Les attributs globaux associés à l'ensemble de points *{ep}* ;

Autres variables et attributs :

- variable *TimeRef* : date de référence du fichier ;
- variable *Convention* : «IRSN V1.0»

#### 4.3.3.4 Fichier des sources (entrée)

Philosophiquement[4], pX ne connaît que des calculs par isotope, i.e. les données qu'il attend en entrée sont les résultats de MER correspondant à :

- Des isotopes (pX ne connaît pas le groupement en espèces)
- … sommés sur tous les chemins de fuite possibles. Cela correspond aux valeurs de la forme `Isotopes+<espece>+ActiviteRejet` dans l'attribut `groupfilter` de la balise `<FilterLogger>` du fichier de JDD de MER ;

Les données d'une source dépendant du temps sont théoriquement fournies en tout point d'un maillage, mais pX v1.0 ne traite pour l'instant que le cas d'un maillage réduit à un point.

[4] Discussion avec Vincent, 2017/01/03

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

Le fichier est au format NetCDF, respectant les conventions annoncées en §4.3.3.1, pour les variables, dimensions et attributs qui suivent :

- variable *NSource* : nombre des sources définies par ce fichier ;
- dimension *NIsoSource{s}* : nombre d'isotopes de la source {s} ;
- dimension *NTimeSource{s}* : Nombre de paliers de rejets.
- variable *TimeBeginSource{s}* : nombre de s entre Timeref et le début de rejet de la source s ;
- variable *TimeSource{s}* : instants (en s) de fin de palier de rejets, au nombre de *NTimeSource{s}* ;
- *DebitSource{s}* : débits de rejet de la source {s}, palier par palier et isotope par isotope, donc au nombre de NTimeSource{s}*NIsoSource{s} ;
- *IsoIdListSource{s}* : identifiants des isotopes rejetés par {s}, au nombre de NisoSources{s}, dans le fichier de solubilité .pxs ;
- *IsoNameListSource{s}* : noms des isotopes rejetés par {s}, soit au total *NisoSources{s}*NMaxChar* ;

L'association entre une source {s} et le maillage concerné {ep} se fait comme suit :

- attribut global *Source_long_name{s}* : le nom de la source ;
- attribut global *Source_RefPointsSet{s}* : la valeur *{ep}* associée à la source ;
- attribut global *PointsSet_long_name{ep}* : le nom du maillage *{ep}* ;

On retrouve les variables et attributs décrivant le maillage {ep}:

- dimensions *DimK{ep}*, *DimJ{ep}*, *DimI{ep}*
- variables *CoordX0{ep}*, *CoordY0{ep}*, *CoordZ0{ep}*
- *PointsSet_Type{ep}* : =0 car seul les maillages sont considérés pour le fichier des sources ;
- *PointsSet_IsMobile{ep}* : =0 car pX ne considère que les maillages fixes pour les sources ;

Les variables communes y sont bien entendu :

- Attribut *Convention* : «IRSN V1.0»

### 4.3.3.5 Les résultats (sortie)

Reprend tout le contenu du fichier des résultats demandés qui avait été fourni en entrée.

Donne en plus les informations suivantes :

- Variable *NSource* : nombre des sources ;
- Pour chaque source *{s}* :
  - dimension *NIsoSource{s}* : comme dans le fichier des sources.
  - variable *IsoIdListSource{s}* : identifie les *NIsoSource{s}* isotopes rejetés ;
  - variable *IsoNameListSource{s}* : *NisoSource{s}* noms d'au plus *NmaxChar* caractères, donnant les isotopes rejetés par *{s}* ;
  - Pour chaque résultat *{r}* associé à un ensemble de points *{ep}*:
    - variable *ResultInst{s}-{r}* : valeurs du résultat *{r}* (=1 ou 4), au nombre de *NTimeResult{r}*<dimension-spatiale-de-ep>*NisoSource{s}* ;
    - variable *ResultInteg{s}-{r}* : idem pour r=2 ou r=4
    - variable *ResultDepotSec{s}-{r}* : idem pour r=3 ou r=4
    - variable *ResultDepotHumide{s}-{r}* : idem pour r=3 ou r=4
    - variable *ResultDepotIntegSec{s}-{r}* : idem pour r=3 ou r=4
    - variable *ResultDepotIntegHumide{s}-{r}* : idem pour r=3 ou r=4

## 4.4 CONSX

#### *4.4.1 LIGNE DE COMMANDE POUR CONSX*

D'après [MCX] §2.2.1 Fonctionnement, le calcul se fait aux instants et points du calcul de dispersion (pX), et d'après [MCX] §2.2.2 la commande est :

`consx <fichier-parametrique>` : conséquences radiologiques seules (pas de données de dispersion)

`-d <fichier-parametrique>` : la sortie contient une copie des données de dispersion de l'entrée.

`--dispersion <fichier-...>`: idem

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

### *4.4.2 LES FICHIERS STATIQUES*

On désigne ainsi les fichiers que consX prend en compte mais qu'Evariste n'a pas besoin de modifier.

#### 4.4.2.1 Solubilités

Texte tabulé décrit dans [MCX] §5.3.1 ; les colonnes sont :

- ID : identifiant de l'isotope dans une certaine base Ecrin ;
- NomIso : nom de l'isotope dans les données du code de dispersion (ce doit être celles de pX) *dosecoef* (TODO : pourquoi peut-il y avoir divergence avec le champ d'après ?)
- NomDoseCoef : nom de l'isotope utilisé par le module python *dosecoef* ;
- SolubiliteDefaut : ? (voir les valeurs dans le fichier des coefficients de dose)
- DispoF : ?
- DispoM : ?
- DispoS : ?
- Granu : ?

