

Intégrer des données d'un fichier Excel™ depuis l'objet Excel-input

Description

Dans Workbench / Spectre 7.2, il est possible d'intégrer des données d'objectifs (budgets, benchmarks, ...) maintenus dans des onglets Excel™. Cela s'effectue via l'objet **Excel-input** dans un script Spectre .build.

Supposons que l'on souhaite récupérer les données du fichier Excel ci-dessous :

The screenshot shows an Excel spreadsheet titled 'objectifs vendeurs.xlsx'. The data is as follows:

	A	B	C	D	E	F	G	H
1	Vendeur	Objectifs						
2	Anselme	434						
3	Aumale	120						
4	Bailey	43						
5	Balme	355						
6	Beaulieu	472						
7	Berger	144						
8	Blondeau	680						
9	Bonnard	382						
10	Boulangier	328						
11	Calvet	294						
12	Cardon	808						

Les données peuvent être intégrées dans un script .build et grâce à un objet **Excel-input** comme ci-dessous :

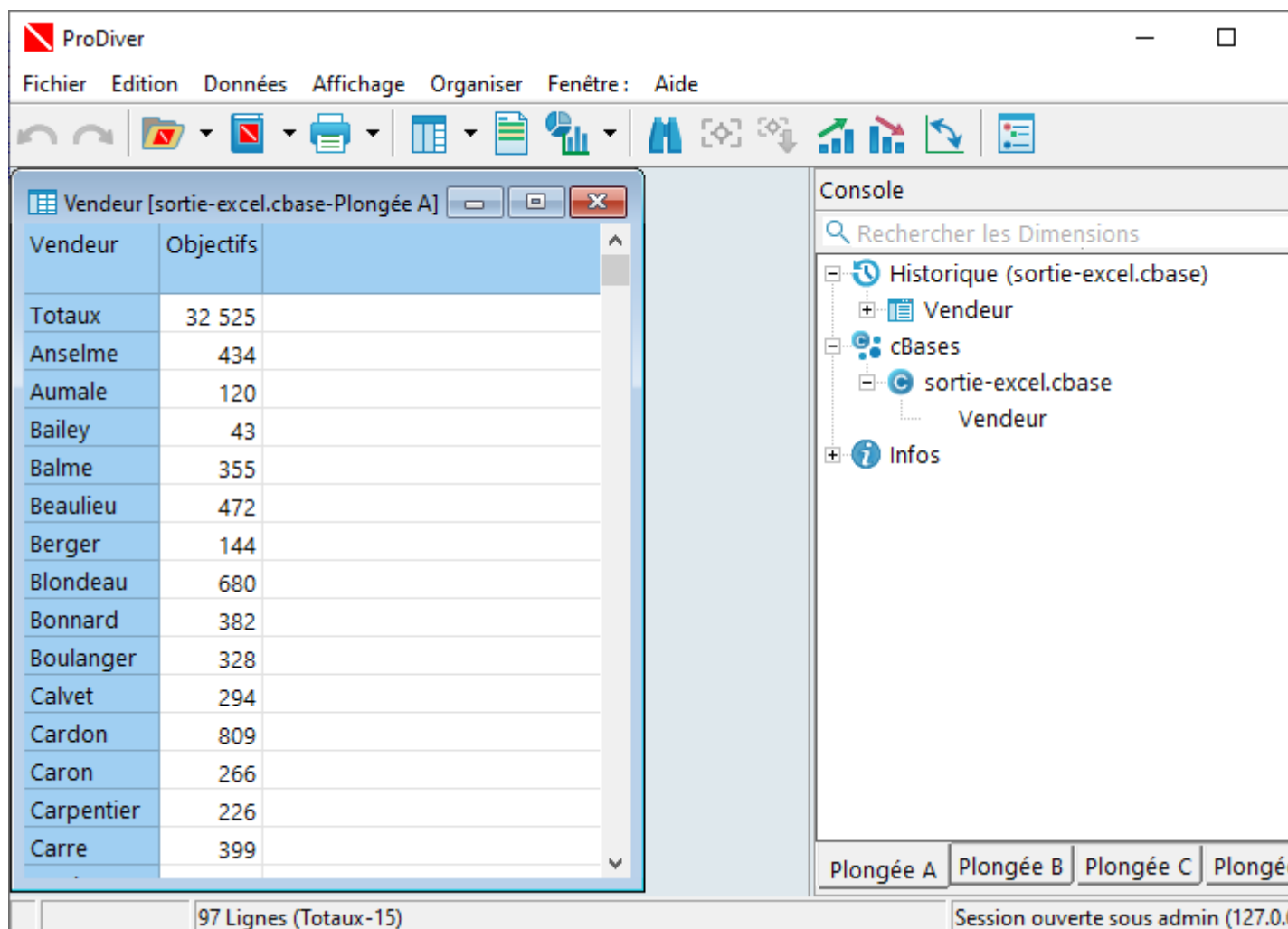


Voici le code du script associé à notre fichier Excel :

```
build {
  //FlowNodeAttributes=""
  //EditorComments="Lit un fichier Excel comme celui-ci :
  //
  //**
  //""
  excel-input "/data/objectifs vendeurs.xlsx" name="excel-input_1"

  output "/cbases/sortie-excel.cbase" name="output_2"
  layout-data {
    node name="excel-input_1" position="8 8"
    node name="concat" position="88 8"
    node name="output_2" position="168 8"
  }
}
```

L'exécution du script génère le cbase **sortie-excel.cbase** avec le contenu suivant :



The screenshot shows the ProDiver interface. The main window displays a table with the following data:

Vendeur	Objectifs
Totaux	32 525
Anselme	434
Aumale	120
Bailey	43
Balme	355
Beaulieu	472
Berger	144
Blondeau	680
Bonnard	382
Boulangier	328
Calvet	294
Cardon	809
Caron	266
Carpentier	226
Carre	399

The console window on the right shows the execution history for 'sortie-excel.cbase', including a tree view with 'Vendeur' and 'Infos' nodes. The status bar at the bottom indicates '97 Lignes (Totaux-15)' and 'Session ouverte sous admin (127.0...'.

