

**3iaxper : manuel de référence.****v1.1****Synopsis.**

```
3iaxper -g base
```

**Description.**

Le programme `3iaxper` produit un fichier au format `.3ia` à partir de la base Xper2 `base`. Le fichier `.3ia` est produit sur `stdout`.

`base` est le nom de la base Xper2, éventuellement précédé de son chemin. Ce nom constitue la base des noms de fichiers constituant effectivement la base. `3iaxper` utilise les fichiers suivants :

<code>base.xpd</code>	Description de l'univers traité.
<code>base.xpx</code>	Relations entre les états des descripteurs et les taxons.
<code>base.xpr</code>	Relations père / fils entre les descripteurs.
<code>base.xpt</code>	Fichier <i>d'exportation</i> . Note : contrairement aux trois autres, ce fichier n'est pas créé par Xper2 et doit être écrit manuellement.

Auteur : J. Ducasse, juillet 2007.

**Options.**

`g` Option de *debug*, affichant les données telles que `3iaxper` les a lues depuis les fichiers d'entrée, avant toute interprétation, ainsi qu'un certain nombre d'opérations intermédiaires.

**Exemples.**

```
% 3iaxper /home/xper2/baseInsectes > insectes.3ia
% 3ia insectes.3ia out
  La première commande produit le fichier insectes.3ia à partir de la base Xper2 baseInsectes.
  La seconde procède à l'analyse 3ia.
```

**Clefs de configuration.**

Le programme utilise le fichier de configuration `tts.cfg`. Ce fichier est recherché en priorité dans le répertoire courant ou, à défaut, dans le répertoire référencé par la variable d'environnement `TTS_PKG`. Les entrées suivantes sont utilisées :

<code>SED_PROG</code>	Exécutable <code>sed</code> ou équivalent utilisé pour filtrer les fichiers modèles. Par défaut, <code>SED_PROG="sed"</code> . Le programme sera lancée avec les conventions habituelles : on peut donc soit spécifier son chemin complet, soit donner seulement son nom et laisser le système utiliser la variable d'environnement <code>PATH</code> . Note : sur Windows, si <code>sed</code> (utilitaire Unix) n'est pas disponible, on pourra utiliser l'utilitaire <code>minised</code> livré.
<code>FORMAT</code>	La valeur est une série de mots clefs spécifiant des options de format en sortie. Les valeurs suivantes sont acceptées : <code>vg/bl</code> Format des arbres produits : les nœuds frères sont séparés par une virgule si <code>vg</code> est spécifié, un blanc si <code>bl</code> est spécifié (valeur par défaut).

**Format des fichiers en entrée.**Fichier `base.xpd`.

Format XML. Les informations extraites sont :

- Le titre de l'univers.
- La liste des taxons, avec pour chacun d'eux son identifiant et son libellé.
- La liste des descripteurs, avec pour chacun d'eux son identifiant et son libellé.
- Pour chaque descripteur, la liste de ses états, avec pour chacun d'eux son identifiant et son libellé.

Note : l'extraction des informations par 3iaxper est assez sommaire et n'utilise pas toute la souplesse de XML : le fichier généré par Xper2 ne doit pas être trop modifié manuellement !

#### Fichier *base.xpx*.

Chaque ligne donne le ou les états d'un taxon pour un descripteur. Elle est formée d'un entier, qui doit être interprété comme un masque de bits, chaque bit correspondant à un état et ayant la valeur 1 ou 0 selon que le taxon a ou n'a pas cet état.

Les descripteurs, taxons et états sont identifiés par leur position (indice), qui correspond à leur identifiant défini dans le fichier *base.xpd*. Ainsi la valeur de l'état E du descripteur D pour le taxon T est-elle donnée par le bit E-1 (compté à partir de 0 à droite) de l'entier à la ligne L (comptée à partir de 1), avec  $L = (T-1) \times N + D$ , où N est le nombre de descripteurs de l'univers.

Remarque : il existe une certaine incohérence entre l'identification par un identifiant numérique *a priori* libre dans le fichier *base.xpd* et l'adressage par position dans le fichier *base.xpx*. En particulier, la signification de N est ambiguë dans le cas où la série des identifiants ne serait pas consécutive. Bien que dans la pratique les identifiants générés automatiquement par Xper2 soient consécutifs à partir de 1, 3iaxper est plus souple et utilise les règles suivantes :

- Les identifiants sont numériques et >1, et tous différents (ce dernier point n'est cependant pas contrôlé).
- La série des identifiants peut comporter des trous.
- La série des identifiants peut être dans le désordre.
- La position est toujours calculée par la formule précédente. Les trous de la série doivent donc être occupés par des valeurs sans signification.
- Le nombre N est la valeur du plus grand identifiant de descripteur.

