

# **Export XML dans Newgen**

**Présentation et propositions**

**Laurent DAVERIO - 30 janvier 2009**

# Introduction

## \* **Objectifs du projet**

- \* Doter Newgen d'un **export XML**, ce qui permettra à terme l'utilisation et l'interfaçage d'outils nouveaux.
- \* Un **import XML** pourra être envisagé dans un deuxième temps.

## \* **Buts de ce document**

- \* **Fournir un premier aperçu** du travail réalisé, comme base pour la future **spécification du XML généré** (DTD, schéma), en fonction :
  - \* des **buts recherchés**, à expliciter (RK, FI, CA, FC, ...)
  - \* des **spécificités** de NewGen (PJ)
  - \* de l'**état de l'art XML** (GAS)
- \* **Déterminer des pistes** pour la suite du projet.

# Exemple de XML généré

- \* **2 exemples** fournis sous forme papier, basés sur **newres.negwen** :
  - \* **“Reservation”** :
    - \* *AND, OR,*
    - \* *Arrays, Sets*
    - \* *External*
    - \* *Tabulated*
  - \* **“Identification”** :
    - \* *ARROW*
- \* Pour chaque exemple, on a fourni :
  - \* l'**extrait de la définition** .newgen pertinente
  - \* la sortie correspondante en format **texte compact**

# Structure générale du XML généré

- \* Structure générale inspirée de HTML (blocs “**head**” + “**body**”):

```
<?xml version="1.0"?>
```

```
<newgen xmlns="http://www.cri.mines-paristech.fr/pips/schemas/  
newgen-1.0">
```

```
<head>
```

```
<!-- Paramètres (“métadonnées”) de l’export -->
```

```
</head>
```

```
<body>
```

```
<!-- Représentation des données Newgen -->
```

```
</body>
```

```
</newgen>
```

- \* L’URL de spécification des schémas mentionne la version du format
- \* Ex. : <http://www.cri.mines-paristech.fr/pips/schemas/newgen-1.0>

# Points de départ de l'export XML

- \* **Trois points de départ**

- 1) `gen_write` → **`gen_xwrite`**

- \* *cas général de traversée*

- 2) `gen_write_without_sharing` → **`gen_xwrite_without_sharing`**

- \* *traversée sans partage d'objets*

- 3) `gen_write_tabulated` → **`gen_xwrite_tabulated`**

- \* *traversée d'un domaine tabulé (avec partage d'objets)*

- \* Chacune de ces fonctions donne lieu à une génération de **données XML complètes** (i.e. : `<newgen>...</newgen>`)

- \* **NB : Indentation** du XML produit :

```
$ ./reserv < newres.in | xmllint --format -
```

# Paramètres de l'export

- \* **gen\_xwrite :**

```
<head>  
  <meta sharing="on" shared_objects="4" />  
</head>
```

- \* **gen\_xwrite\_without\_sharing :**

```
<head>  
  <meta sharing="off" />  
</head>
```

- \* **gen\_xwrite\_tabulated :**

```
<head>  
  <meta sharing="on" shared_objects="4" />  
  <meta tabulated="on" wdomain="5" />  
</head>
```

# Fonctions mises en jeu

- \* Le code XML est construit de proche en proche au cours de la **traversée des données** par des fonctions déclarées dans une structure “**driver**” (cf. *genClib.c*) :

```
dr.null          = xml_p ? xwrite_null : write_null;
dr.leaf_out     = gen_null;
dr.leaf_in      = xml_p ? xwrite_leaf_in : write_leaf_in;
dr.simple_in    = xml_p ? xwrite_simple_in : write_simple_in ;
dr.array_leaf   = xml_p ? xwrite_array_leaf : write_array_leaf;
dr.simple_out   = xml_p ? xwrite_simple_out : write_simple_out;
dr.obj_in       = xml_p ? xwrite_obj_in : write_obj_in;
dr.obj_out      = xml_p ? xwrite_obj_out : write_obj_out;
```

- \* La traversée des données n'est pas modifiée par l'export XML
- \* Les pages qui suivent **recensent toutes les fonctions mises en jeu** dans la génération de l'export XML.

