

Tri alternatifs dans Spectre

Description

Nous avions déjà évoqué les tris dans les astuces précédentes :

https://blog.difrance.com/tri-naturel-versus-tri-alphanumerique/ https://blog.difrance.com/changer-lordre-de-tri-des-mois/ https://blog.difrance.com/dimension-triee-en-fonction-d-une-autre-colonne/

Lâ??astuce actuelle se focalise sur la mise en place de tris via Spectre.

Nous souhaitons répondre aux questions suivantes :

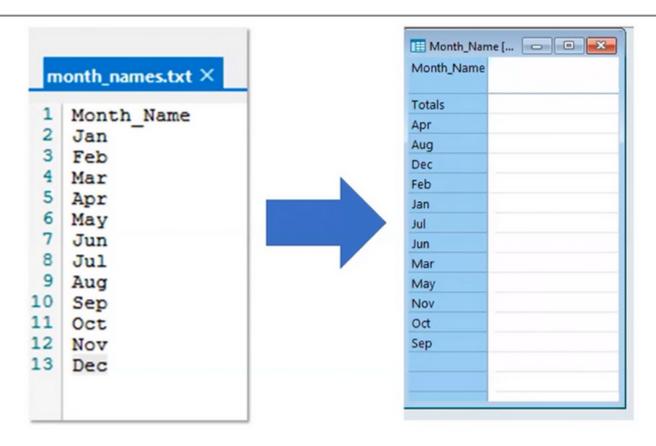
Comment créer une Dimension avec les noms de mois débutant par avril ?

Comment définir les valeurs de Dimension dans un ordre chronologique, c'est à dire commençant par janvier?

La réponse est lâ??utilisation d'un tri alternatif.

Le tri naturel est le tri par défaut dans Spectre. Les chaines de caractÃ"res comme les noms de mois, par exemple "Janvier", sont triés par ordre alphabétique (cf. https://blog.difrance.com/tri-naturel-versus-tri-alphanumerique/).





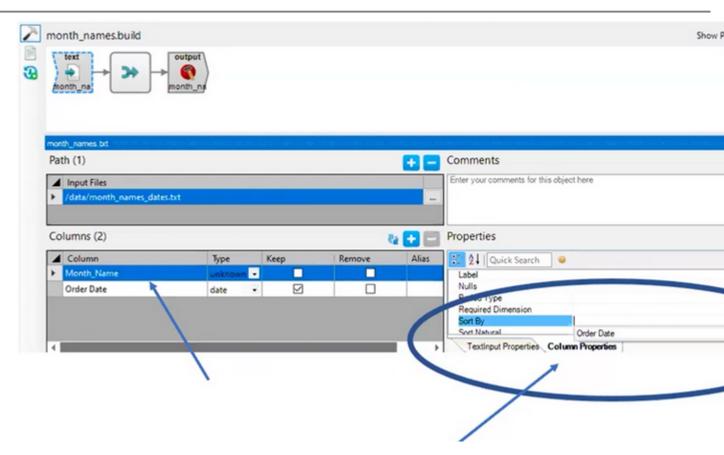
Fichier texte cBase

Un tri alternatif peut être appliqué pour modifier l'ordre d'affichage initial des valeurs de Dimension.

Un tri alternatif requiert lâ??utilisation de l'attribut **sort-by** et une colonne de tri dans le flux de données (ou une expression dans un cPlan contenant l'ordre associé.

Dans un script .build, l'option sort-by se trouve ici, au niveau de l'objet d'entrée texte :





Voici l'extrait du m \tilde{A}^a me script faisant r \tilde{A} ©f \tilde{A} ©rence \tilde{A} l'option **sort-by** vu depuis l' \tilde{A} ©diteur de texte de Workbench :

```
...
text-input "/data/month_names_alt_sort.txt" {
column "month_name" sort-by="alt_sort_month_names"
...
}
```

Ci-dessous le fichier texte utilis é pour générer le cBase :



```
month_names_alt_sort.txt ×
   Month Name
                 alt sort month names
 2
   Jan 1
 3
   Feb 2
   Mar 3
   Apr 4
 6
   May 5
 7
   Jun 6
 8
   Jul 7
 9
   Aug 8
10
   Sep 9
11
   Oct 10
12
   Nov 11
13
   Dec 12
```

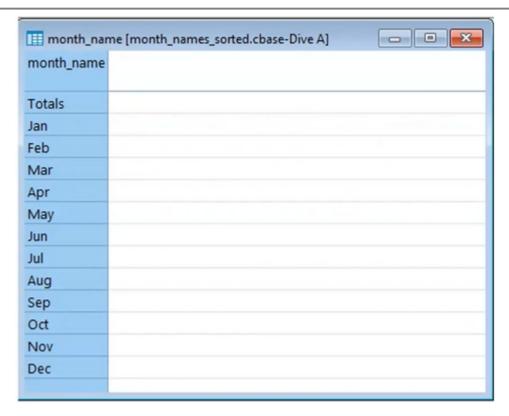
Ci-dessous le script complet utilisé pour générer le cBase :

```
month_names_alt_sort.build X

1 build {
2 text-input "/data/month_names_alt_sort.txt" {
3 column "month_name" sort-by="alt_sort_month_names"
4 column "alt_sort_month_names"
5 }
6 remove "alt_sort_month_names"
7
8 output "/cbases/month_names_sorted.cbase"
9 }
10
```

Voici le résultat obtenu dans ProDiver :





Le tri alternatif peut être appliqué sur toute colonne de chaîne de caractÃ"res.

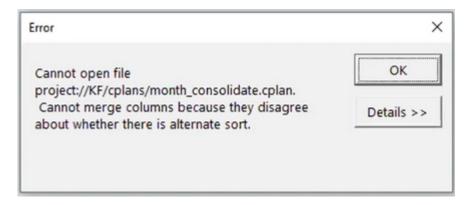
La colonne de tri utilisée doit être une chaîne de caractÃ"re ou un entier (Integer).

Voici un autre exemple de tri :

Catégorie âge patient :

- nouveau né
- enfant
- adolescent
- adulte
- personne âgée

Si l'on combine des cBases dans un cPlan, le tri doit Ãatre identique pour une mÃame Dimension.



Vous pouvez:



- utiliser un fichier Lookup contenant la colonne de tri
- créer une colonne d'ordre de tri au sein du script

L'exemple ci-dessous montre la colonne de tri contenue dans le fichier d'entrée.

```
month_names_alt_sort.build ×
 1 build {
     text-input "/data/month names dates.txt"
 2
 3
        column "month name"
        column "Order Date" type="date"
 4
 5
 6
 7
     add "MO sort order" `month(value("Order Date")
     replace "Month name" 'value ("month name") ' sor
 8
 9
10
     output "/cbases/month names date sort.cbase"
11
12
```

La colonne de tri alternative doit tenir compte de toutes les valeurs de la colonne à trier.

Un tri alternatif peut-être ajouté dans un cPlan en utilisant une expression :

```
month_sort_by_date.cplan X

1 cplan {
    cbase-input "/cbases/month_names_date.cbase"
    dimension "Month" `month_name(value("Order Date"))` sort-by=`month(value("Order Date"))`
}
```

Tags

- 1. cBase
- 2. cplan
- 3. script
- 4. Spectre