Cette politique permet par exemple de supprimer manuellement un taxon (ou état ou descripteur (sauf le dernier)) pour un test dans le fichier *base.xpd*, sans avoir à modifier les autres fichiers.

#### Fichier *base.xpr*.

Ce fichier donne les relations de dépendance entre les descripteurs. Chaque descripteur est représenté par un groupe de trois lignes, le N<sup>ième</sup> groupe (à compter de 1) correspondant au descripteur d'identifiant N (voir la remarque ci-dessus concernant la relation identifiant / position).

Dans le groupe de 3 lignes correspondant au descripteur D, la première donne l'identifiant du descripteur père de D, ou 0 s'il n'en a pas. Seule cette ligne est utilisée par 3iaxper.

#### Fichier *base.xpt*.

Voir par ailleurs la définition complète de ce fichier. Seules les sections Echantillonnage et Representation-3ia sont utilisées par 3iaxper. Les lignes commençant par "#" sont des commentaires.

#### Exemple :

```
Echantillonnage
  1-10 15-25
;
Representation-3ia
[1] (1:1 (1:2))
[2] (!2:2 (2:2 (3:2) (3:1)))
[3] ((4:1) (4:2+3))
[4] (!1:3+4 (1:3+4))
;
```

#### • Section Echantillonnage :

Cette section permet de limiter l'ensemble des taxons utilisés pour générer le fichier en sortie. Elle est constituée d'une seule ligne donnant les identifiants des taxons à prendre en compte, soit unitairement, soit sous forme

d'intervalle avec le séparateur "--". Les numéros ne correspondant pas à des identifiants définis dans l'univers sont ignorés silencieusement.

La restriction de l'univers à un sous-ensemble de taxons a les conséquences suivantes :

- Seuls les taxons de ce sous-ensemble seront produits dans la section `Taxons` en sortie.
- Seuls les taxons de ce sous-ensemble seront pris en compte dans les expansions de modèles de caractères donnés par la section suivante.

Si la section `Echantillonnage` manque, il n'y a pas de restriction et tous les taxons de la base sont pris en compte.

• Section `Representation-3ia` :

Cette section décrit un ensemble de modèles de caractères, formés d'un identifiant suivi d'un arbre. Les caractères sont définis par compréhension : les taxons sont représentés par des définitions concernant les taxons ayant ou n'ayant pas tel état pour tel descripteur.

La syntaxe est la suivante :

- Un caractère est de la forme :  $(A (B (C (D) ) ) )$
- Chaque terme de niveau zéro A, B, etc. est formé de :
  - un terme de niveau un X,
  - ou plusieurs termes de niveau un X séparés par une virgule.
- Chaque terme de niveau un X est formé :
  - de la négation "!" optionnelle,
  - d'un numéro de descripteur D,
  - du séparateur ":",
  - de un ou plusieurs numéros d'état séparés par "+".

Exemple :  $(!3:1+2+3,4:6,!5:8+9 (.....) )$

Cas particuliers :

- Des termes de niveau zéro séparés par un sous-arbre sont équivalents aux mêmes termes séparés par une virgule :  $(X (Y (...)) Z)$  est équivalent à  $(X, Z (Y (...)))$ .
- La négation concerne uniquement le terme de niveau 1 qui suit.
- Le "+" entre les états représente l'alternative (ou disjonction non exclusive, souvent notée  $\vee$ ) :  $3:1+2$  signifie présence de l'état 1 ou de l'état 2 ou des deux.
- L'ordre des états séparés par "+", ainsi que celui des termes séparés par ",", est sans importance.

### Génération des caractères.

Chaque modèle de caractère fourni dans la section `Representation-3ia` du fichier `base.xpt` génère un caractère de même identifiant dans le fichier en sortie.

L'expansion des termes du modèle en listes de taxons est basée sur les informations suivantes extraites de la base `Xper2` :

- Pour un taxon donné, chacun des descripteurs de l'univers peut être défini ou non défini.
- Pour un descripteur défini, le taxon possède certains états du descripteur (éventuellement zéro) et pas les autres.
- Par ailleurs, chaque descripteur peut être isolé ou intégré dans une chaîne de descripteurs par une relation père/fils.

La génération des caractères procède en deux étapes : l'expansion des modèles, suivie de l'application d'un certain nombre de règles pouvant amener à la production de messages d'avertissement et/ou à la modification du caractère.

