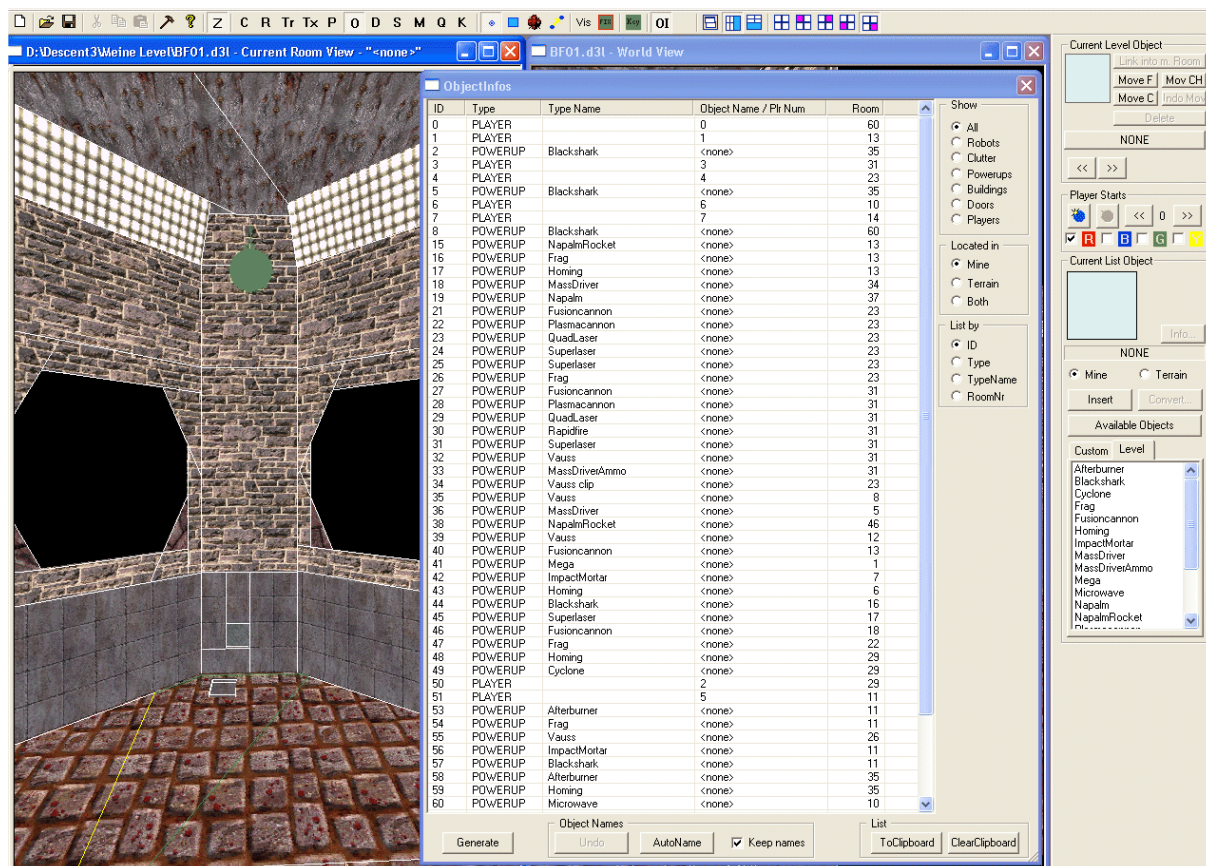


Dialogue d'Infos des Objets (OI) (v1.0) ATAN

Le dialogue OI est un outil utile pour organiser les objets d'un niveau.
Il l'est beaucoup pour les niveaux Solo, mais il peut aussi l'être pour les niveaux Multijoueurs.
Il est plus facile de voir quels objets ont été placés, et de quelle sorte d'objet il s'agit.
Pour voir à quoi le dialogue ressemble, ouvrons un petit niveau Multijoueurs.

Appuyer sur le bouton OI dans la barre des tâches :

OI



Le dialogue OI s'ouvre et nous montre une quantité d'informations.

