

# ODSelect : utilitaires pour la sélection par filtre d'object data

## Problématique

La sélection par valeur d'object data [est très lente en lisp](#), car il n'est pas possible d'ajouter ses critères à SSGET, il faut parcourir l'ensemble des entités, et chercher la table, puis éventuellement la valeur.

On trouve [ici \(message #38\) MQselect](#) une application très intéressante pour sélectionner selon des critères précis.

Il manquait encore une version lisp, afin de pouvoir combler les lacunes de l'API lisp pour Map.

## Télécharger et utiliser

Il faut télécharger Odeslect.zip, le décompresser, et le mettre où vous voulez.

On peut automatiser le chargement des dll avec plein de techniques différentes, voir le lien [Chargement automatique des DLLs .NET](#)

Une fois chargé Odselec.dll via \_netload, vous aurez les messages suivant :

*Object Data Selection Application Initialized.*

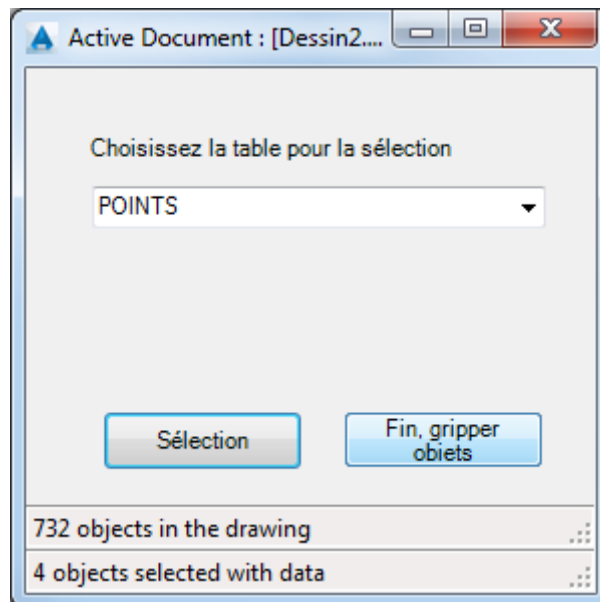
*Use ODSelect command at Autocad prompt*

*Lisp command SelParTablelsp : (SelParTablelsp 'TABLE' ) -> (<Selection set: 74>)*

*Lisp command SelParODValLsp : (SelParODValLsp 'TABLE' 'CHAMP' myVal [precision or caseSensitive]) -> (<Selection set: 74>)*

## La commande ODselect

Affiche une case de dialogue, et permet de visualiser tour à tour les éléments de chaque table, puis de gripper le dernier jeu de sélection, afin éventuellement d'en modifier les propriétés.



## ***Les commandes lisp SelParTableLsp et SelParODValLsp***

### **Sélection par nom de table**

```
(SelParTableLsp "E-TRONCO" ) -> (<Selection set: 74>)
```

Exclusivement sensible à la casse, aucun joker accepté

### **Sélection par nom de table, champs et valeur**

Le champ diamètre est de type entier :

```
(SelParODValLsp "E-TRONCO" "DIAMETRE" "100") -> nil
(SelParODValLsp "E-TRONCO" "DIAMETRE" 100) -> (<Selection set: 74>)
```

Le champ Zaval est de type réel, le dessin contient des valeurs de 1.05 à 1.09

```
(setq sel (car (SelParODValLsp "E-TRONCO" "ZAVAl" 1.0))) -> nil
```

On ajoute l'argument précision :

```
(setq sel (car (SelParODValLsp "E-TRONCO" "ZAVAl" 1.0 0.1))) -> (<Selection set: 74>)
```

Le champ RESEAU est de type chaîne, le dessin contient la valeur « TRANPOT »

```
(setq sel (car (SelParODValLsp "E-TRONCO" "RESEAU" "TRANPOT"))) -> (<Selection set: 74>)
```

Avec un joker (fonctionnement identique à la fonction LIKE, \*, ? et # sont valides :

```
(setq sel (car (SelParODValLsp "E-TRONCO" "RESEAU" "Tranpo*"))) -> (<Selection set: 74>)
```

On ajoute l'argument CaseSensitive (T ou nil) :

```
(setq sel (car (SelParODValLsp "E-TRONCO" "RESEAU" "Tranpot" t))) -> nil
```

Le champ POINT est de type point, le dessin contient des valeurs de type 2.05,2.01,2.09 :

```
(setq sel (car (SelParODValLsp "POINTS" "POINT" '(2.0 2.0 2.0)))) -> nil
```

On ajoute l'argument précision :

```
(setq sel (car (SelParODValLsp "POINTS" "POINT" '(2.0 2.0 2.0) 0.1))) ->  
<Selection set: 74>
```

[g-eaux](#)