#### 4.4.2.2 Coefficients de dose

Fichier de texte tabulé décrit dans [MCX] §5.3.2 ; (*CoefPX_BEPAG.pxs*)

- Première partie : correspondances entre IDs et noms, réparti en sections «Cibles», «VA» (voies atteintes), «Granulometrie», «Solubilite», «Organes», «TI» (Tranche d'Intégration, i.e. la catégorie d'âge visée). solubilités, organes ; composé de plusieurs sections
  - Ligne 1 : //«section»
  - Ligne 2 : Nombre d'IDs de la section
  - Lignes suivantes : <id> : <valeur>
- Seconde partie : tableau :
  - Colonne 1 «IdRad» : ID de l'isotope
  - Colonne 2 «NomRad» : Nom de l'isotope
  - Colonne 3 «Cible» : cible
  - Colonne 4 «VA» : voie d'atteinte
  - Colonne 5 «Granu» : granulométrie
  - Colonne 6 «Solub» : solubilité
  - Colonne 7 et suivantes : coefficients de doses par organe : «Eff» (corps entier), «Moelle», «Os», «Poumon», «Sein», «Tyro?de»

### *4.4.3 FICHIERS D'ENTRÉE*

#### 4.4.3.1 Le fichier de paramètres

La structure recommandée est donnée par [MCX] §3.1.1 «Structure du fichier» :

*[Paramètres génériques] [Tache0] [Tache1] ...*

La syntaxe requise est celle-ci (Les mots-clefs ne comportent pas d'accent) :

- Commentaires: # ...
- Paramètres génériques: <*mot-clef*> = *valeur*
- Paramètres propres à une tâche: <*mot-clef*>.<*num-tache*> = *valeur*
- Paramètres à fournir obligatoirement pour chaque tâche:
- *Nom de la tache / Nature de la tache / Identifiant du resultat*

#### 4.4.3.2 Le fichier d'entrée

L'appellation «fichier d'entrée» est quelque peu abusive puisque ce fichier n'est pas fourni seul à ConsX, mais avec un fichier de paramètres. Cependant on se tiendra ici au vocabulaire actuel.

Le format est mentionné par [MCX §5.1 «Fichier d'entrée»] et respecte la convention «IRSN V1.0» de pX.

Le fichier définit pour tout résultat :

- type (activité instantanée, intégrée, etc.)
- ensemble de points (maillage, liste de points)
- liste d'instants
- variables comportant les résultats (1 par activité et par source)
- nombre de sources et liste des isotopes par source

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

#### 4.4.3.3 Les fichiers statiques

Ces fichiers indiquent :

- Solubilités: cf [MCX §5.3.1]
- Coefficients de dose: cf [MCX §5.3.2]
- Rayonnement: cf [MCX §5.3.3]
- Rayonnement: cf [MCX §5.3.4]

### *4.4.4 FICHIER DE SORTIE*

ConsX produit un fichier de sortie au format NetCDF-3, avec les particularités suivantes :

- chaque tâche produit au moins une variable dans le fichier.

#### 4.4.4.1 Format de fichier pour ConsX

Le format du fichier de sortie est décrit par [MCX] §5.2. En voici une synthèse :

- Les index suivants servent à définir les noms :
  - {s} : identifie la source, comme pour pX ;
  - {r} : correspond à un type de résultat demandé pour ConsX (la valeur entre 1 et 4) ;
  - {c} : index de la «conséquence», i.e. numéro de la tâche exécutée
- Une notation ambigüe de la spec [MCX} est {s+1} , qui désigne le nombre de sources fournies à ConsX plus 1 ; on préférera appeler ici S le nombre des sources, et noter {S+1} l'index de la source fictive qui somme toutes les sources fournies.
- variable *NCsx* : donne le nombre de résultats consx ;
- variable *NSource* : semble ne pas donner le nombre de sources fournies à pX (=1), mais plutôt le nombre de points indiqué par le fichier des résultats demandés à pX (cf 4.3.3.3, dimension *DimNPts0*)
- Pour chaque tâche {c} demandée par le fichier de paramètres :
  - variables du résultat {r} de pX attaché à la conséquence {c} ;
  - attribut global *Csx_long_name{c}* : nom de la conséquence
  - attribut global *Csx_desc{c}* : description de la conséquence
  - attribut global *Csx_som{c}* : type de sommation : Std|Isotopes|Sources|SourcesEtIsotopes
  - attribut global *Csx_refPointsSet{c}* : index *{ep}* de l'ensemble de points attaché à la conséquence ;
  - attribut global *Csx_solu{c}* : chaîne contenant les solubilités des isotopes qui ne seraient pas à leur valeur par défaut ;
  - attribut global *Csx_granu{c}* : chaîne contenant les granulométries des isotopes qui ne seraient pas à leur valeur par défaut ;
  - dimensions du résultat *{r}* de pX attaché à la conséquence *{c}* ; en particulier consX n'ajoute aucune dimension ;
  - pour chaque source *{s}* fournie en entrée, et pour la source fictive *{s}={S+1}* on trouve des variables:
    - variable *Csx<type>{s}-{c}* : valeurs calculées pour *{c}*, instant par instant, point par point, isotope par isotope, donc au nombre de *NTimeResult{r}*<dimension_spatiale_de_ep>*NIsoSource{s}* ;
    - variable *Csx<type>{s}{c}* : valeurs calculées pour *{c}*, instant par instant, point par point, donc au nombre de *NTimeResult{r}*<dimension_spatiale_de_ep>

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

# 5 DÉTAIL DES RÉVISIONS SVN

Cette section liste les numéros de révision SVN associés à chaque thème, avec un bref résumé. Elle s'adresse plutôt à ceux qui rechercheraient la date (et la raison) d'une modification de code particulière.

