
Tri alternatifs dans Spectre

Description

Nous avons 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/>

L'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 ?

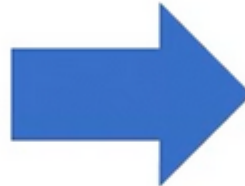
Comment définir les valeurs de Dimension dans un ordre chronologique, c'est à dire commençant par janvier?

La réponse est l'utilisation d'un tri alternatif.

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

month_names.txt	
1	Month_Name
2	Jan
3	Feb
4	Mar
5	Apr
6	May
7	Jun
8	Jul
9	Aug
10	Sep
11	Oct
12	Nov
13	Dec

Fichier texte



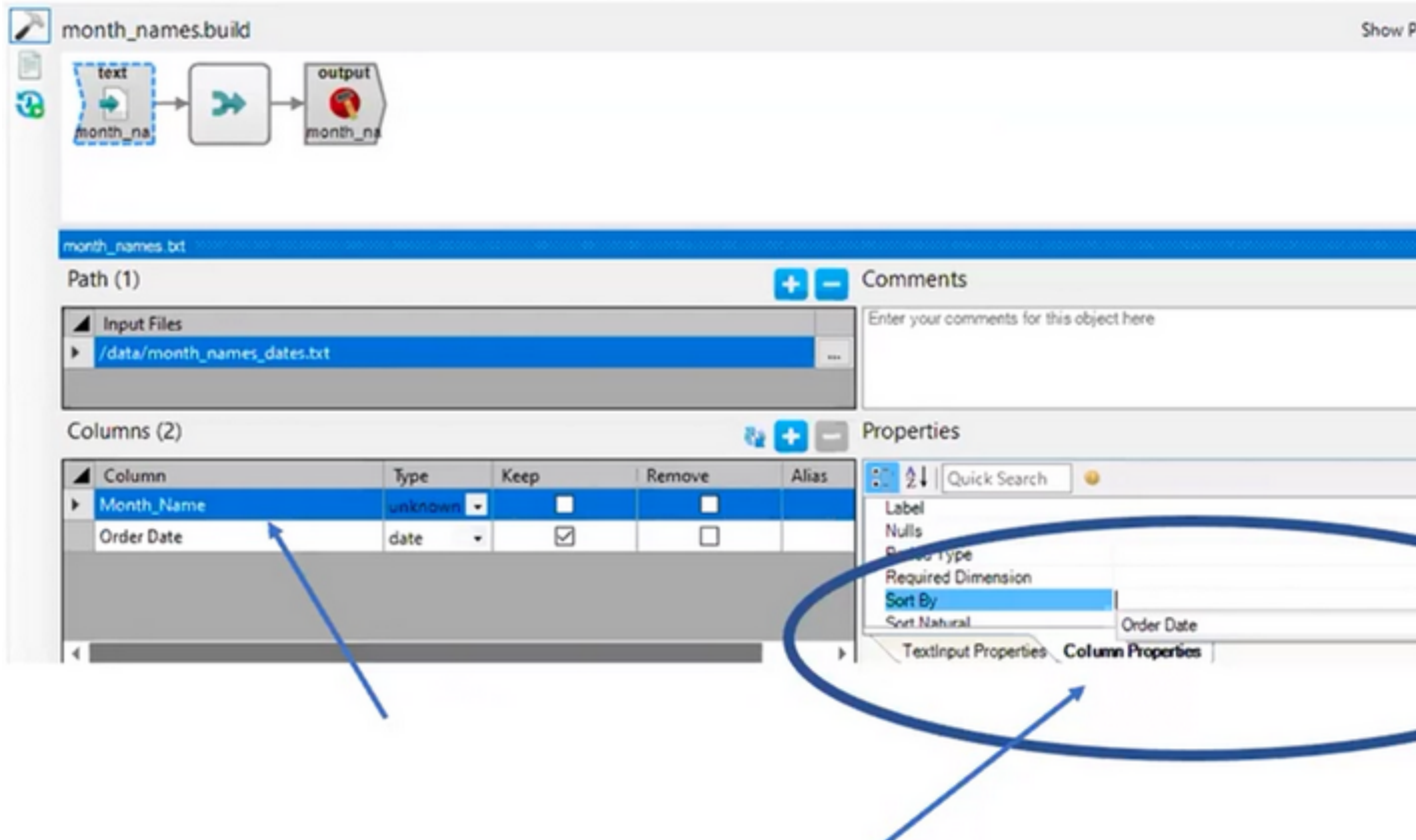
Month_Name [...]	
Month_Name	
Totals	
Apr	
Aug	
Dec	
Feb	
Jan	
Jul	
Jun	
Mar	
May	
Nov	
Oct	
Sep	