Supposons maintenant que nous ayons un fichier Excel contenant des lignes vides :

The screenshot shows the Microsoft Excel interface with the following data in the spreadsheet:

	A	B	C	D	E	F	G	H	I
1	Vendeur	Objectifs							
2									
3									
4	Anselme	434							
5	Aumale	120							
6	Bailey	43							
7	Balme	355							
8	Beaulieu	472							
9	Berger	144							
10	Blondeau	680							
11	Bonnard	382							
12	Boulangier	228							

Lors de l'exécution du script générant le fichier de sortie cbase, les logs indiquent que des lignes vides ont été trouvées et que ces dernières ont été tout simplement ignorées :

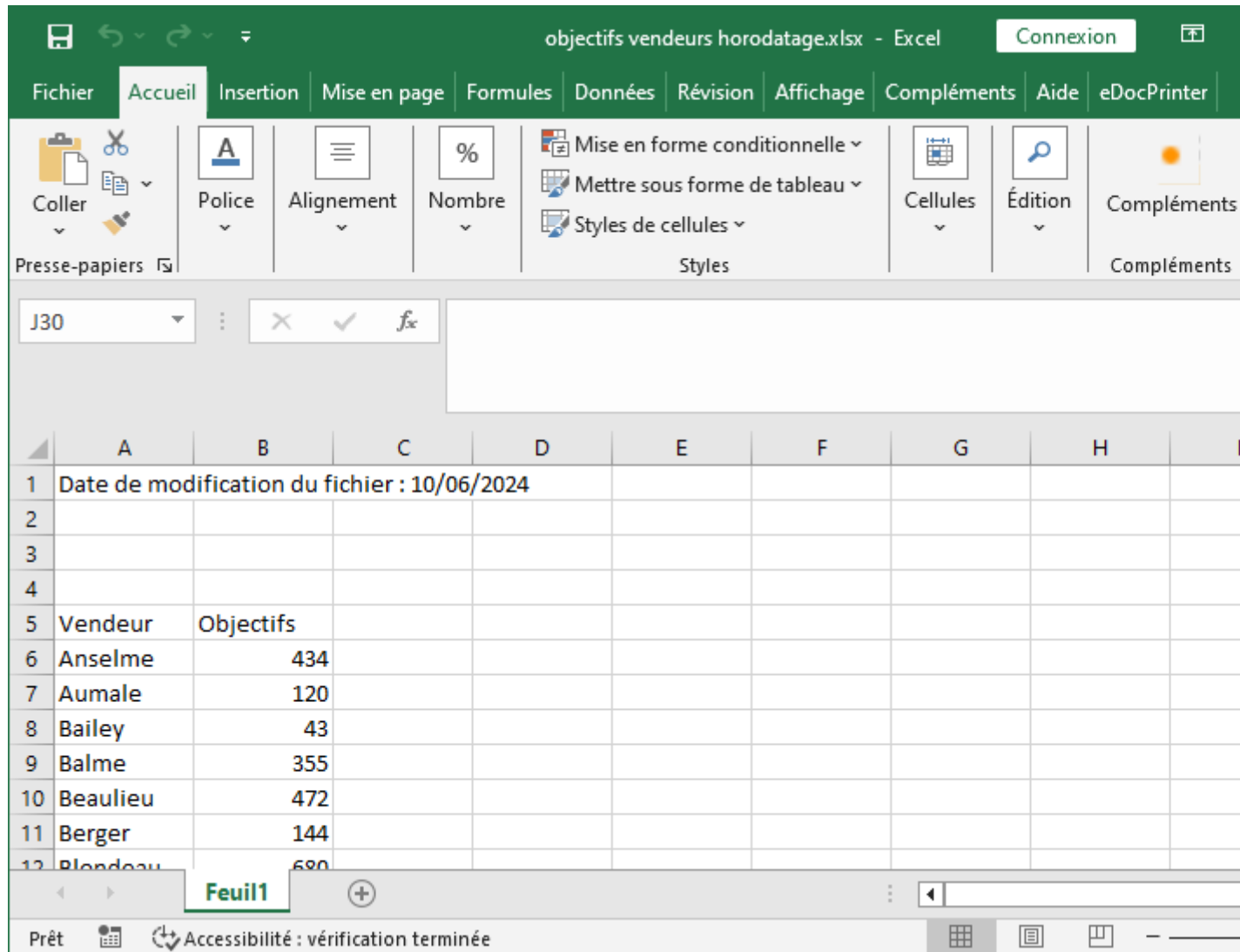
```
Starting run: spectre build /programs/entree excel.build
Working directory: /programs
Command Line: D:\DI_v72\Solution\diveline\bin\spectre.exe build "/programs/entree exc
00:00:00.000: WARNING: <generated>:39: Tag 'proreporter' is deprecated
00:00:00.000: WARNING: <generated>:42: Tag 'reporter-portlet' is deprecated
spectre build 7.2 (14)
Licensed to: Bimaxis

00:00:00.039: Mon Jun 10 15:11:25 2024
00:00:00.041: Temp folder: project://Excel spectre/temp
00:00:00.046: Using first worksheet "Feuill1"
00:00:00.047: D:\DI_v72\DI_Projects\Excel spectre\data\objectifs vendeurs lignes vide
00:00:00.047: D:\DI_v72\DI_Projects\Excel spectre\data\objectifs vendeurs lignes vide

00:00:00.049: Column "Vendeur" is unique for every row
00:00:00.051: 97 rows, 2 columns
00:00:00.051: Sorting strings...
00:00:00.051: Writing D:\DI_v72\DI_Projects\Excel spectre\cbases\sortie-excel.cbase..
Finished in 0,079 seconds with 0 warnings and 0 errors
```

00:00:00.079: This build used 7MB of memory.
00:00:00.079: Total cBase size: 2KB
System Process returned 0

Supposons maintenant que nous ayons un fichier Excel dont les données utiles commencent à une ligne précise :



Dans notre cas les données utiles débutent à la ligne 5, ligne où se trouvent les en-têtes.