## 5.1 PORTAGE DE MER POUR LINUX

Il s'agit là des numéros de révision du code, avec le titre du commit original, et un mot d'explication supplémentaire

- r1971 :« MER: chemins cohérents avec l'architecture »
- r1970 : «ecart_grilles_txt.pl: script simple de comparaison calcul VS référence; + nettoyage MER_Ecarts» : ce commit comporte un script simple de vérification des résultats de l'exécutable MER, avec son mode d'emploi.
- r1969 : «src/ et workspace/ : ajout de Makefile et sources.mk pour Linux» : ce commit est celui qui permet la compilation sous Linux. Dans les versions ultérieures du code, «sources.mk» devra être complété en fonction des fichiers apparus ou disparus.
- r1968 : «*.cpp: Portage pour Linux (hors Makefile), avec compilateur g++ 4.4.7.» : ce commit modifie légèrement un certain nombre de fichiers C++ des sources pour que MER compile sous Linux. Les fichiers apparus ultérieurement peuvent demander des modifications semblables si l'on veut compiler une version plis récente de MER sous Linux.
- r2046 : «Mise à jour pour compilation Linux, et nettoyage .sdf, .suo,» : rafraîchissement du portage pour la version r2045.

## 5.2 CODAGE DE L'IHM ET DU MOTEUR DE CALCUL

Le travail fait fin 2016 a été complètement révisé : les révisions intéressantes à ce stade sont les suivantes :

- r278 «IHM: Révision 1-20 septembre : élasticité + multiples bugfix» : première moûture évoluée de l'interface graphique, incluant un redimensionnement de fenêtres, et des fonctions presque complètes.
- r279 «CALCUL_EPS2: 1-20 septembre : MPI, bugfix, histogrammes» : implémentation complète du moteur, y compris pour les possibilités de lancement par MPI.
- r284 «IHM + CALCUL_EPS2.exe: compil Linux + bugfix doublons, NA CSV, choix ISO» : code compilable dans un environnement Linux ; ajout du choix de produire (ou non) des fichiers de rejets par isotopes ;
- r286,287,288 : correction des bugs remontés.

# 6 INSTALLATION/UTILISATION DE MPI

Le moteur de calcul est conçu pour pouvoir être lancé en ligne de commande avec MPI (Message Passing Interface).

Quelques contraintes et informations utiles à l'utilisation de l'implémentation MPI d'Intel sous Windows 7 suivent. Elles sont très largement reprises de la documentation des travaux sur le logiciel KANT.

## 6.1 INTRODUCTION À MPI

MPI (Message Passing Interface) est, comme son nom l'indique, une API qui permet de synchroniser plusieurs processus en envoyant, attendant ou recevant des messages.

### *6.1.1 CONTENU D'UNE IMPLÉMENTATION MPI*

Parmi les nombreuses fonctions que la norme spécifie, voici des exemples des plus courantes:

- initialisation : `MPI_Init() ;`
- pour attendre des messages : `MPI_Recv() ;`

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- pour envoyer des messages : `MPI_Send() ;`
- pour rassembler des données : `MPI_Gather()`, `MPI_Reduce()`, ... ;
- pour connaître le rang du processus courant : `MPI_Comm_rank()`.

#### *6.1.2 CONTRAINTES MATÉRIELLES*

- Les machines concernées doivent pouvoir communiquer entre elles. Cela se fait en général par un réseau Ethernet ou un réseau Infiniband[5].
- L'envoi et la réception de messages entre les processus demandent un peu de mémoire supplémentaire.

#### *6.1.3 CONTRAINTES DE CONFIGURATION*

- Les ports réseau des machines utilisant MPI doivent être ouvert pour qu'elles puissent écouter ou recevoir les messages.
- Les fichiers exécutables doivent être accessibles avec le même chemin sur toutes les machines concernées. Cela passe le plus souvent par l'utilisation d'un lecteur réseau, mais on peut se contenter d'une simple copie pour les cas les plus simples.
- C'est en général un service MPI qui écoute les messages envoyés ou reçus. Il faut pouvoir l'installer en général avec un accès aux périphériques (mode administrateur).

### 6.2 MISE EN RÉSEAU DE PCS SOUS WINDOWS 7

Ce paragraphe décrit les étapes nécessaires pour configurer les 3 PCs de notre environnement de développement, de façon à ce qu'ils disposent d'adresses IP fixes, d'un espace de travail commun WORKGROUP et de la possibilité de partager des lecteurs sur le réseau.

#### *6.2.1 CONFIGURER LES ADRESSES IP*

Paramétrer les connexions réseau de chaque PC pour lui attribuer une adresse IP fixe.

Par exemple, `192.168.0.x` convient à un petit réseau, avec :

- `x` compris entre 0 et 255 ;
- un masque de réseau `255.255.255.0`.

A ce stade, "ping" devrait marcher entre les diverses IP attribuées :

- ouvrir l'invite de commande sur chaque PC concerné ;
- entrer la commande `ping<ip>` pour chacune des adresses du réseau.

