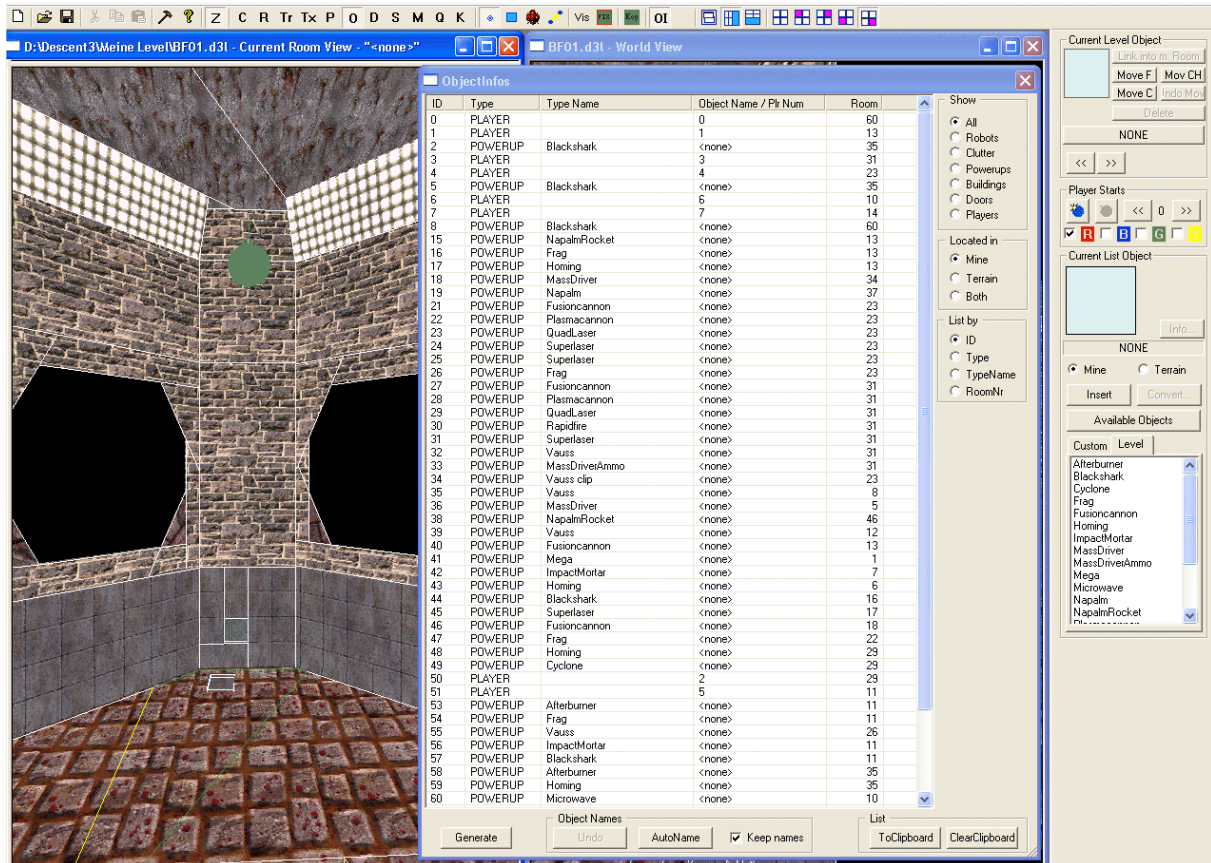


## Object Info Dialog (OI) (v1.0)

OI Dialog is an useful Tool for organizing Level Objects.  
While it's mainly useful for SP Levels, it might be useful for MP levels too.  
Easy task to see which and how many objects are placed.  
To see how it looks like we load a small MP Level first.