Le script permettant de lire les données de l'onglet Excel à partir de la ligne 5 est le suivant :

```
build {
  //FlowNodeAttributes=""
  //EditorComments="Lit un fichier Excel comme celui-ci :
  //
  //**
  //""
  excel-input "/data/objectifs vendeurs horodatage.xlsx" start-row=5
  name="excel-input_1"
  output "/cbases/sortie-excel.cbase" name="output_2"
  layout-data {
    node name="excel-input_1" position="8 8"
    node name="concat" position="88 8"
```


```
node name="output_2" position="168 8"  
}  
}
```

Si dans Workbench, on bascule sur l'éditeur de flux du script, afin d'avoir une interface graphique, en cliquant sur l'objet **Excel-input** on retrouve la propriété dans la zone encadrée :

Comments

Lit un fichier Excel comme celui-ci :

Properties

Recherche rap... 

Excel Options

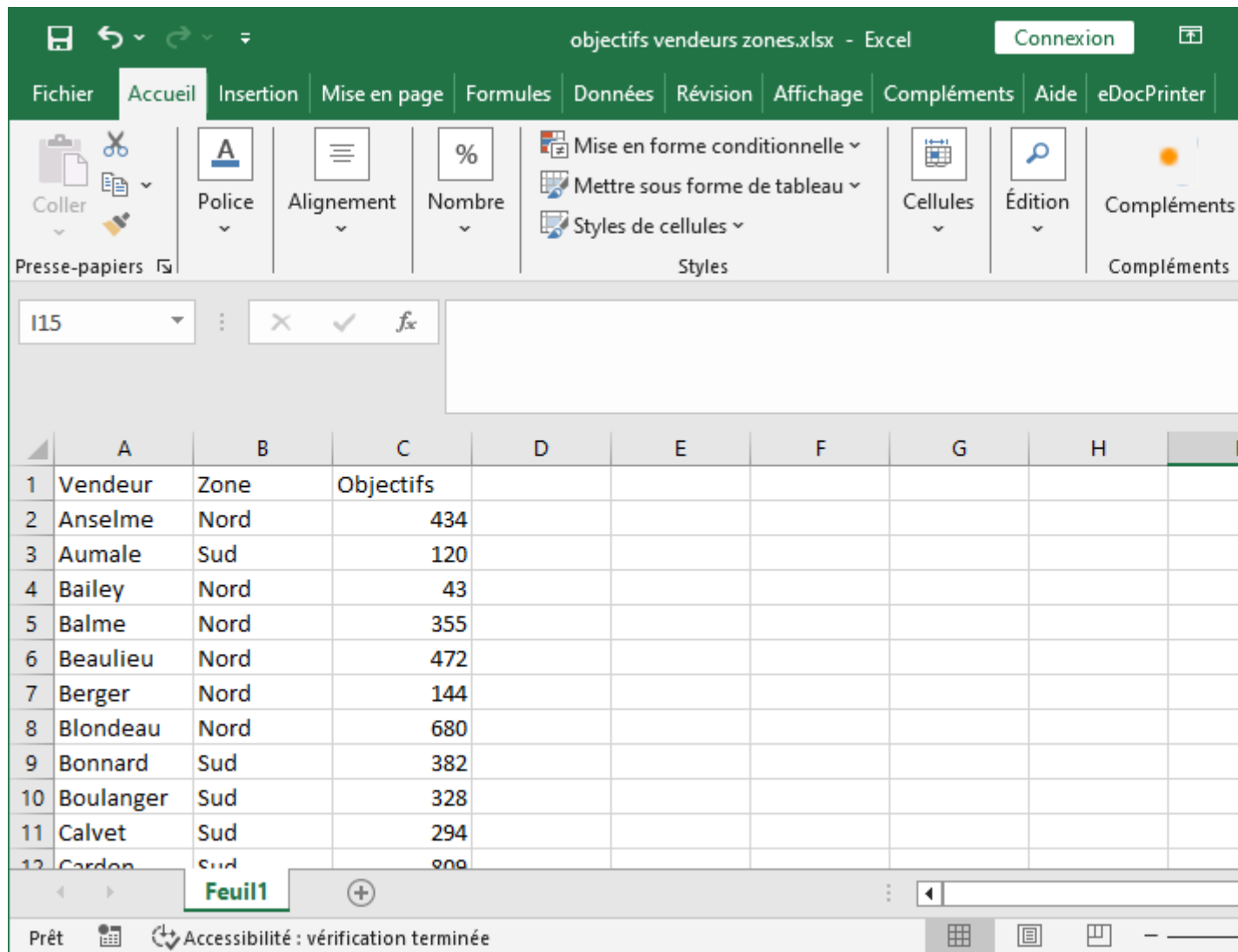
Include Other Columns	False
Headers	
Limit Rows	
Password	
Prefix	
Range	
Sheet	
Start Row	5
Trim	

Input

Excel File Path	/data/objectifs vendeurs horodatage.xlsx
-----------------	--

Excel Input Properties | Column Properties

Supposons maintenant que l'on ait le fichier Excel ci-dessous et que l'on souhaite récupérer les données uniquement pour la Zone Nord :



objectifs vendeurs zones.xlsx - Excel

Connexion

Fichier Accueil Insertion Mise en page Formules Données Révision Affichage Compléments Aide eDocPrinter