A ce stade, on peut aussi vérifier que taper \\<ip> dans la barre d'adresse de l'explorateur permet de voir les ressources partagées de la machine concernée.

#### *6.2.2 CONFIGURER LES CONNEXIONS RÉSEAU POUR AVOIR UN GROUPE DE TRAVAIL*

Il convient de commencer par configurer les PCs pour que chacun associe le même nom à chaque adresse IP d'un PC du groupe de travail. Le plus simple pour cela consiste à utiliser un fichier *LMHOSTS* identique sur chaque PC :

- Aller dans `%SYSTEMROOT%\SYSTEM32\DRIVERS\ETC`

[5] C'est donc possible sur un réseau d'entreprise, même si le partage du débit entre tous les postes de travail influe sur le temps passé en envoi de messages de synchronisation.

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- Copier `LMHOSTS.SAM`[6] sous le nom `LMHOSTS`
- Editer `LMHOSTS` en s'aidant des commentaires qu'il contient.

Exemple de fichier *LMHOSTS* :

192.168.0.10 a12w1010 #PRE
192.168.0.8 a12w1008 #PRE
192.168.0.5 a12w1005 #PRE

Les groupes de travail de Windows XP sont gérés par le service "Computer Browser Service".

Ce service utilise en particulier le protocole NetBios de Microsoft sur les connexions TCP/IP.

Il faut donc vérifier que NetBios est activé sur ces connexions, et que le PC utilise bien le fichier *LMHOSTS* :

- connexion réseau"-->"Propriétés"-->"TCP/IP" :
  - utiliser une adresse IP fixe ;
  - "advanced", "Activerrecherche LMHOSTS" ;
  - "advanced", "Activer NetBIOS" sur TCP/IP ;
- "gestion de l'ordinateur"-->"Services"-->"Explorateur d'ordinateur": redémarrer ;
- "favoris réseau"-->"Afficher tous les ordinateurs du groupe de travail" doit marcher à présent.

Par défaut, les différents PCs dotés de « Computer Browser Service » élisent un maître. Pour limiter le trafic réseau on peut indiquer un PC en particulier pour tenir ce rôle :

- lancer l'éditeur de registre (« regedit ») ;
- aller à `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\Parameters`
- `\MaintainServerList = Yes` (pour les PCs que l'on veut maîtres), No sinon.

La liste des ordinateurs du groupe de travail n'est mise à jour qu'à des intervalles de 10 à 15 minutes ;

Cependant, taper `\\<machine>` doit déjà faire accéder aux ressources partagées de la machine concernée.

On reconnaît le groupe de travail dans les propriétés de l'ordinateur (raccourci Windows-Pause) : il se trouve alors dans la rubrique « Paramètres de nom d'ordinateur, de domaine et de groupe de travail».

### *6.2.3 CRÉER ET PARTAGER DES LECTEURS RÉSEAU*

Partager un répertoire :

- clic droit, Propriétés, onglet Partage ;
- à ce stade, le système demande alors un nom d'utilisateur légitime sur la machine qui possède le dossier partagé.

Attribuer une lettre de lecteur réseau :

- Poste de travail, "Connecter un lecteur réseau".

[6] Le .SAM signifie « sample » : il contient essentiellement des commentaires expliquant la structure d'un fichier LMHOSTS.

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

## 6.3 INSTALLATION DE INTEL MPI SOUS WINDOWS 7

Cette procédure est identique à ce qui se faisait sous Windows XP.

### *6.3.1 DOCUMENTATION*

Après l'installation de «Intel MPI», la documentation est accessible à partir du menu Démarrer.

En particulier, les explications qui suivent sont largement inspirées du document "Getting Started" fourni par l'installation.

### *6.3.2 INSTALLATION SOUS WINDOWS 7*

L'installateur fourni par Intel automatise la plus grande partie de l'installation.

Le résultat, accessible depuis le menu Démarrer, ressemble à ceci :

```
Démarrer
+- Programmes
  +- Intel Development Products
     +- Intel MPI
+- Build environment
                |  |- Build Environment
                |  |- wmpiconfig
                |  |- wmpiexec          : pour lancer un programme MPI
                |  +- wmpiregister      : pour configurer les droits d'exécution
                |- Documentation index
                |- Getting started
                |- Intel Software Manager
                |- Intel MPI Library Reference
                +- Release Notes
```

Les fichiers installés se trouvent dans

```
C:\Program Files\Intel\MPI\4.1.0.023
|-doc\         : les fichiers de doc
|-ia32\
| | bin\         : exécutables des services MPI
| | |-mpicc.exe   : wrapper pour compiler un programme pour MPI
| | +-mpiexec.exe : commande l'exécution d'un programme MPI
| |   ...
| |-etc\
| |-include\     : fichiers .h (pour C/C++) et .mod (pour ?)
| +-lib\         : fichiers .lib
|   |-impi.lib    : pour les compilations en C
|   +-impicxx.lib : pour les compilations en C++
|-setup\         : les installateurs (pas besoin d'y retoucher a priori)
+-test\          : sources des programmes de test.
  |-test.c
  +-test.cpp
    ...
```

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

Les services installés par Intel MPI se voient dans les outils d'administration de l'ordinateur :

- "Intel MPI Library Process Manager" : c'est le service système utilisé par Intel MPI pour faire communiquer les processus lorsqu'ils sont distribués sur plusieurs postes. Il correspond à l'exécutable *smpd.exe*.