La fenêtre de Sortie contient les colonnes suivantes :

ID	Type	Type Name	Object Name / Plr Num	Room
----	------	-----------	-----------------------	------

Show

☒ All

☐ Robots

☐ Clutter

☐ Powerups

☐ Buildings

☐ Doors

☐ Players

Located in

☒ Mine

☐ Terrain

☐ Both

List by

☒ ID

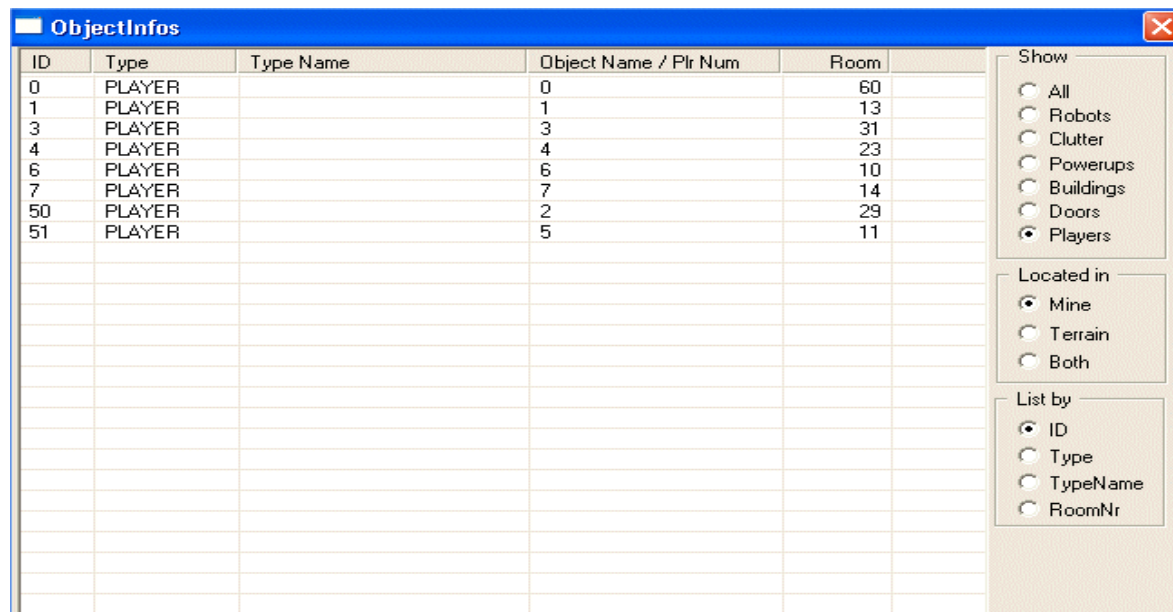
☐ Type

☐ TypeName