Coller Presse-papiers Police Alignement Nombre Styles Mise en forme conditionnelle Mettre sous forme de tableau Styles de cellules Cellules Édition Compléments

I15

	A	B	C	D	E	F	G	H
1	Vendeur	Zone	Objectifs					
2	Anselme	Nord	434					
3	Aumale	Sud	120					
4	Bailey	Nord	43					
5	Balme	Nord	355					
6	Beaulieu	Nord	472					
7	Berger	Nord	144					
8	Blondeau	Nord	680					
9	Bonnard	Sud	382					
10	Boulanger	Sud	328					
11	Calvet	Sud	294					
12	Cardon	Sud	809					

Feuil1

Prêt Accessibilité : vérification terminée

Le script permettant de lire les données uniquement pour cette zone est le suivant :

```
build {
  //FlowNodeAttributes=""
  //EditorComments="Lit un fichier Excel comme celui-ci :
  //
  //**
  //""
  excel-input "/data/objectifs vendeurs zones.xlsx" name="excel-input_1"
  {
    filter `value("Zone")="Nord"`
  }
  output "/cbases/sortie-excel.cbase" name="output_2"
  layout-data {
    node name="excel-input_1" position="8 8"
    node name="concat" position="88 8"
    node name="output_2" position="168 8"
  }
}
```

L'exécution du script génère le cbase **sortie-excel.cbase** avec le contenu suivant :

The screenshot shows the ProDiver software interface. The main window displays a table with the following data:

Zone	Objectifs
Totaux	17 293
Nord	17 293

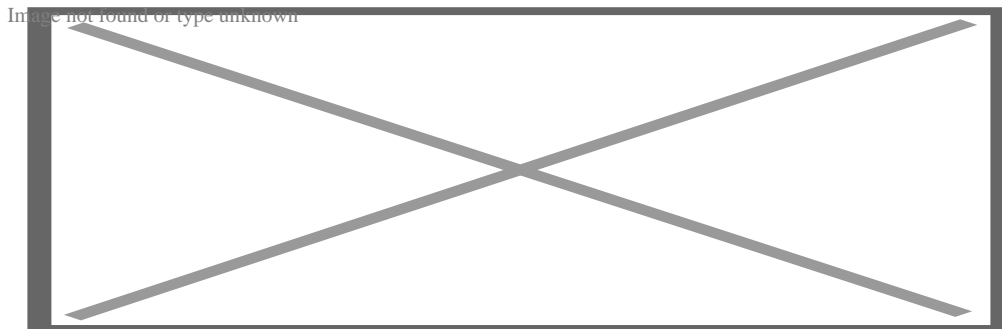
The console window on the right shows a tree structure of dimensions:

- Historique (sortie-excel.cbase)
 - Zone
- cBases
 - sortie-excel.cbase
 - Vendeur
 - Zone
- Infos

At the bottom, there are buttons for 'Plongée A', 'Plongée B', 'Plongée C', and 'Plongée D'. The status bar indicates '1 Lignes (Totaux-1)' and 'Session ouverte sous admin (127.0.0.1)'.

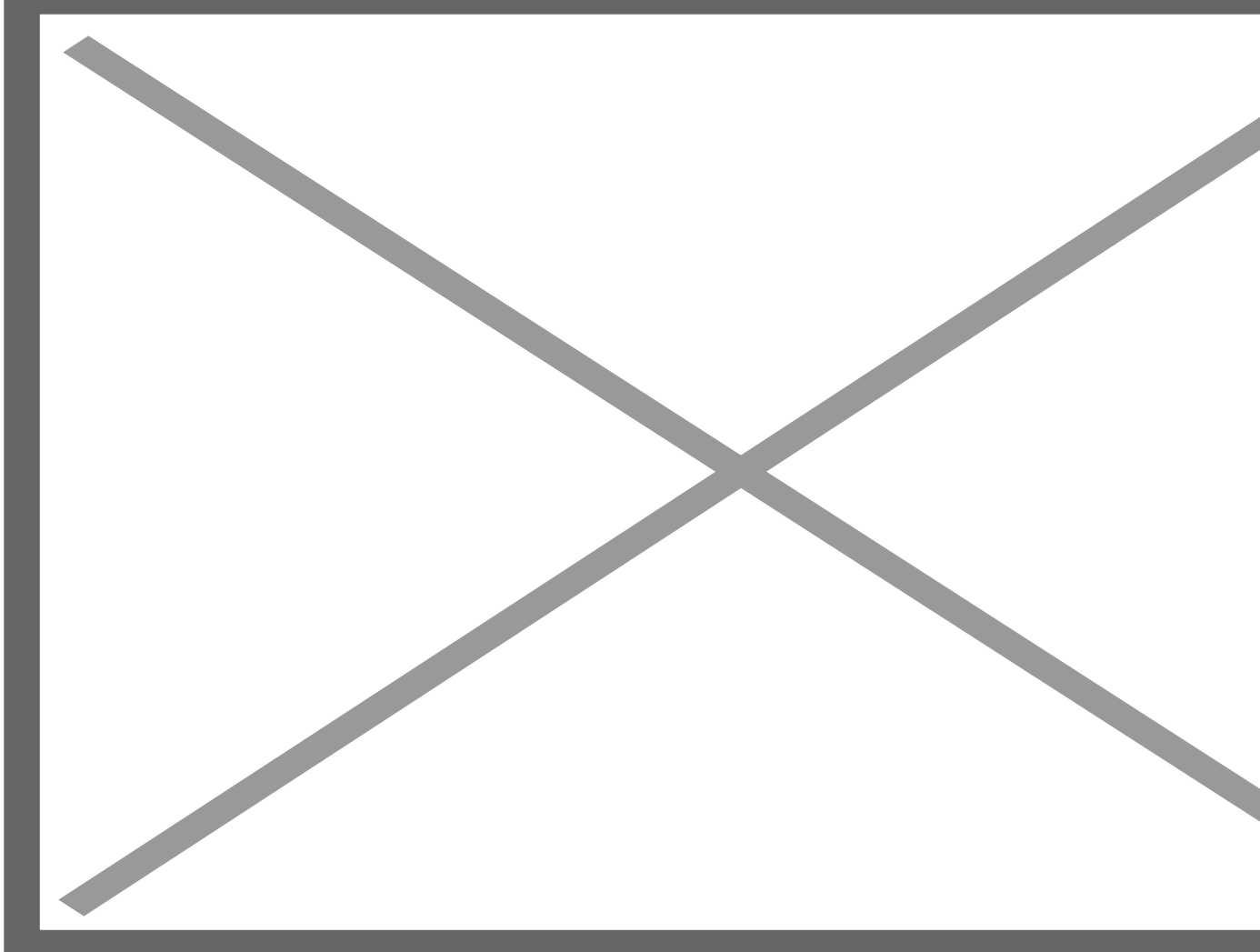
Les données ont bien été filtrées sur la zone souhaitée.