L'intégration dans VS2010 est proposée et se traduit par quelques menus supplémentaires.

- la fenêtre Debug-->Windows-->Processes gagne :
  - une colonne "Transport" (la façon de transporter les messages MPI) ;
  - une colonne "Transport Qualifier" (le nœud concerné).

Ces indications supplémentaires permettent de savoir par où et sous quel nom les processus débogués enverront leurs messages aux autres processus MPI.

### *6.3.3 TEST DE COMPILATION*

Compilation dans Visual C++ 2010 :

- créer un projet vide ;
- y ajouter *test.c* [7] ;
- properties-->"Configuration Properties"-->C/C++ :
  - « additional include directory » : ajouter le chemin du répertoire *include* décrit au §6.3.2 ;
  - properties-->"Configuration Properties"-->Linker :
    - Additional dependencies[8] : *impi.lib* ;
    - General-->Additional Library Directories : le chemin du répertoire *lib*.

### *6.3.4 ENREGISTREMENT D'UN MOT DE PASSE AVEC WMPIREGISTER*

Le lanceur `mpiexec` doit savoir sous quelle identité les processus doivent être exécutés :

- lancer l'utilitaire `wmpiregister` ;
- lui donner un compte Windows valide pour le poste où `wmpiregister` a été lancé ;
- lui donner le mot de passe correspondant au compte Windows ;
- faire ainsi sur chaque machine susceptible de participer à l'exécution d'une tâche MPI.

Remarques :

- L'ouverture d'une session locale doit être autorisée, pour le compte concernée, sur la machine concernée.
- si l'utilisateur change de mot de passe, il faudra de nouveau exécuter `wmpiregister` sur chacune des machines concernées.

### *6.3.5 EXÉCUTION PAR WMPIEXEC*

Exécuter `wmpiexec` (ce programme affiche moins d'erreurs que `wmpiconfig`) :

[7] Voir §6.3.2 pour le chemin ;

[8] Si on avait utilisé *test.cpp*, il faudrait aussi lier avec *impicxx.lib*, car il a recours à l'API C++ de MPI qui est désuette.

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- Application, chemin de l'exécutable : doit être le même sur toutes les machines concernées ;
- "Advanced options" : Hosts = `a12w1010 a12w1008 a12w1005 ;`
- "Show Command" ;
- Execute.

Si le programme est bloqué en cours d'exécution, le bouton « Break » l'arrêtera. Toutefois, des processus issus d'une exécution plantée peuvent continuer à tourner : il faut les arrêter manuellement sur les machines concernées[9].

### *6.3.6 INSTALLATION EN CLUSTER*

Le manuel MPI Reference (référence **Erreur ! Source du renvoi introuvable.**) propose, au §3 *User authorization,* une façon de simplifier l'installation lorsque l'on peut mettre des autorisations via Active Directory (sur un réseau d'entreprise dont les administrateurs auront en général installé un tel service).

Les PCs de développement n'étant pas connectés à un service Active Directory, on s'est contenté de les configurer pour disposer d'un groupe de travail WORKGROUP (voir §6.2).

### *6.3.7 CONFIGURATION AVEC WMPICONFIG*

Cet outil est censé faciliter la configuration de Intel MPI sur plusieurs postes.

Il se trouve dans les chemins mentionnés par §6.3.2.

On peut néanmoins s'en passer et n'utiliser que `wmpiregister`.

Ajouter les hôtes :

- le programme commence par afficher une erreur "`No servers available for thisdomain`" ... car le domaine par défaut est PROTON par défaut et le PC n'y est plus connecté ;
- ensuite, nous pouvons sélectionner le "vrai" domaine WorkGroup paramétré pour nos PCs de développement ;
- sélectionner chaque hôte ajouté ;
- s'y connecter à l'aide de `wmpiregister` (il est automatiquement appelé, mais parfois un peu long à venir).

## 6.4 DÉBOGAGE DE PROGRAMMES MPI AVEC VC2010

En environnement MPI, déboguer à l'aide de `printf()` est compliqué par le fait que les sorties des différents processus arrivent avec des décalages difficiles à contrôler.

Il est plus efficace de s'aider d'un débogueur comme celui de Visual C++ 2010, utilisé comme suit :

- créer une boucle infinie `while(true)` ; dans `main()` ;
- recompiler le projet à déboguer (ici *CALCUL_EPS2*.exe)[10] ;
- lancer l'exécutable à débogueur (c'est plus facile avec un script *.bat*) ;
- dans le menu, utiliser le Debug-->Attach to process (ou le raccourci C-A-p) pour choisir un processus *CALCUL_EPS2.exe* où s'attacher ;
- utiliser Debug->"Other Windows"-->Processes pour avoir la liste des processus attachés au débogueur ;
- faire Debug-->"Break all" pour interrompre les processus en cours ;

[9] Sur Windows 7, le moniteur de performances se montre incapable d'afficher tous les processus. Passer par des utilitaires efficaces comme «Process Explorer» de la suite Microsoft SysInternals.

[10] Une option –wait a été ajoutée pour ne pas avoir à construire spécialement un nouvel exécutable pour chaque séance de débogage.

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- le processus est alors arrêté dans la boucle infinie ;
- utiliser la commande `C-S-F10` pour faire continuer le processus après la boucle infinie ;
- détacher le processus ;
- faire de même avec les autres processus MPI.

# 7 ANNEXES

Contient notamment les indications et astuces sur les outils utilisés.