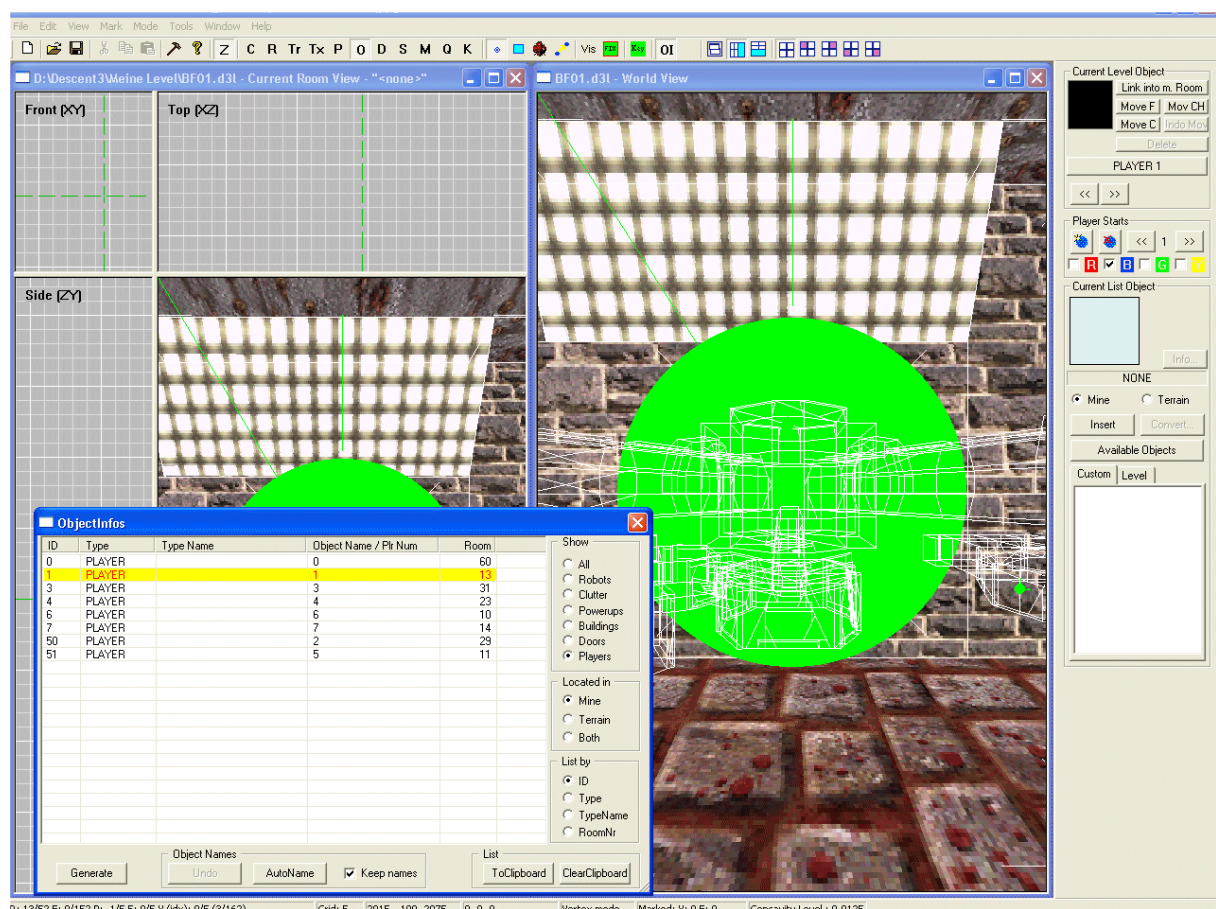
☐ RoomNr

On peut sortir les informations sur les objets de plusieurs façons dans cette fenêtre. Dans l'exemple ci-dessus, les objets de la mine sont listés par leur ID.

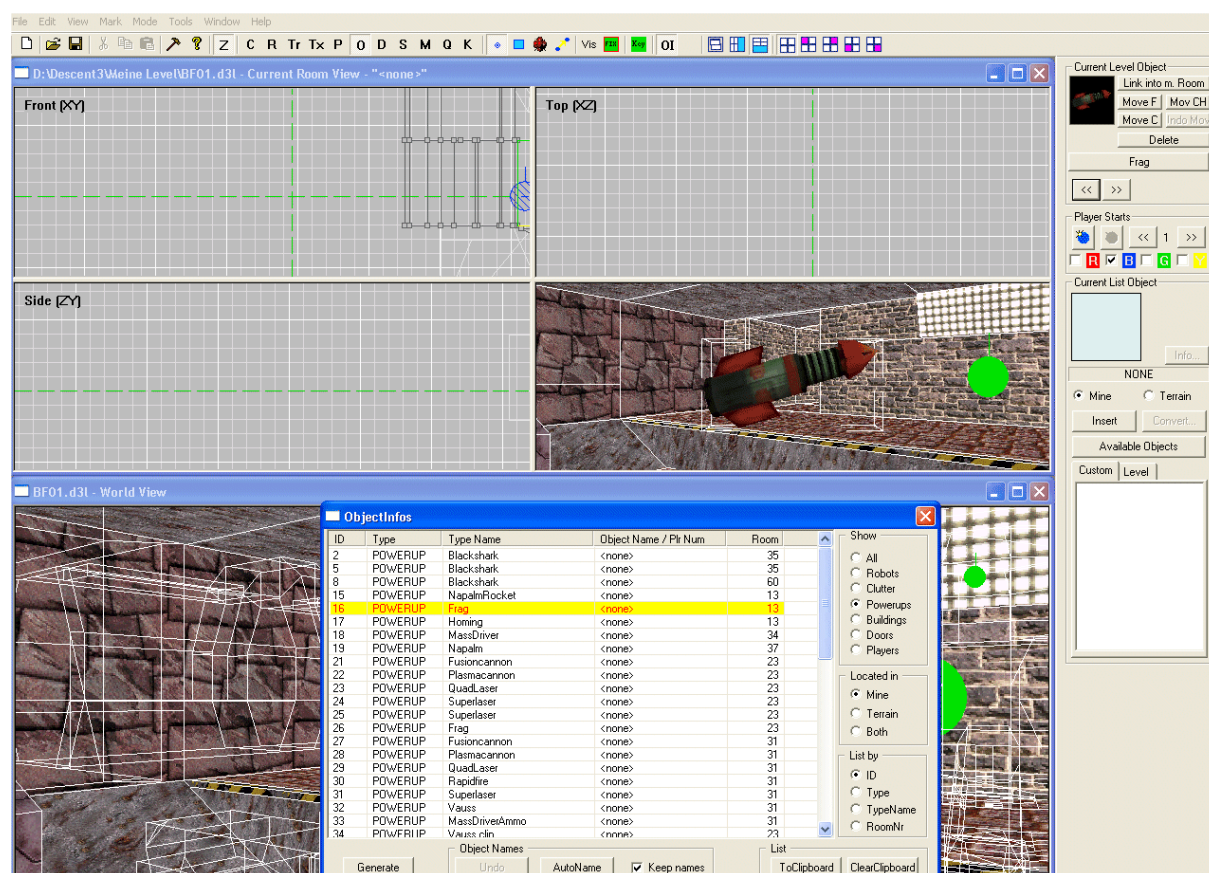
Pour jeter un coup d'œil sur le nombre de PlayerStart insérés dans le niveau, cocher *Players*.