Si dans Workbench, on bascule sur l'éditeur de flux du script, afin d'avoir une interface graphique, on s'aperçoit qu'un objet **filter** est présent avec les propriétés suivantes :



Supposons que l'on souhaite récupérer les données uniquement dans la zone sélectionnée du fichier Excel ci-dessous :

Image not found or type unknown

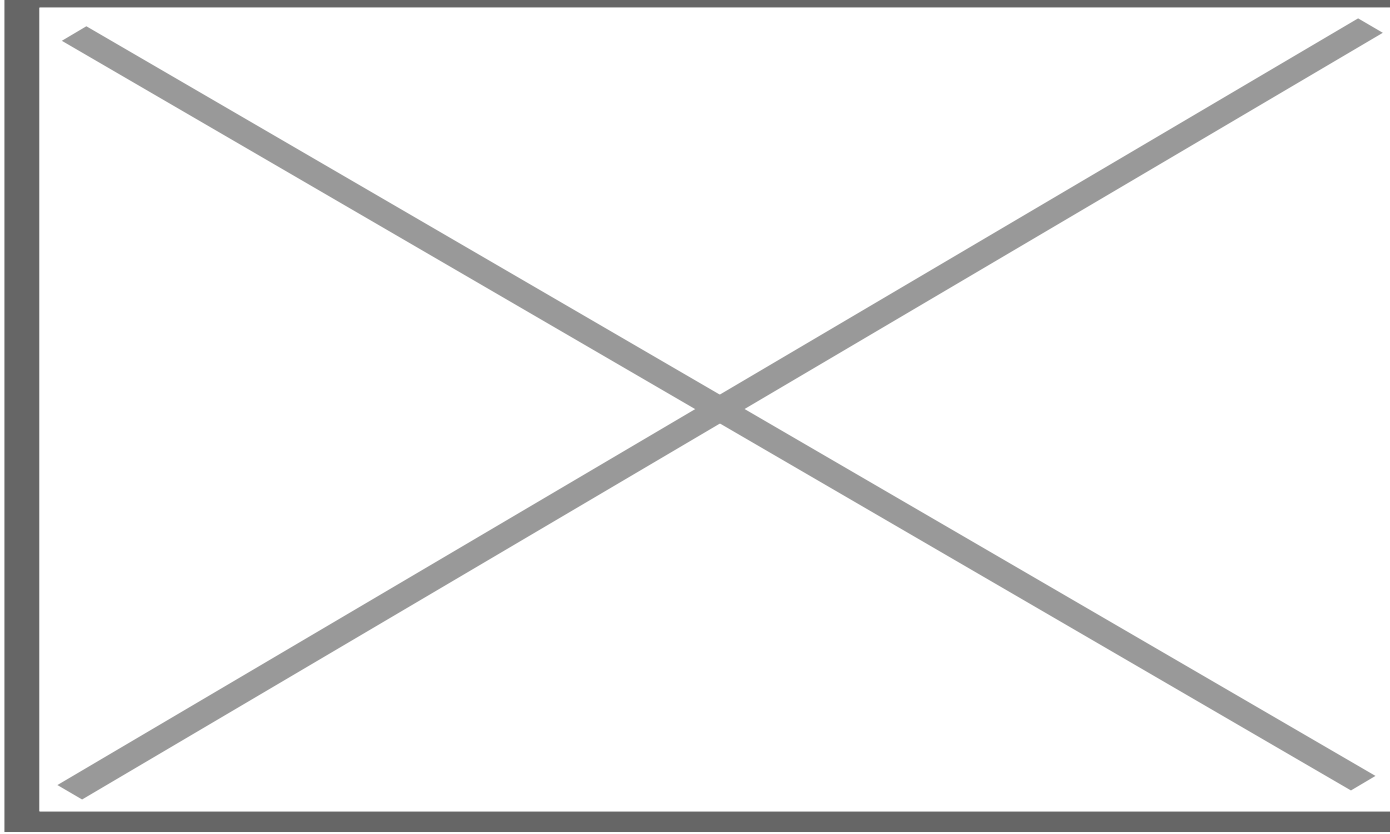


Le script permettant de lire les données uniquement pour cette zone est le suivant :

```
build {  
  //FlowNodeAttributes=""  
  //EditorComments="Lit un fichier Excel comme celui-ci :  
  //  
  //**  
  //"  
  excel-input "/data/objectifs vendeurs zones.xlsx" range="A1:C10"  
  name="excel-input_1"  
  output "/cbases/sortie-excel.cbase" name="output_2"  
  layout-data {  
    node name="excel-input_1" position="8 8"  
    node name="concat" position="158 8"  
    node name="output_2" position="238 8"  
  }  
}
```