# xwrite\_obj\_in / out

- \* Usage : **domaine complexe** (notamment **CONSTRUCTED**)
- \* Représentation de l'objet :  
<**obj dom**="identification">  
  <!-- contenu de l'objet -->  
</obj>
- \* Type de construction (à l'intérieur du bloc <**obj**>):
  - \* <**and**>...</and>  
  → éléments produits par *xwrite\_simple\_in*
  - \* <**or**>...</or>  
  → éléments produits par *xwrite\_simple\_in*
  - \* <**arrow**>...</arrow>  
  → éléments produits par *gen\_trav\_obj\_constructed*



# xwrite\_simple\_in / out

- \* Usage : membres d'un domaine construit **AND** ou **OR**
- \* Le **nom** du membre figure dans une balise `<member>` :  
`<member name="nbpasseur">`  
`<int>3</int>`  
`</member>`
- \* Le membre peut être une collection, éventuellement **undefined**, figurant à l'intérieur du bloc `<member>`) :
  - \* `<array>...</array>`  
→ éléments produits par *xwrite\_array\_leaf*
    - \* **NB** Le nombre total d'éléments n'est pas exporté
  - \* `<list>...</list>`
  - \* `<set>...</set>`
    - \* **NB** Le type du **set** n'est pas exporté

# gen\_trav\_obj\_constructed

- \* Bien que ce soit une fonction de traversée, il a été nécessaire de la modifier pour produire le contenu d'un **arrow** : paires (clé, valeur)

```
<item>  
  <key>  
    <!-- Clé -->  
  </key>  
  <value>  
    <!-- Objet associé à la clé -->  
  </value>  
</item>
```

- \* NB Ordre (key, value) ?
- \* Sortie déterministe ? (ordre des clés)

# xwrite\_array\_leaf

- \* Usage : éléments d'un *array*

<elem>

<!-- Contenu de l'élément -->

</elem>

- \* **NB** L'indice de l'élément n'est pas exporté

# xwrite\_leaf\_in (1/2)

- \* Usage : production des feuilles

- \* Différents cas de figure :

## 1. **Tabulated** :

- \* **Undefined tabulated** → appel de `xwrite_null()`

- \* Sinon → **domaine tabulé** :

```
<obj tabulated="on" dom="date" key= "_d39017000" />
```

- \* **TODO** : inclure/pointer vers les données tabulées

## 2. **External** :

- \* `<external dom="login" >`

```
<!-- Données externes sous forme de chaîne -->
```

```
</external>
```

- \* **TODO** : spécifier un format pour les données externes ?

# xwrite\_leaf\_in (2/2)

- \* Différents cas de figure, **suite** :

## 3. **Inlinable** :

- \* **Unit** :

<**unit** />

- \* **Bool** :

<**bool**>1</**bool**>

- \* **Int** :

<**int**>2</**int**>

- \* **Float** :

<**float**>3.0</**float**>

- \* **String** :

<**string**>quatre</**string**>

- \* NB : undefined, échappement

# xwrite\_null

- \* Ecriture d'un objet "undefined"
- \* `<undefined />`

# Nœuds partagés

- \* Définition d'un nœud partagé : *xwrite\_define\_shared\_node*  
<shared\_node id="8" />
- \* Référence à un nœud partagé : *xwrite\_shared\_node*  
<link\_shared\_node id="8" />
- \* TODO : références XML ?

# Principe de l'implémentation

- \* Versions modifiées de `gen_write`, etc.
  - \* Ex. : `gen_write` renommée en `gen_any_write`,
    - \* Paramètre additionnel : `bool xml_p`
    - \* `gen_write` et `gen_xwrite` appellent `gen_any_write`, avec `xml_p = FALSE` ou `TRUE` respectivement.
  - \* En principe, les fonctions de traversée d'objets restent inchangées
    - \* Exception : *`gen_trav_obj_constructed`*, mais les modifications ne concernent pas le mécanisme de la traversée proprement dit.
- \* Fichiers concernés :
  - \* *`genClib.c`*, *`genC.h`*, *`genC.c`*
  - \* Branche SVN : [svn.cri.enscm.fr/svn/newgen/branches/daverio](http://svn.cri.enscm.fr/svn/newgen/branches/daverio)