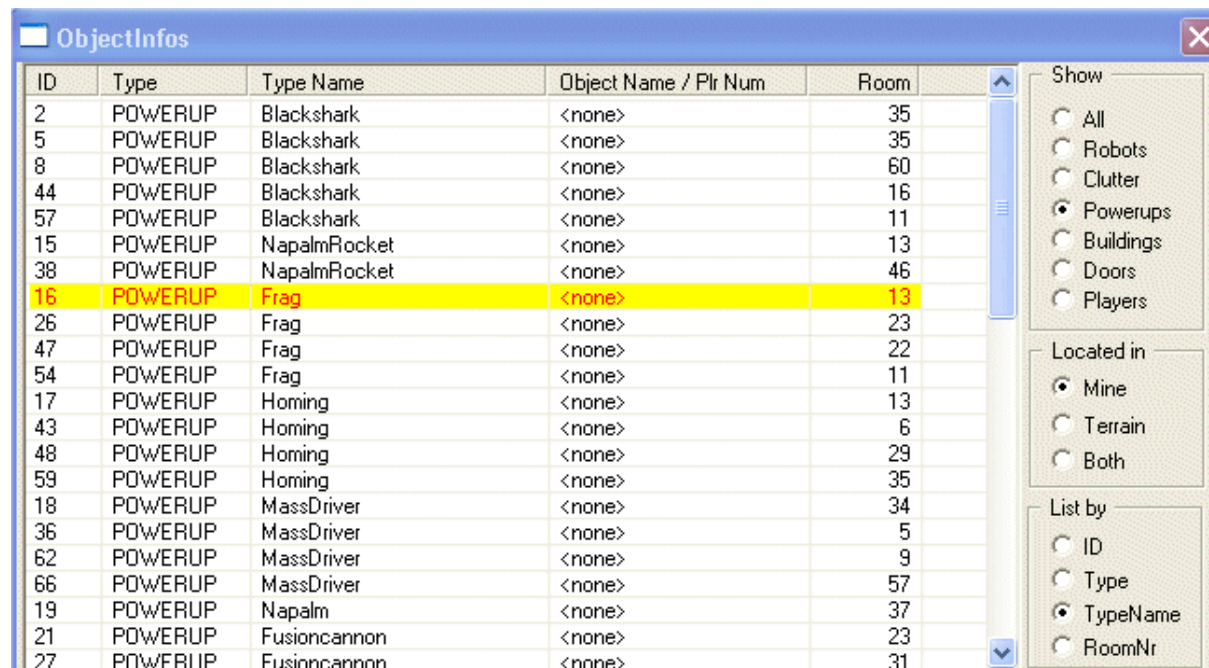
Le Dialogue OI ne nous montre maintenant plus que les Objets Joueurs (Startpoints). Ils sont sortis par ID, c'est à dire selon l'Index que le D3Edit leur a attribué dans le niveau. Il peut être plus intéressant de les sortir par RoomNr pour pouvoir vérifier dans quelles pièces ils sont. Pour un objet précis, on peut cliquer gauche sur la ligne. Le D3Edit va focaliser la vue sur cet objet dans la Mine/Pièce et la ligne va se colorier dans le dialogue OI.