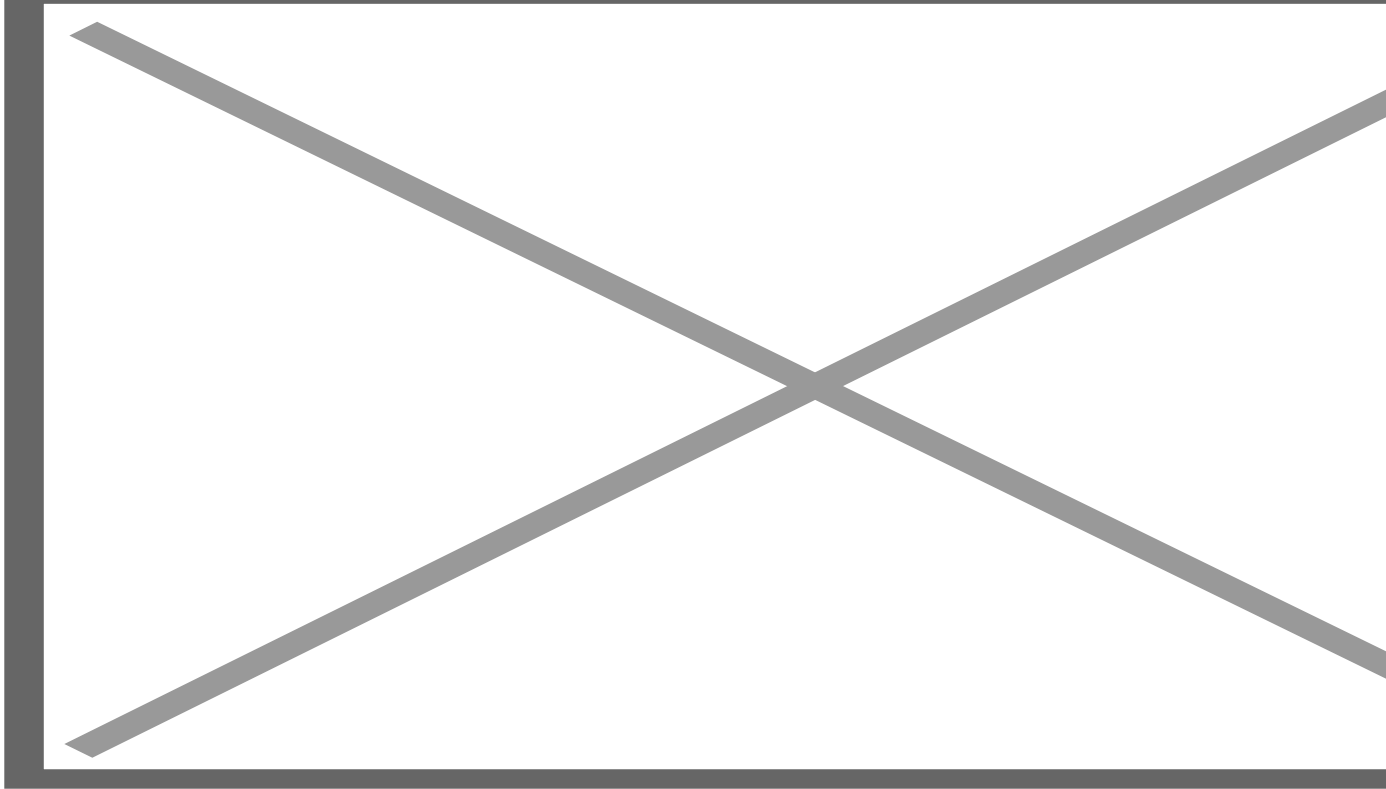
Et avec le résultat suivant dans le cbase une fois le script exécuté :

Image not found or type unknown



Supposons que l'on ait un fichier Excel contenant plusieurs feuilles / onglets :