#### 1. Expansion du modèle.

Les règles suivantes s'appliquent pour interpréter les modèles de caractères :

- Le terme  $D : E+F$  représente tous les taxons de l'univers pour lesquels  $D$  est défini et ayant au moins l'état  $E$  ou l'état  $F$  ou les deux pour le descripteur  $D$ . Le taxon peut avoir en plus d'autres états du même descripteur.
- Le même terme précédé de la négation (ex. :  $!D : E+F$ ) représente le complémentaire. Plus précisément, un taxon  $T$  est inclus dans la liste générée par  $!D : E+F$  :
  - si  $D$  est défini pour  $T$  et si  $T$  a un état autre que  $E$  et  $F$  pour le descripteur  $D$  (il peut aussi avoir l'état  $E$  et/ou l'état  $F$  en sus).
  - ou bien si :
    - le descripteur  $D$  est non défini pour  $T$ ,
    - ou  $T$  n'a aucun état de  $D$  (bien qu'il soit défini),
 et si en même temps :
  - le descripteur  $D'$ , père de  $D$ , s'il existe, est défini pour  $T$ ,
  - ou si le descripteur  $D''$ , père de  $D'$ , s'il existe, est défini pour  $T$ ,
  - et ainsi de suite jusqu'au descripteur de plus haut niveau (\*).
 Autrement dit, le taxon  $T$  est accepté s'il a un état autre que  $E$  et  $F$ , ou bien s'il n'a aucun état mais que l'un quelconque des ascendants du descripteur  $D$  est tout de même défini pour  $T$ .
- Plusieurs termes de niveau un, avec ou sans négation, peuvent être cumulés avec le séparateur virgule. La liste finale obtenue est l'union des listes fournies par chaque terme. Les listes des différents termes sont calculées de façon tout à fait indépendante.
- Dans tous les cas, si l'univers a été restreint par une directive `Echantillonnage`, les taxons n'appartenant pas à la restriction sont éliminés.

(\*) Si, lors de la remontée des descripteurs, un ou plusieurs descripteurs non définis pour  $T$  (y compris  $D$ ) sont rencontrés avant de trouver un descripteur défini, la description du taxon est dite *incomplète* pour ce descripteur.

## 2. Application de règles et contraintes.

Une fois les caractères créés par expansion des modèles comme spécifié en 1, les règles suivantes sont appliquées. Noter que seules les règles 4, 5 et 6 modifient le caractère produit, les autres règles ayant pour seule conséquence l'affichage d'un message d'avertissement.

Remarques préliminaires :

- Ces règles sont appliquées dans l'ordre dans lequel elles sont présentées ci-dessous. L'ordre d'application n'est pas indifférent.
- Dans tous les traitements, les classes portant sur un même descripteur mais exprimées par des termes séparés sont traitées indépendamment. Ainsi,  $D : a, E : b, D : c$  peut éventuellement donner des résultats différents de  $D : a+c, E : b$ .

### • Règle 1 : Contrôle de redondance de classes.

Lorsqu'un terme  $D : a+b$  est utilisé plusieurs fois, que ce soit dans le même caractère, y compris dans un même nœud, ou dans des caractères différents, un message d'avertissement est affiché.

Cas particuliers :

- Les termes avec négation sont ignorés par ce traitement. Les redondances de la forme :
  - $!D : a+b$  exprimé plusieurs fois,
  - $D : a+b$  et  $!D : a+b$  exprimés,
 ne sont donc pas signalées.
- Les compositions en états doivent être identiques pour être signalées. Ainsi, les termes  $D : a$  et  $D : a+b$  ne sont pas signalés comme redondants. Par contre,  $D : a+b$  et  $D : b+a$  sont signalés comme redondants.

### • Règle 2 : Négation hors de la racine.

Lorsqu'un terme avec une négation " ! " apparaît sur un nœud autre que le nœud racine du caractère, un message d'avertissement est affiché.

### • Règle 3 : Relations père/fils entre descripteurs.

Tous les couples de descripteurs sans relation père/fils apparaissant dans un même caractère, y compris dans un même nœud, sont signalés. La relation est entendue au sens large : relation ascendant/descendant avec d'éventuels niveaux intermédiaires.

Les nœuds avec négation sont pris en compte comme les autres.

- Règle 4 : Suppression des taxons générés par des descripteurs en relation père/fils.

Si un caractère possède deux nœuds N1 et N2, associés chacun à deux descripteurs D1 et D2, et si :

- D1 et D2 sont liés par une relation père/fils, dans laquelle D2 est descendant de D1,
- et N2 est dans le sous-arbre descendant de N1 (sous-arbre strict, N1 différent de N2),

alors on supprime de la liste des taxons générés par le descripteur D1 sur N1 tous les taxons également générés par le descripteur D2 sur N2.