Par exemple, on sélectionne les Powerups et l'objet n° 16.
Le D3Edit passe dans la pièce qui contient ce powerup et zoome sur le missile Frag.
La barre Object change ausse. Elle nous montre alors l'objet courant (le Frag).

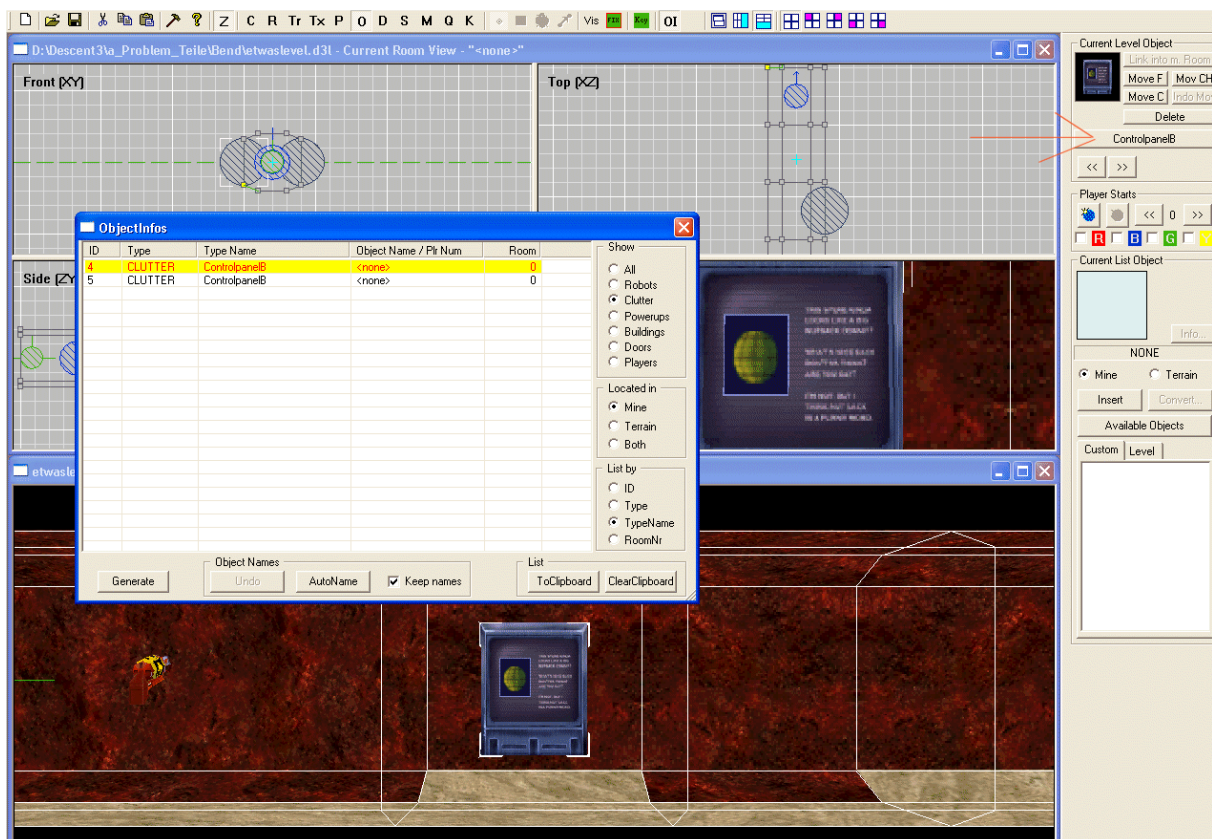


Pour voir combien de Frag on a inséré dans la mine,
on doit lister les objets par *TypeName* :



De cette façon, on voit combien il y a de Frag et où ils ont été insérés.
Un clic de souris sélectionnera un de ces objets et nous amènera jusqu'à lui.