Image not found or type unknown



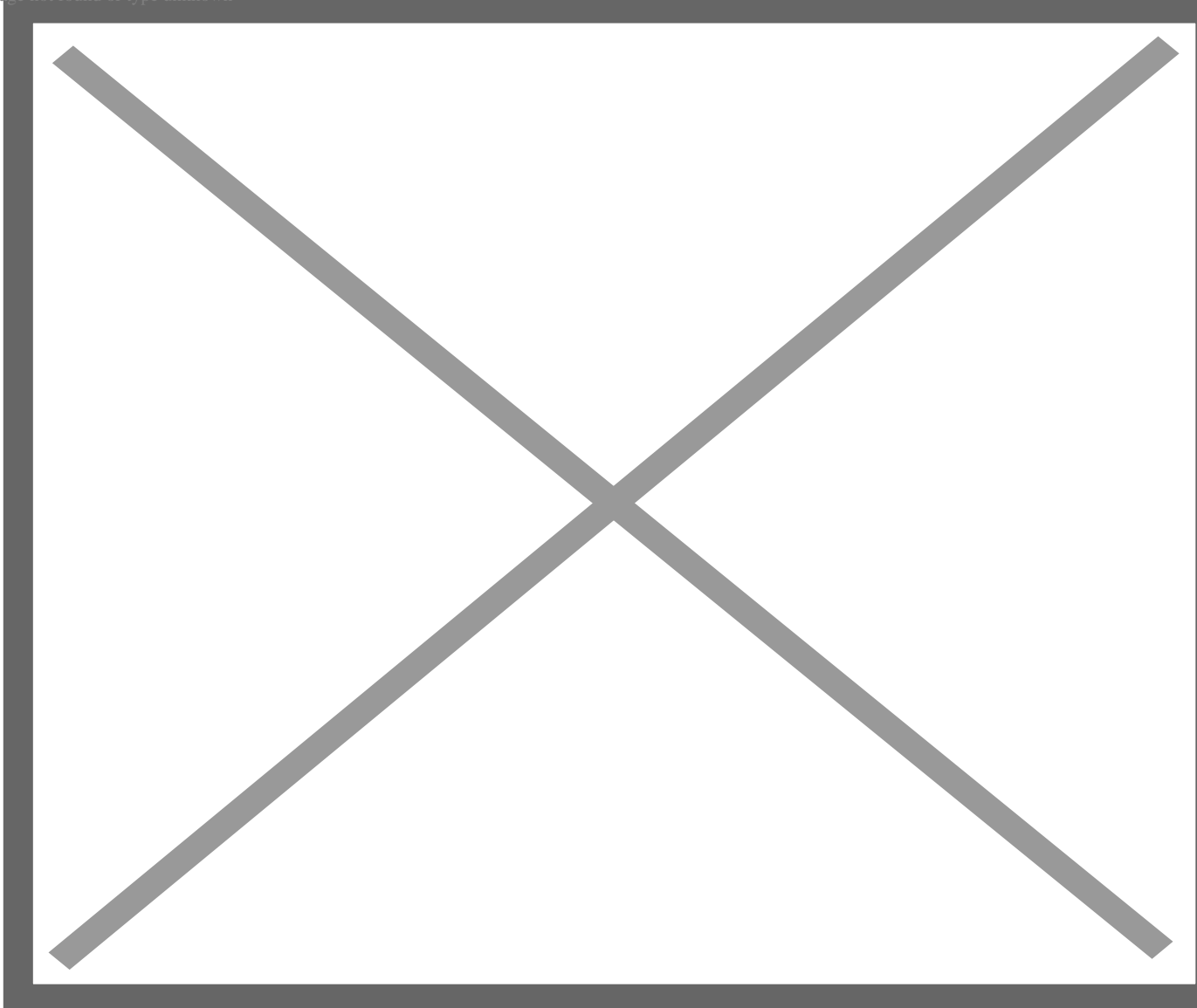
et que nous souhaitons récupérer les données d'un onglet spécifique, par exemple **Objectifs vendeurs**.

Le script permettant de lire les données de l'onglet souhaité est le suivant :

```
build {
  //FlowNodeAttributes=""
  //EditorComments="Lit un fichier Excel comme celui-ci :
  //
  //**
  //""
  excel-input "/data/objectifs multi feuilles.xlsx" sheet="Objectifs vendeurs"
name="excel-input_1"
  output "/cbases/sortie-excel.cbase" name="output_2"
  layout-data {
    node name="excel-input_1" position="8 8"
    node name="concat" position="88 8"
    node name="output_2" position="168 8"
  }
}
```

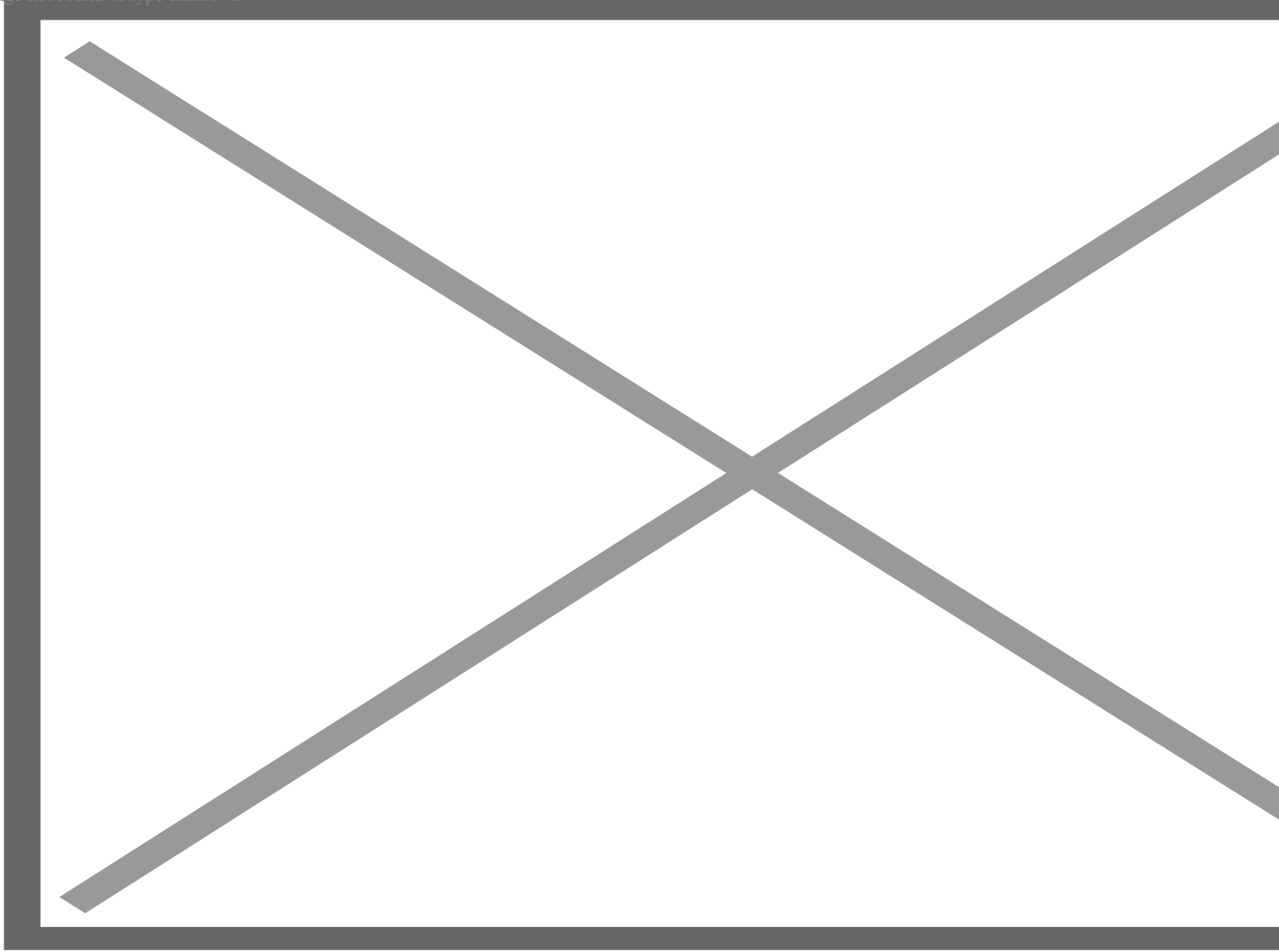
Si dans Workbench, on bascule sur l'éditeur de flux du script, afin d'avoir une interface graphique, en cliquant sur l'objet **Excel-input** on retrouve la propriété dans la zone encadrée :