Hit the *OI Button* inside the Tool Bar then:



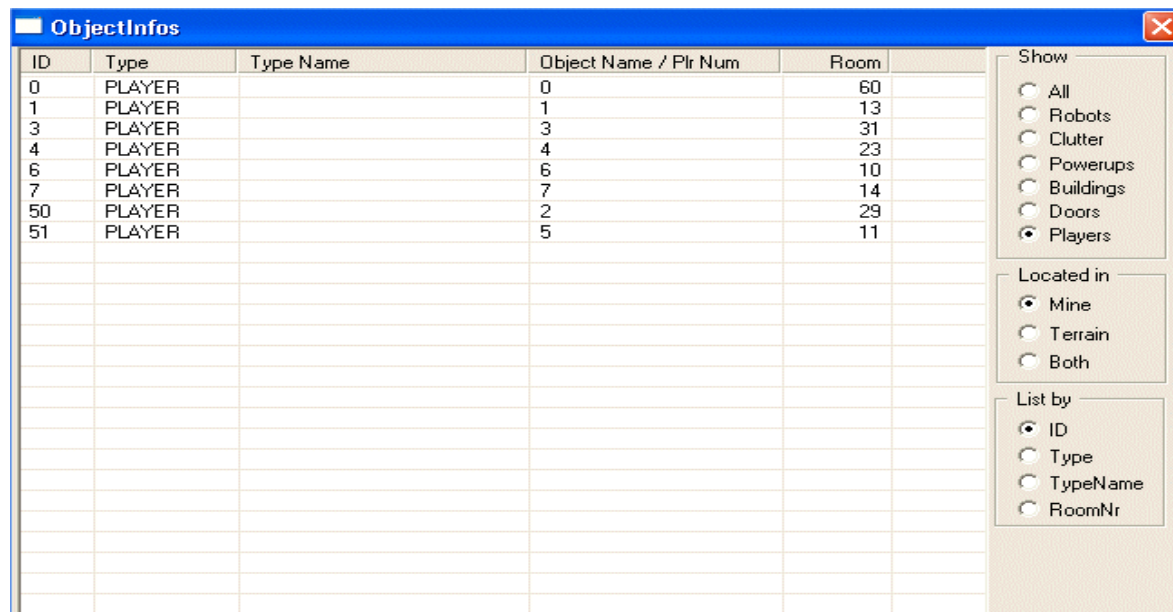
The OI-Dialog opens with lots of informations for us.  
Inside the output window we see this rows here:

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

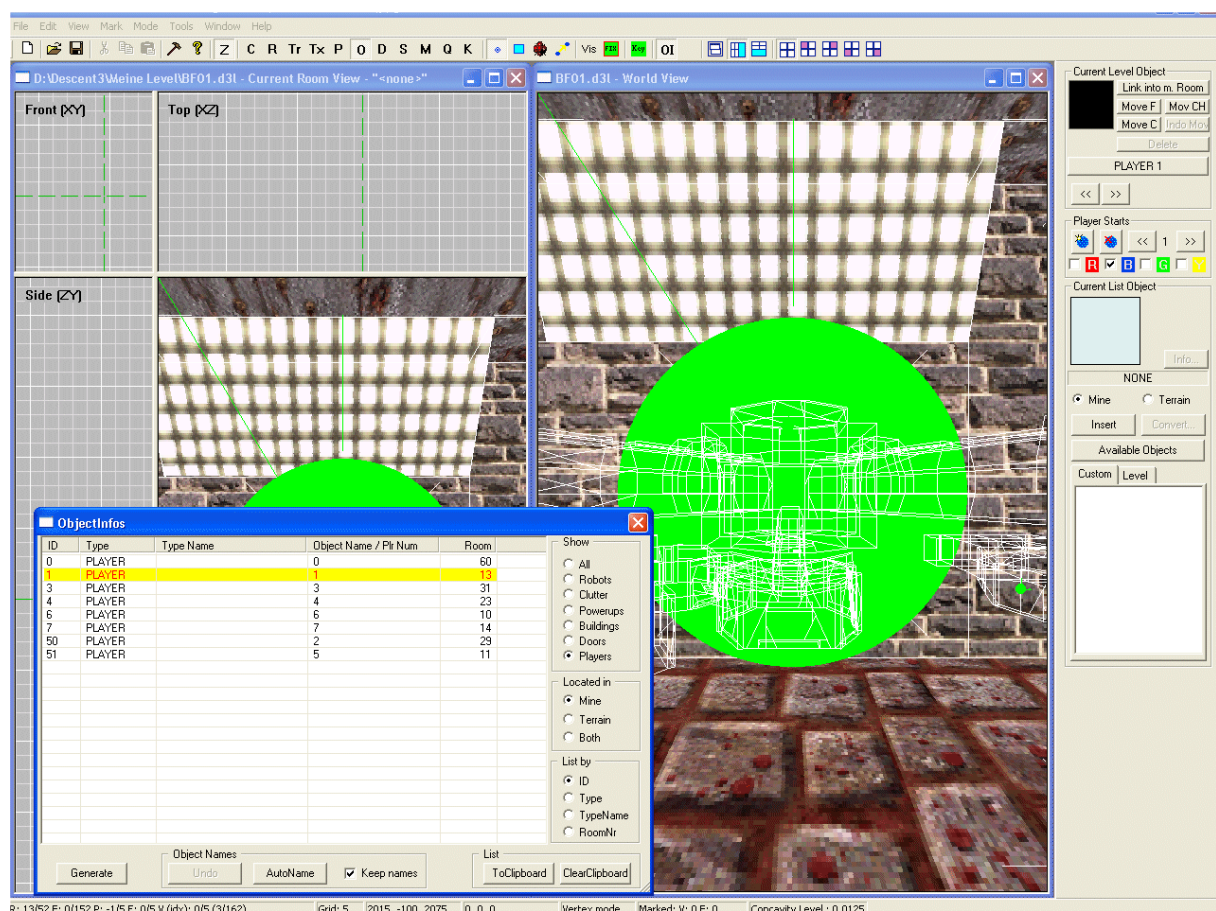


There are several ways to show the Object-informations inside this window. In the example above all Objects inside the Mine are listed by their ID.

To have a quick look on how many playerstart points are inserted inside this level tick *Players*.