Si on veut utiliser un objet dans le DALLAS, cet objet doit avoir un nom.
 Pour nommer un objet, appuyer sur ce bouton dans la barre Object (flèche rouge).
 Quand on appuie sur ce bouton, une fenêtre d'édition s'ouvre pour y qu'on y écrive un nom.



On peut aussi cliquer gauche dans la cellule de la colonne - *Object Name* - .
 De cette façon, on obtient un nouveau champ de texte (flèche rouge).

ObjectInfos				
ID	Type	Type Name	Object Name / Plr Num	Room
4	CLUTTER	ControlpanelB	<none>	0
5	CLUTTER	ControlpanelB	<none>	0

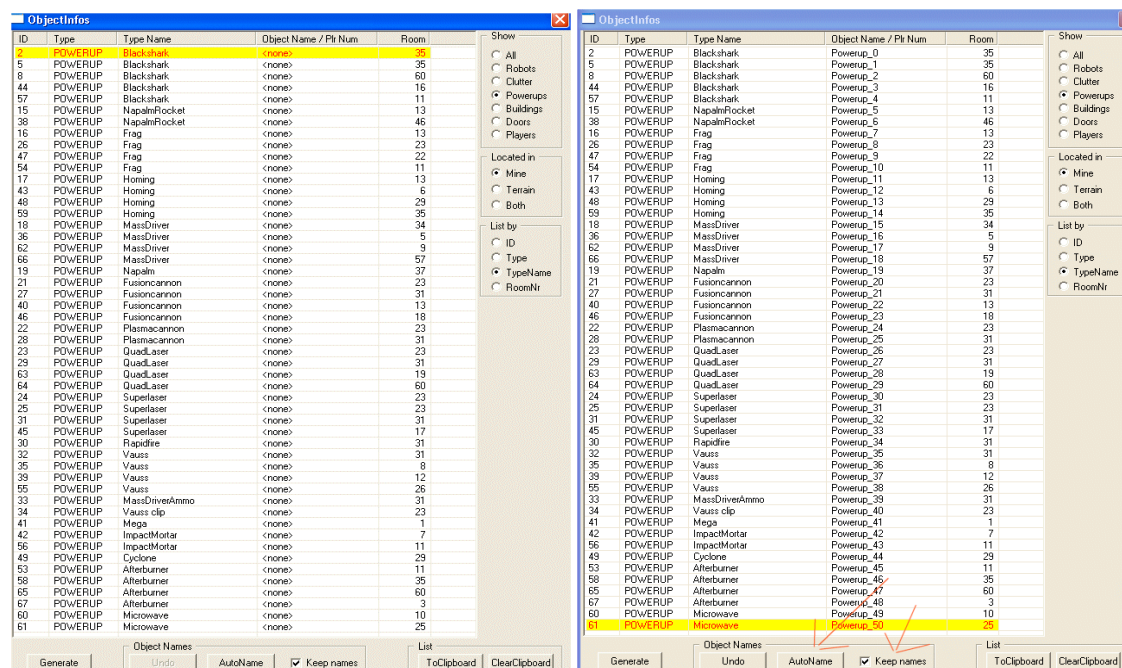
Écrire un nouveau nom d'objet et appuyer sur Entrée.

ObjectInfos				
ID	Type	Type Name	Object Name / Plr Num	Room
4	CLUTTER	ControlpanelB	Überwachungspanel_1	0
5	CLUTTER	ControlpanelB	<none>	0



Ce nom sera apparent aussi dans la barre Object.
 Le fait de changer le nom dans la barre Object rafraîchit le nom dans le Dialogue OI également.

Le Dialogue OI propose une fonction pour attribuer des noms auto-générés.
Quand on regarde la liste des Powerups du niveau, on voit :



Par défaut, tous les noms sont présélectionnés sur *< none >*, c'est à dire qu'ils n'ont pas de noms.
En appuyant sur le bouton *-Auto Name-*, on va numéroter automatiquement tous les objets du même type. Au cas où des objets de ce type auraient déjà un nom et qu'on voudrait garder ce nom, il suffit d'appuyer sur *-Keep Names-*.
-Auto Name- ne changera alors pas leur nom.
Il y a un bouton *-Undo-* au cas où on le résultat ne nous plaît pas ou que quelque chose a mal tourné.

-ToClipboard – envoie la liste courante dans le presse papiers.
-ClearClipboard – efface le contenu du presse papiers.