## 7.1 TORTOISE SVN

Quand un dépôt SVN est trop volumineux, et que seule une petite partie de l'arborescence nous intéresse :

- Clic droit → «SVN Checkout»
  - Si un unique sous-répertoire du dépôt est à copier en entier:
    - explorer le champ «URL of repository» pour atteindre ce sous-répertoire.
  - Sinon choisir depth=«immediate children only»
  - Clic sur «Ok» pour fermer la fenêtre «SVN Checkout» et copier les repertoires voulus (ou juste leur racine, en function du choix ci-dessus);
- Ensuite, sur les répertoires qui nous intéressent:
  - Clic droit → Tortoise SVN → «Update to revision» (plus précis que le «Update» par défaut)
  - Par défaut le champ «Update depth» vaut "Working copy", i.e. SVN met à jour les fichiers déjà présents sur le disque, mais ne rapatrie pas les autres fichiers depuis le serveur.
  - Clic sur le bouton «Choose items» : un arbre développable apparaît, où l'on peut cocher les répertoires et fichiers voulus ;
  - après validation de ce choix, le champ «Update depth» passe indique désormais «custom depths» ;
  - après validation, les fichiers et répertoires sélectionnés sont créés (mais sans sous-répertoires)
- Enfin, sur les répertoires créés précédemment et dont on veut tout le contenu :
  - Clic droit → Tortoise SVN → «Update to revision» (plus précis que le «Update» par défaut)
  - Clic sur le bouton «Update depth» → "Fully recursive" : SVN rapatriera tous les répertoires et sous-répertoires.

## 7.2 GIT SVN

L'outil Git a été fort productif pour deux usages :

- synchroniser le développement entre les machines ayant accès au serveur SVN réseau et les autres.
- Créer des branches de test et les «jeter» (ou les fusionner) quand on a fini de s'en servir ;

Les paragraphes qui suivent illustrent comment Git peut convenir à ces deux usages.

### *7.2.1 GIT SVN*

Pour se synchroniser avec le dépôt SVN sur les PCs connectés, Git propose la commande «git svn», présentée en détail sur.

Les sous-commandes les plus utiles à ce stade sont :

- `git svn log --show-commit [-r <n>]` : affiche l'historique tel que le voit SVN, mais avec le numéro de commit Git correspondant ;

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- `git svn clone <source> mon_depot` : crée un sous-répertoire *trunk/* contenant tout l'historique de SVN, sous forme de commits que l'on peut visiter avec Git par la suite. C'est une étape longue (1h pour le dépôt OSCAR dans le cadre d'un autre projet).

#### *7.2.2 GIT BUNDLE*

Cette commande de Git permet de reporter les modifications d'un dépôt git hors-ligne vers un autre via un simple fichier qui se limite aux différences par rapport à une ou plusieurs versions de base ;

Les détails sont sur *https://git-scm.com/docs/git-bundle*, et des exemples se trouvent dans *http://www-cs-students.stanford.edu/~blynn/gitmagic/ch06.html* ;

Pour les utilisateurs habitués à la ligne de commande, les opérations se présentent ainsi :

```
#Bizarrement, on ne peut pas utiliser les hash des commits
<départ>$ git bundle create ref-v1-v2.1.bundle master~2..master
#Pour vérifier si on dispose des versions de référence
<arrivée>$ git bundle verify ref-v1-v2.1.bundle
 #Importe ce que l'on n'a pas déjà sur la branche master
<arrivée>$ git fetch ref-v1-v2.1.bundle master
<arrivée>$ git merge FETCH_HEAD
```

En dépit du jargon parfois intimidant de `git`, cette opération est plus simple qu'un report de multiples patchs ;

### 7.3 VISUAL C++ 2010 : NETTOYAGE DE PROJETS

Reprendre un volumineux dépôt SVN est l'occasion de gagner de la place en repérant:

- les fichiers obsolètes : ceux qui ne sont plus utilisés dans les compilateurs actuels.
- les fichiers inutiles : fichiers temporaires pour la compilation ou pour l'éditeur de code.

On peut notamment les reconnaître à leurs extensions de nom de fichier. Pour Visual Studio, ces extensions sont très nombreuses, et documentées par exemple dans :

- *https://msdn.microsoft.com/en-us/library/xhkhh4zs.aspx* (VS 2015)
- *https://msdn.microsoft.com/en-us/library/3awe4781.aspx*

#### *7.3.1 FICHIERS DE VISUAL STUDIO 2010*

- *.sln* : fichier solution, XML utilisé pour organiser plusieurs projets. Doit être gardé.
- *.suo* : (Solution User Options) sert pour les options propres à l'utilisateur ou au débogage (comme la liste des points d'arrêt par exemple). Il ne semble pas indispensable de l'archiver.
- *.vcxproj* : fichier projet VC++, en XML : le fichier donne en particulier la liste des sources du projet, et doit être gardé.
- *.vcxproj.filters* : (Visual Studio Project Filter Files) : décrit comment grouper les fichiers dans l'explorateur de solution de l'IDE Visual Studio. Pas vital mais appréciable pour l'organisation.
- *.vcxproj.user* : Options propres à l'utilisateur ; a priori ce fichier peut disparaître.
- *.sdf* : Données de navigation (SQL Server Compact Edition Database). Ce fichier binaire, très volumineux est automatiquement régénéré en fonction des besoins. Il faut absolument éviter de l'archiver.