Cas particuliers :

- Les états des taxons sur D1 et D2 n'ont aucune importance dans ce mécanisme.
- Les termes avec négation sont ignorés de ce traitement.
- Un taxon supprimé de la liste générée par D1 peut demeurer présent sur N1, dans le cas où il aurait aussi été généré par un autre descripteur D3. Exemple : classe de N1 : D1 : a, D3 : c ; classe de N2 : D2 : b. Si D1 : a, D2 : b et D3 : c génèrent tous trois le taxon T, et même si D2 est descendant de D1, le taxon T subsistera sur les deux nœuds N1 et N2 (polymorphisme).

- Règle 5 : Elimination des nœuds vides.

Les nœuds vides sont éliminés. Il s'agit des nœuds ne portant ni taxon ni sous-arbre. Par contre les nœuds portant uniquement un sous-arbre et pas de taxons ne sont pas éliminés.

- Règle 6 : Elimination des redondances.

Pour chaque nœud de chaque caractère, la liste des taxons générés par l'ensemble des descripteurs est nettoyée des redondances.

- Règle 7 : Affichage du polymorphisme.

Les taxons apparaissant plusieurs fois dans un même caractère, constituant donc un polymorphisme, sont signalés. Ce traitement ayant lieu *après* le précédent, il s'agit forcément de redondances entre nœuds, les nœuds étant forcément non redondants.

### Messages affichés.

#### En mode normal :

En plus des messages d'erreur entraînant un arrêt immédiat du programme, les messages d'information suivants sont affichés :

```
! [3] Classe 2:2+3 deja utilisee dans car. [2]
    Application de la règle n° 1.
```

```
! [3] Negation ailleurs que sur la racine
    Application de la règle n° 2.
```

```
! [3] Descripteurs sans relation pere/fils : 3 5
    Application de la règle n° 3.
```

```
! [3] Taxon 7 supprime du descr. 4 ascendant du descr. 7
    Application de la règle n° 4. Affiché seulement avec l'option -g.
```

```
! [3] Noeud vide supprime
    Application de la règle n° 5.
```

! [3] Taxon 7 polymorphe  
Application de la règle n° 7.

! Descripteur 5 : description incomplete pour le taxon 7  
Signale un taxon pour lequel la description est incomplète pour le descripteur indiqué, selon la définition donnée plus haut.

#### Avec l'option -g :

Si l'option -g est fournie, les informations supplémentaires suivantes sont produites :

Taxons generes par les conditions :

Lors de l'analyse des modèles de caractère, pour chaque terme de niveau 1 est affichée la liste des taxons générée. Les taxons générés, dans un terme avec négation, par héritage d'un descripteur non défini sont signalés par un astérisque. Exemple :

2:1+2 → 3 4 \*5

Titre :

Titre de l'univers.

Descripteurs/Etats :

Donne la liste des descripteurs et, pour chacun d'eux, son père et son fils (0 si non concerné), puis la liste des états que peut prendre ce descripteur.

Taxons/Masques :

Donne la liste des taxons. Ceux non pris en compte parce qu'éliminés par la section Echantillonnage du fichier *base.xpt* sont signalés par un astérisque. Ensuite, pour chaque descripteur est affichée la liste des états de celui-ci que peut prendre le taxon, ou la mention « non défini » si ce descripteur n'est pas défini pour ce taxon. La mention « description incomplète » est ajoutée le cas échéant.

Caracteres :

Les caractères produits sont enfin affichés.

#### **Format du fichier en sortie.**

Le fichier produit constitue un fichier au format pris en charge par le programme 3ia renseigné par les informations calculées à partir de la base Xper2 :

- Le titre est celui de la base Xper2.
- La section Dimensions est renseignée.
- La section Taxons donne la liste des taxons définis dans la base, éventuellement restreinte par une directive Echantillonnage ; les taxons n'apparaissant dans aucun caractère figurent tout de même ici. Les descripteurs pour lesquels la description du taxon est incomplète (voir définition plus haut) sont signalés.
- La section Referentiel est renseignée comme suit : (*à faire*).
- La section Descriptions reprend la liste des caractères donnée par le fichier d'exportation, où les expressions de conditions ont été substituées par les taxons correspondants comme expliqué plus haut.

#### **Historique.**

version 1.1 : décembre 2009

Changement de nom → 3iaxper.

version 1 : juillet 2007