Le bouton *-Generate-* présente une fonctionnalité très particulière.
Les scripteurs du DALAS expérimentés vont certainement l'apprécier. Le résultat en est copié dans le presse papier.

Voici par exemple, sur la page suivante, la liste générée par cette fonction et que j'ai utilisée dans le script de *Mission to Saturn Part II*.

<pre> /* \$ENUM MyRobots 0:"B_P2Gr2" 1:"B_P3Gr2" 2:"B1" 3:"B2" 4:"B3" 5:"Bot_32" 6:"Bot_33" 7:"Bot_34" 8:"Bot_35" 9:"B_P2Gr1" 10:"B4" 11:"KameraDoorOpen" 12:"Bot_51" 13:"Bot_52" 14:"Bot_55" 15:"Bot_56" 16:"B5" 17:"B6" 18:"B_P1Gr2" 19:"Bot_53" 20:"Bot_54" 21:"Bot_43" 22:"Bot_44" 23:"Bot_45" 24:"Bot_59" 25:"Bot_46" 26:"Bot_60" 27:"Bot_61" 28:"Bot_62" 29:"Bot_63" 30:"Bot_64" 31:"Bot_65" 32:"Bot_66" 33:"Bot_67" 34:"Bot_68" 35:"Bot_69" 36:"B8" 37:"B9" 38:"B10" 39:"B_P3Gr5" 40:"B_P1Gr5" 41:"B_P2Gr5" 42:"B_P1Gr4" 43:"B_P2Gr4" 44:"B_P3Gr4" 45:"B_P3Gr6" 46:"B19" 47:"B20" 48:"B21" 49:"B22" 50:"B23" 51:"B24" 52:"B25" 53:"B26" 54:"B27" 55:"B28" 56:"Bot_70" 57:"Schweisser1" 58:"Bot_36" 59:"Bot_37" 60:"Bot_71" 61:"B29" 62:"Bot_38" 63:"Bot_85" 64:"Bot_86" 65:"B11" 66:"B12" 67:"B13" 68:"B14" 69:"B15" 70:"B16" 71:"B17" 72:"B18" 73:"Bot_87" 74:"Bot_88" 75:"Bot_47" 76:"Bot_48" 77:"Bot_49" 78:"Bot_72" 79:"Cam-Teleport1" 80:"B_P1Gr6" 81:"B_P2Gr6" 82:"B_P3Gr1" 83:"B_P1Gr3" 84:"B_P2Gr3" 85:"B_P3Gr3" 86:"B_P1Gr1" 87:"B7" 88:"CrashPlage" \$END */ </pre>	<pre> #define NUM_BOT_OBJECT_NAMES 89 char *Bot_Object_names[NUM_BOT_OBJECT_NAMES] = { "B_P2Gr2", "B_P3Gr2", "B1", "B2", "B3", "Bot_32", "Bot_33", "Bot_34", "Bot_35", "B_P2Gr1", "B4", "KameraDoorOpen", "Bot_51", "Bot_52", "Bot_55", "Bot_56", "B5", "B6", "B_P1Gr2", "Bot_53", "Bot_54", "Bot_43", "Bot_44", "Bot_45", "Bot_59", "Bot_46", "Bot_60", "Bot_61", "Bot_62", "Bot_63", "Bot_64", "Bot_65", "Bot_66", "Bot_67", "Bot_68", "Bot_69", "B8", "B9", "B10", "B_P3Gr5", "B_P1Gr5", "B_P2Gr5", "B_P1Gr4", "B_P2Gr4", "B_P3Gr4", "B_P3Gr6", "B19", "B20", "B21", "B22", "B23", "B24", "B25", "B26", "B27", "B28", "Bot_70", "Schweisser1", "Bot_36", "Bot_37", "Bot_71", "B29", "Bot_38", "Bot_85", "Bot_86", "B11", "B12", "B13", "B14", "B15", "B16", "B17", "B18", "Bot_87", "Bot_88", "Bot_47", "Bot_48", "Bot_49", "Bot_72", "Cam-Teleport1", "B_P1Gr6", "B_P2Gr6", "B_P3Gr1", "B_P1Gr3", "B_P2Gr3", "B_P3Gr3", "B_P1Gr1", "B7", "CrashPlage", }; </pre>
--	--