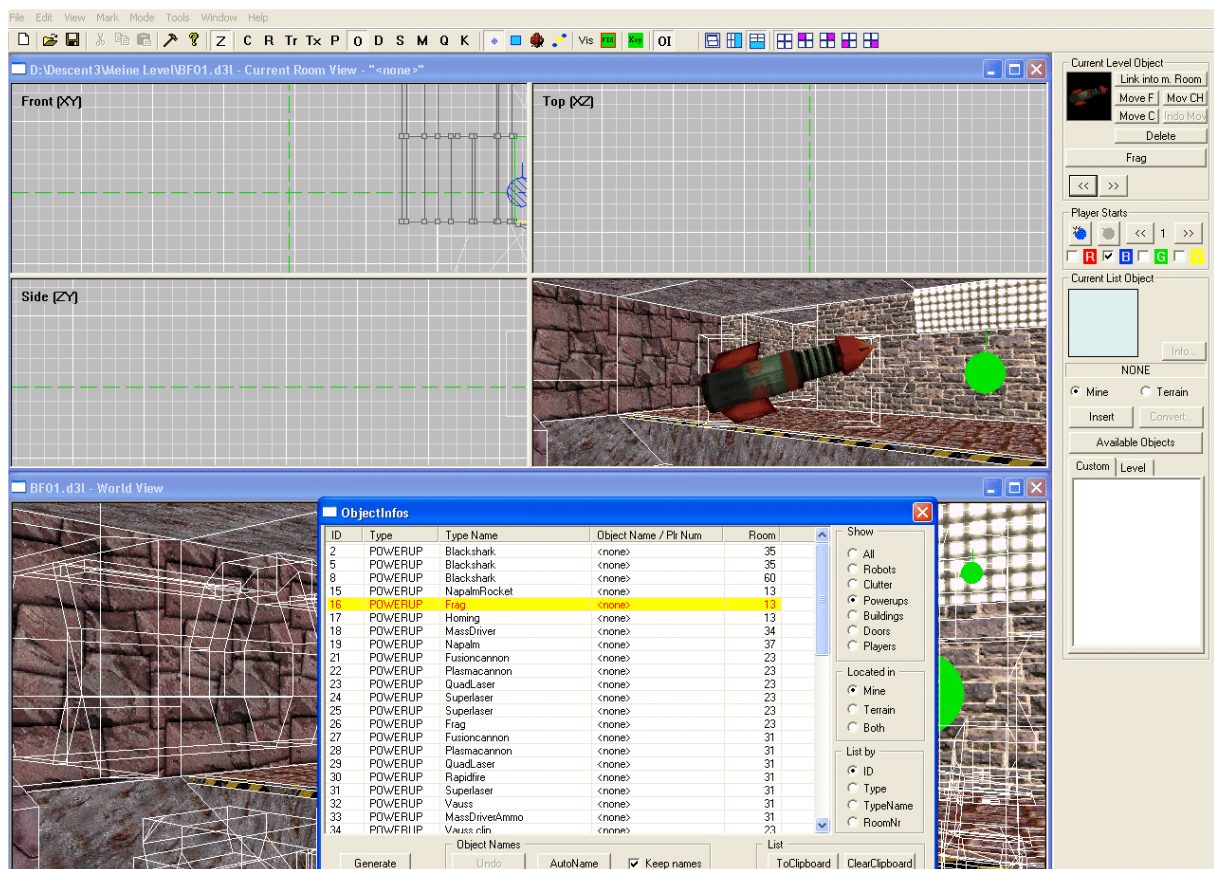


Now the OI Dialog shows the Player Objects (Startpoints) only.  
 They are shown by ID, means by which Index D3Edit stored the Object into the Level.  
 More interesting would be to check RoomNr to see inside which Room each object is.  
 If we are interested into one object we may left click on the line. D3Edit will move to that object inside the Mine/Room and the selected line inside OI Dialog will be colored.