#### *7.3.2 PRODUITS DE COMPILATION*

A l'exception (éventuelle) des fichiers *.pdb*, les extensions ci-dessous peuvent disparaître des archives SVN où elles n'auraient d'ailleurs jamais dû entrer :

- *.pdb* : Program Debug Data (symboles de debug); ce fichier ne sert que dans certains cas de débogage.
- *.pch* : Precompiled Header File (supprimable)

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

- *.tlog* : file access timestamp log (régénéré lors de compilation)
- *.plg* : log de compilation (structuré): non-indispensable
- *.sbr* : Source Browser intermediate file (supprimable)
- *.log* : log de compilation (supprimable)
- *.lastbuildstate*
- *.ipch* : Precompiled Header Files (supprimable et très gros)
- *ipch/* : répertoire contenant des headers pré-compilés, supprimable.

### *7.3.3 FICHIERS POUR VISUAL STUDIO 6.0*

A partir du moment où les projets de VS 6.0 ont été convertis pour VS 2010, il n'y a pas de raison de conserver les fichiers propres à VS 6.0 :

- *.ncb* : données Intellisense (inutiles car régénérées)
- *.aps* : compilation des fichiers .rc et ressource.h
- *.dsp* : projet Visual C++ 6.0
- *.dsw* : workspace Visual C++ 6.0
- *.idb* : Minimum Rebuild Dependency File
- *.unsuccessfulbuild*
- *.clw* : Fichier ClassWizard

### *7.3.4 PROBLÈME DE COHÉRENCE DE GUIDS*

Le très ennuyeux message «*Le fichier projet " a été renommé ou ne fait plus partie de la solution*» apparaît parfois quand on veut compiler une solution VS2010.

- Il semble que cela soit un effet secondaire de la conversion de workspaces VC6 (*.dsw*) avec projets (*.dsp*) vers solution VC2010 (*.sln*) avec projets (*.vcxproj*)
- Le contournement exposé ici s'inspire de http://social.msdn.microsoft.com/Forums/en-US/vcgeneral/thread/5777b517-e935-4905-bf76-8f38d6d8913a et http://blog.coherent-labs.com/2013/01/dump-references-vs-add-in.html
- Voir les dépendances du projet qui ne compile pas dans *Solution→"Dépendances du Projet"*
- Il se passe que le convertisseur de *.dsw* vers *.sln* a attribué des GUIDs aux projets dans le fichier *.sln* concerné, mais sans les reporter dans les fichiers *.vcxproj* correspondants de chaque projet.
- Or si un fichier *.vcxproj* n'indique pas de GUID, VS2010 considère qu'il en a un par défaut, calculé en fonction de son chemin. Alors si la solution ou ses projets sont déplacés, les GUIDs par défaut des fichiers *.vcxproj* changent, mais pas ceux indiqués dans le fichier *.sln*. La cohérence est alors perdue, et VS2010 y perd son latin.
- Pour l'exemple d'une solution référençant un projet baptisé *Utilitaire*, voici les lignes que l'on doit trouver dans chaque fichier pour avoir la cohérence :
    - (i) Dans le *.sln*:

```
Project("{guid-1}") = "Utilitaire", "..\Utilitaire\Utilitaire.vcxproj",
"{guid-2}"
```

    - (ii) Dans le *Utilitaire.vcxproj*

```
<PropertyGroup Label="Globals">
  <ProjectGuid>{guid-2}</ProjectGuid>
</PropertyGroup>
```

    - (iii) Dans les autres fichiers *.vcxproj* de la solution:

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

```
<ProjectReference Include="..\Utilitaire\Utilitaire.vcxproj">
  <Project>{guid-2}</Project>
</ProjectReference>
```

- Pour retrouver la cohérence, on peut vérifier les points suivants :
  - Si (ii) indique un GUID, le reporter dans le champ *{guid-2}* de (i). La réciproque ne semble pas indispensable ;
  - Le chemin indiqué dans (iii) doit être conforme à celui indiqué dans (i) ;
  - Reporter le *{guid-2}* indiqué dans (iii) vers celui indiqué dans (i)
  - Remarque : c'est bien le *{guid-2}* qu'il faut reporter, car le *{guid-1}* est répété dans le fichier *.sln* et ne différencie donc pas les fichiers ;

### *7.3.5 DÉBOGAGE AVEC VISUAL STUDIO*

- Utilisation d'un profileur de code : µSoft en propose, mais dans les versions Premium et Ultimate de Visual Studio. Voir https://msdn.microsoft.com/fr-fr/library/ms182372%28v=vs.100%29.aspx
- Lenteur de l'application compilée pour la configuration Debug : voir si la variable d'environnement `_NO_DEBUG_HEAP=1` améliore la situation. Le problème disparaît avec Visual Studio 2014. Voir https://blogs.msdn.microsoft.com/visualstudioalm/2015/03/03/make-debugging-faster-with-visual-studio/ et https://ofekshilon.com/2014/09/20/accelerating-debug-runs-part-1-_no_debug_heap-2/

## 7.4 INSTALLATION DE NETCDF POUR WINDOWS

### *7.4.1 COMPILATION DE NETCDF AVEC VISUAL STUDIO*

Les opérations décrites ici viennent des instructions fournies par Unidata pour compiler NetCDF avec CMake sur http://www.unidata.ucar.edu/software/netcdf/docs/getting_and_building_netcdf.html#netCDF-CMake

Il semble que cette méthode fonctionne à partir de NetCDF 4.3.0.