Image not found or type unknown



Supposons que l'on ait maintenant le fichier Excel suivant :

Image not found or type unknown



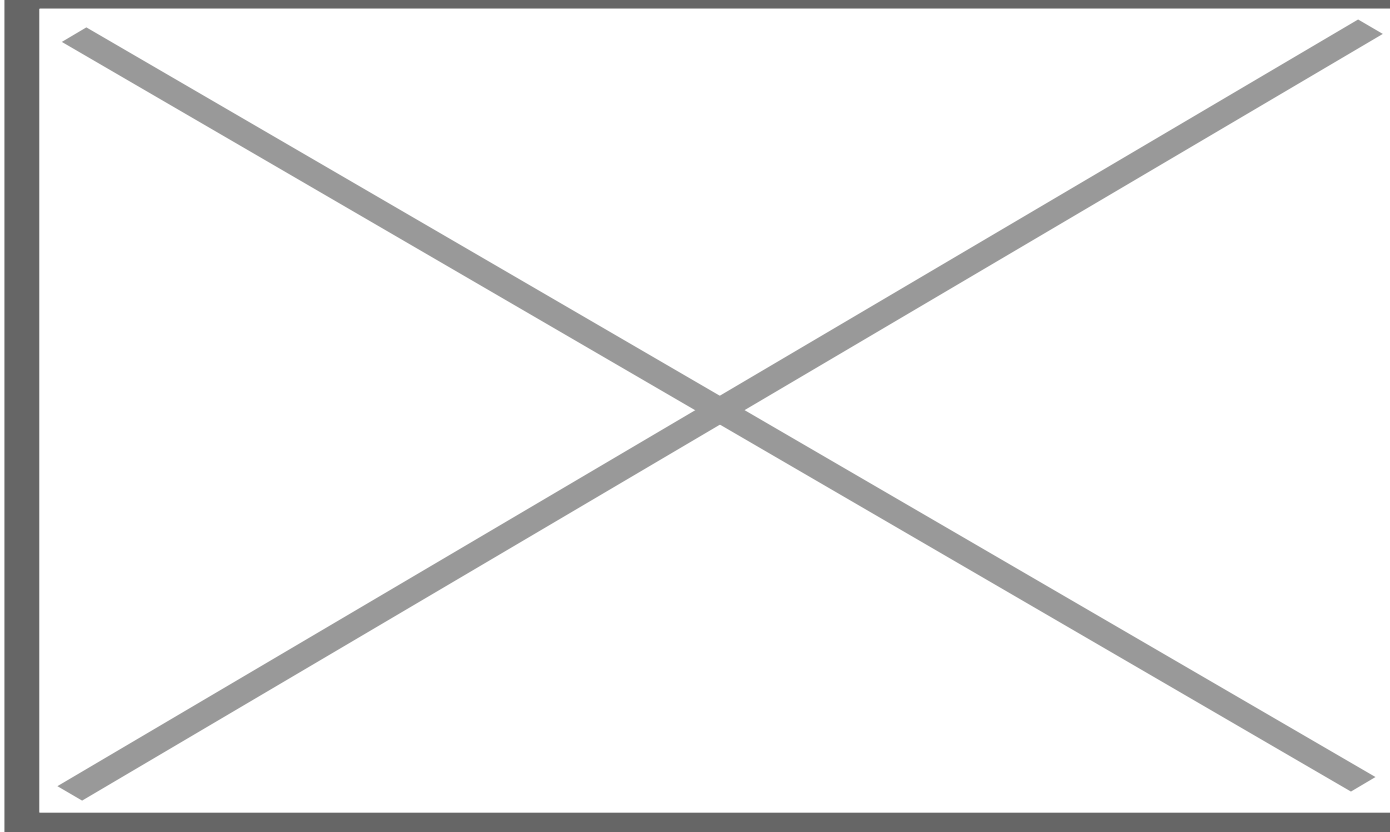
On s'aperçoit que les valeurs de **Zone** ne sont pas systématiquement répétées sur chaque ligne.

Le script permettant répéter les valeurs manquantes est le suivant :

```
build {
  //FlowNodeAttributes=""
  //EditorComments="Lit un fichier Excel comme celui-ci :
  //
  //**
  //""
  excel-input "/data/objectifs zones trimestres.xlsx" name="excel-input_1"{
    column "Zone" repeat-if-blank=true
    include-other-columns
  }
  output "/cbases/sortie-excel.cbase" name="output_2"
  layout-data {
    node name="excel-input_1" position="8 8"
    node name="concat" position="88 8"
    node name="output_2" position="168 8"
  }
}
```

Ci-dessous le résultat dans le cbase une fois le script exécuté :

Image not found or type unknown



Tags

1. script
2. Spectre