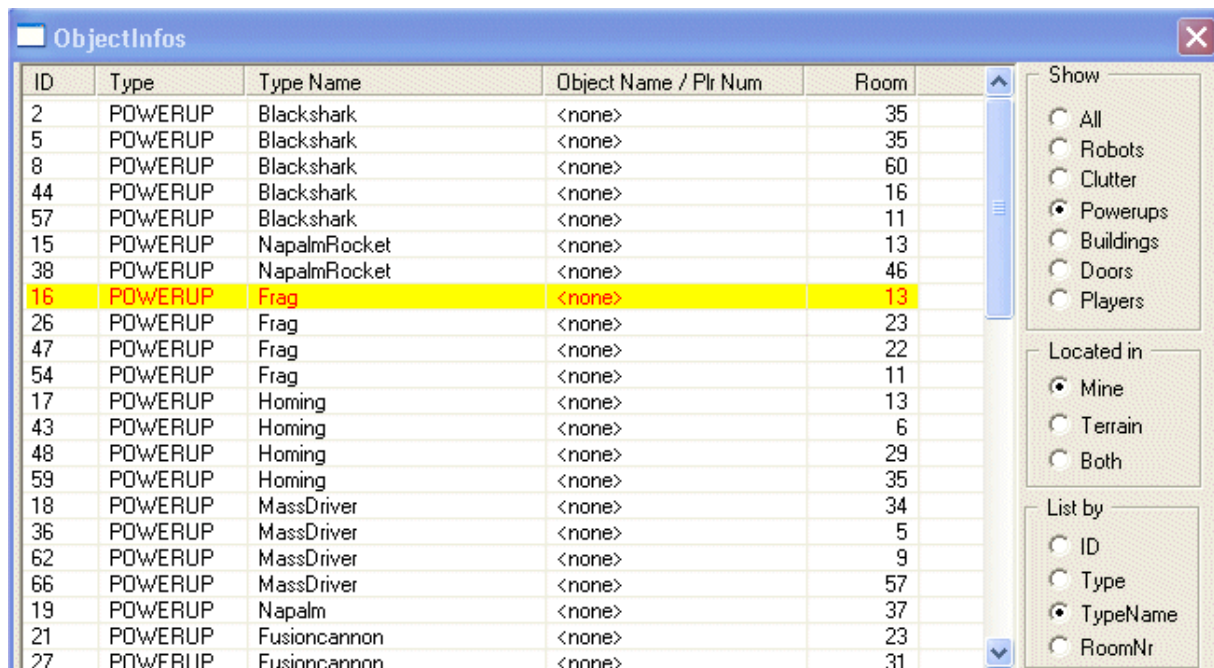




For example we choose Powerups now and select Object Nr 16.  
D3Edit warps inside the Room containing that powerup and is zooming towards the Frag.  
The Object Bar changes too. The current Object (Frag) is shown there now.

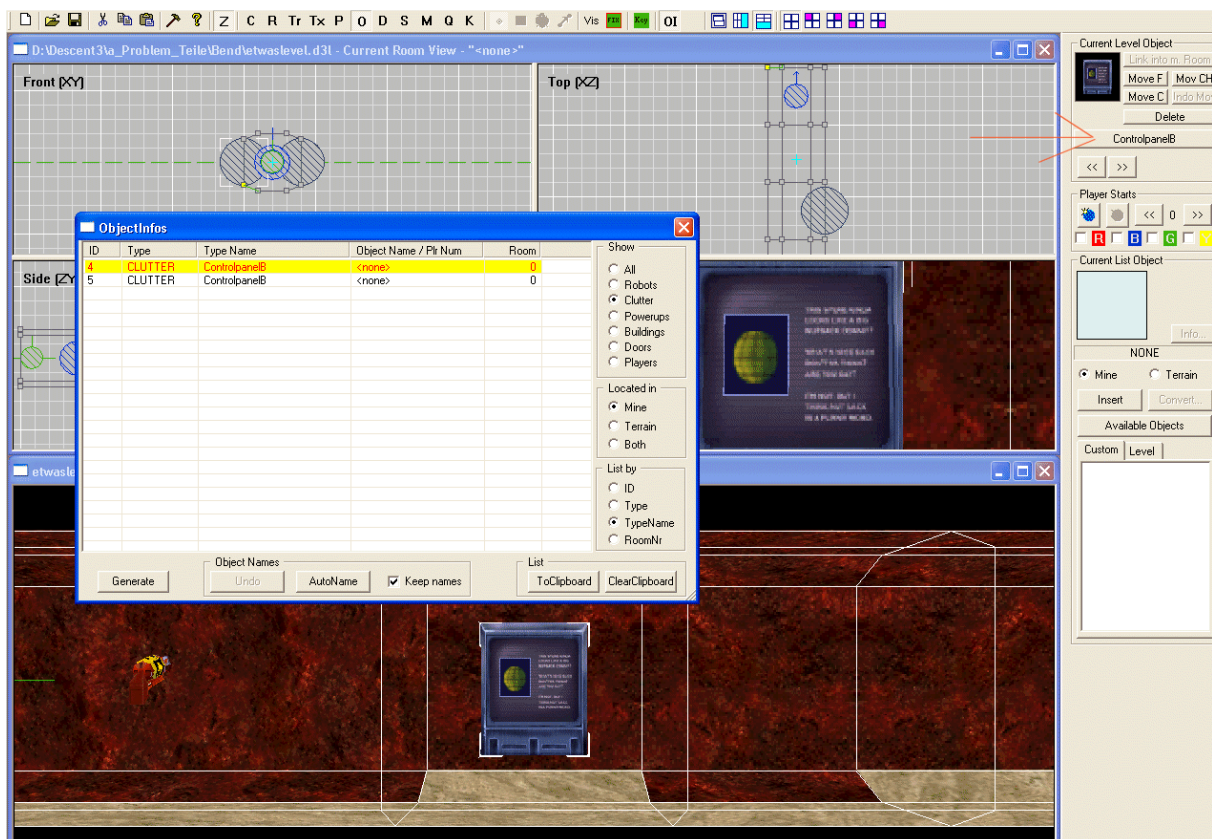


If we want to see how many Frags we inserted into the mine,  
we have to list the Objects by *TypeName* :



This way we can find out how many Frags are inserted and where.  
A mouse click will select an object and move us towards that this way selected object.