- Préalables :
  - installer CMake (https://cmake.org/); J'ai utilisé *cmake-3.7.0-rc2-win64-x64*
  - télécharger NetCDF. J'ai utilisé *netcdf-c-4.4.1* et extraire les fichiers de l'archive.
- Utilisation de CMake pour produire des fichiers de projet pour Visual Studio 2010 :
  - lancer CMake depuis *%CMAKE%\cmake-3.7.0-rc2-win64-x64\bin\cmake.exe*") ;
  - Remplir les champs:
    - *"Where is the source code"*: le répertoire des sources de NetCDF
    - *"Where to build the binaries"*: un répertoire temporaire
  - Clic sur «*Configure*»;
    - Choix du compilateur: ici VS2010 Win64
    - une longue liste de variables apparaît : elles correspondent aux diverses options de compilation de NetCDF; CMake coche automatiquement celles qu'il pense pouvoir produire, et il met en rouge celles qui lui posent un problème.
    - S'il y a eu des erreurs de configuration, retirer les options qui posent un problème et refaire «*Configure*».
      - Nous tenons en particulier aux options :
        - `BUILD_DLL`

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

        - `BUILD_UTILITIES`
        - `CMAKE_INSTALL_PREFIX`
        - `USE_XGETOPT` (remplacement du «getopt», de Unix pour Windows)
        - `ENABLE_XGETOPT`[11]
      - Les options suivantes sont incompatibles avec une installation «de base» du poste de développeur Windows de l'IRSN:
        - `ENABLE_DAP` : cette partie qui accède au réseau ne nous est pas nécessaire ;
        - `USE_HDF5`
        - `USE_NETCDF4` : cela devrait être possible si l'on installe HDF5
    - En cas de doute sur l'état de la configuration, *File-->"Delete Cache"* la reprend au début.
    - Certaines options n'apparaissent qu'après quelques essais (leur apparition peut dépendre du succès d'autres options)
  - Clic sur «*Generate*»;
    - Cela produit les fichiers nécessaires pour compiler avec le «générateur» sélectionné (dans notre cas les fichiers *.vcxproj* et *.sln* pour «Visual Studio 10 2010 Win64»)
    - Cela produit aussi un fichier *config.h* dans le répertoire indiqué «Where to build the binaries» : il faudra le reprendre si on veut compiler l'interface C++ de NetCDF ;
  - Clic sur «*Open Project*»: cela ouvre la solution produite pour VS2010 :
- Compilation de NetCDF dans VS2010 :
  - régénérer le projet «*netcdf*» (cela produit aussi *ncgen*, *ncdump*)
    - repérer les messages «`erreur de syntaxe : absence de ';' avant 'type'`» ou «`utilisation non conforme de ce type comme expression`»[12];
    - les corriger en déplaçant la déclaration concernée au début de son bloc.
    - puis relancer la génération de projet.
  - générer le projet *INSTALL*. Ici des options passées à CMake révèlent leur importance :
    - `USE_XGETOPT`: c'est à l'édition de liens que *ncgen* et *ncdump* requièrent la fonction `getopt()`;
    - `CMAKE_INSTALL_PREFIX` : le droit d'écriture est évidemment nécessaire dans le répertoire mentionné ;
  - Il semble aussi pertinent de copier le fichier *config.h* produit plus haut et le fichier *include/ncconfigure.h* des sources vers le répertoire d'installation : ces fichiers sont nécessaires à la compilation de l'interface C++ de NetCDF.

[11] XGetOpt vient en fait du projet http://www.codeproject.com/Articles/1940/XGetopt-A-Unix-compatible-getopt-for-MFC-and-Win32

[12] Dans *dim.c* et *dfile.c* ; D'après http://stackoverflow.com/questions/9610747 la situation s'est améliorée depuis la version 2013 de Visual Studio.

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*

### *7.4.2 COMPILATION DE NETCDF-CXX AVEC VISUAL STUDIO*

La version compilée ici est le ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-cxx-4.2.tar.gz ;

Son code source est beaucoup plus simple que celui de NetCDF.

Cependant il faut ajouter quelques fichiers produits lors de l'étape de configuration. Comme le code source ne fournit pas d'étape de configuration similaire à celle de NetCDF-4.4.1, on récupère simplement le fichier *config.h* produit à la compilation de NetCDF-4.4.1, et le fichier *ncconfigure.h* présent dans les sources ;

- Extraire les fichiers de l'archive
- Créer un projet netcdf-cxx-4.2/
  - *include/* : *ncvalues.h*, *netcdfcpp.h*
  - *netcdf-cxx-4.2/* : Fichiers propres à Visual Studio (.vcxproj, …)
  - *nctst.cpp*, *ncvalues.cpp*, *netcdf.cpp*
- «Propriétés de configuration»
  - Général :
    - «Type de configuration» : «Bibliothèque statique (.lib)»
    - «Répertoire de sortie» : «$(SolutionDir)\$(Platform)\$(Configuration)\»
  - C/C++→Général :
    - «Autres répertoires Include» : «$NetcdfHome)\include ; ..\include
- «Compilation» :
  - Il est possible qu'un problème se pose pour `fstat` redéfini à la valeur `_fstat64()` : mettre en commentaire la définition de `fstat` dans `config.h` (surprenant, mais mentionné dans des compilations précédentes)

IRSN Ce document est la propriété de l'IRSN et ne peut pas être communiqué, reproduit ou utilisé sans son autorisation écrite préalable
*This document is the property of IRSN and shall not be disseminated, copied or used without its prior formal approval*