cBase

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

Un tri alternatif requiert l'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 même script faisant référence à l'option **sort-by** vu depuis l'éditeur de texte de Workbench :

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

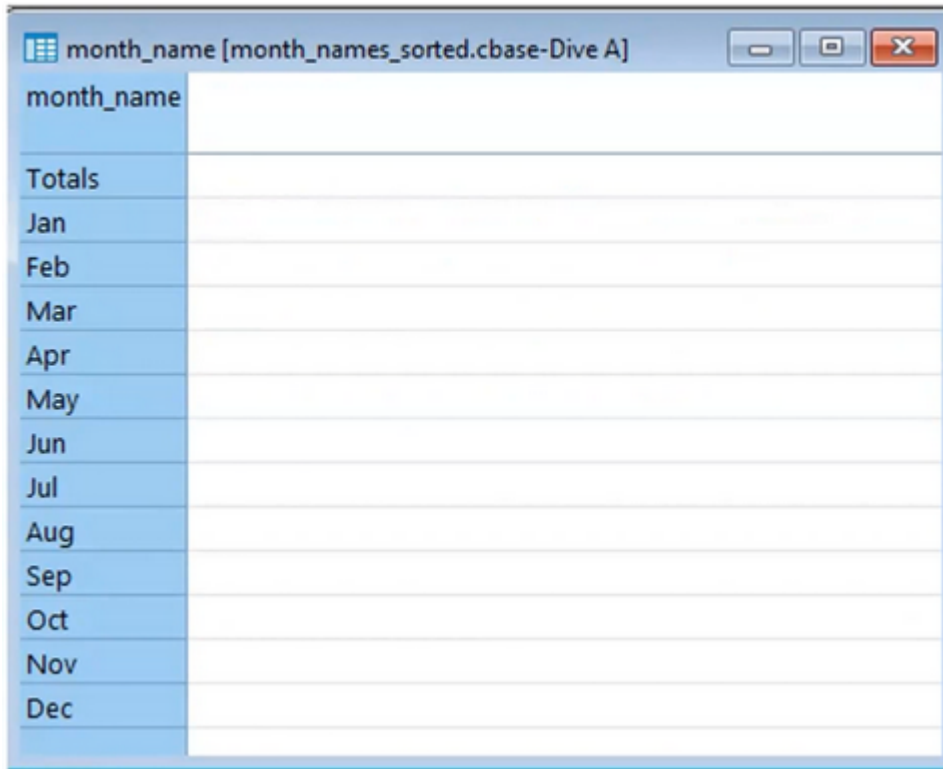
Ci-dessous le fichier texte utilisé pour générer le cBase :

```
month_names_alt_sort.txt X
1 Month_Name alt_sort_month_names
2 Jan 1
3 Feb 2
4 Mar 3
5 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 :



month_name
Totals
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Le tri alternatif peut être appliqué sur toute colonne de chaîne de caractères.

La colonne de tri utilisée doit être une chaîne de caractère 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 être identique pour une même 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 X
1 build {
2   text-input "/data/month_names_dates.txt" {
3     column "month_name"
4     column "Order Date" type="date"
5   }
6
7   add "MO_sort_order" `month(value("Order Date"))
8   replace "Month_name" `value("month_name")` sort
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 {
2
3   cbase-input "/cbases/month_names_date.cbase"
4
5   dimension "Month" `month_name(value("Order Date"))` sort-by=`month(value("Order Date"))`
6
7 }
```

Tags

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