If you want to handle an object inside DALLAS you need to name that object.  
 To do that hit the button inside the Object Bar ( see red Arrow ).  
 If you press that button you will see an EditBox to insert a name.



But it's possible too to left click into the cell which belong to - *Object Name* - .  
 This way we will get a new EditField (red arrow).

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

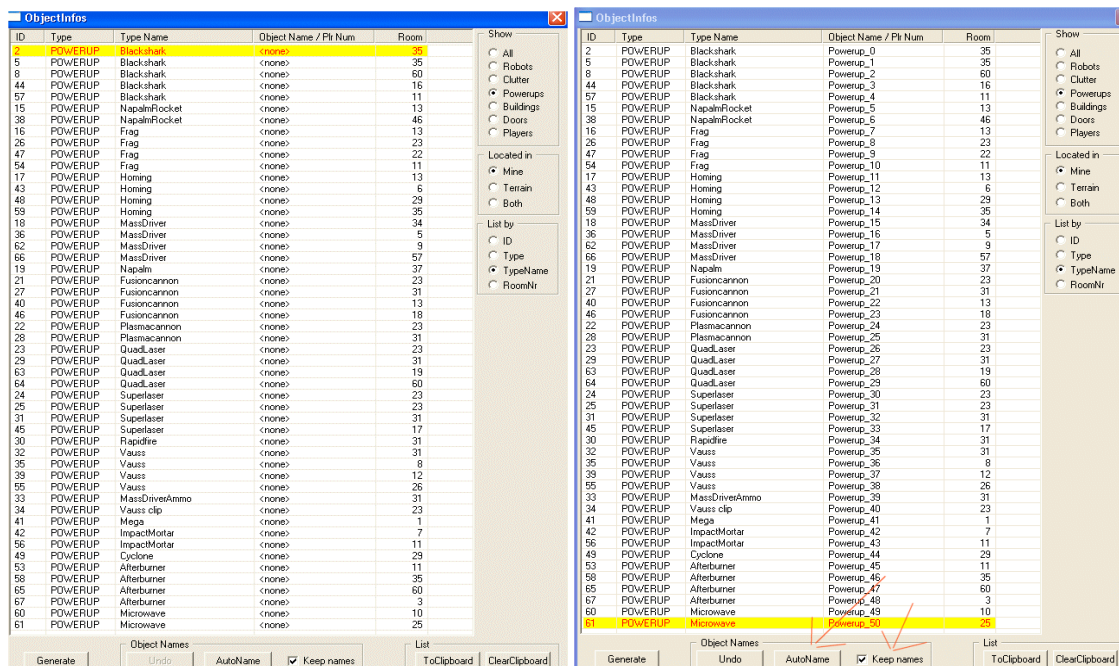
Insert a new Object Name followed by Return.

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



This name is shown inside Object Bar too.  
 Changing a name inside the Object Bar will refresh the name  
 inside the OI dialog too.

OI dialog offers a function to change the name by self generated names.  
If we look into this level Powerup list we'll see:



By default all names are preset by `< none >` means they have no name.  
If we hit the `-Auto Name-` button all objects belonging to the same type are automatically numbered.  
Would these objects be named before and you want to keep those names, use `-Keep Names-`.  
`-Auto Name-` will not change them.  
There is a `-Undo-` button if you don't like the result or something went wrong.

`-ToClipboard` – will send the current list into the clipboard.  
`-ClearClipboard` – will delete clipboards contents.

A very special feature is behind the `-Generate-` button.  
Advanced DALLAS scripter will possibly like this one. The result will be send to the clipboard.

The following page will show for example the such way generated list I used inside the *Mission to Saturn Part II* Script.

<pre> /* \$SENUM 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" \$SEND */ </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>
--	--