

---

# oEd Explorer

version 1.0 – September 2002 - ©Osмосе Editeur

---

*oEd Explorer* is a component that enables the developer to list all his/her methods sorted by Insider's groups and components. Every method of the database is put in the list, including object methods. *oEd Explorer* can not work without the plug-in DynamicStructure.

*oEd Explorer* is free for owners of a « developer » license of DynamicStructure.

Developers who want to use it without purchasing DynamicStructure can get the component with a special licence and a low price (contact [www.osmosе.net](http://www.osmosе.net) or your local distributor)

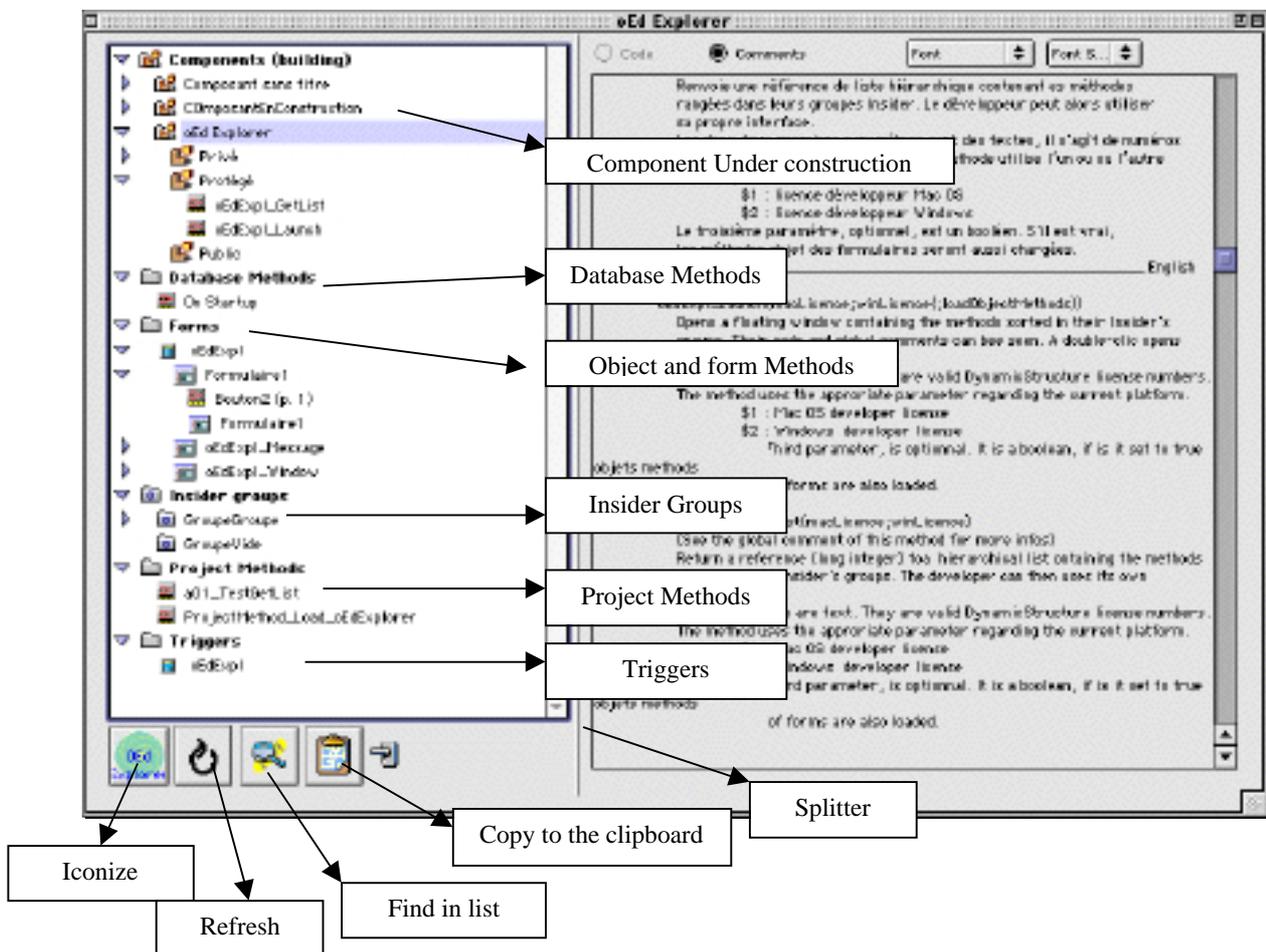
At last, *oEd Explorer* can be tested by using this special serial number : « oEx\*DEMO ». With this special licence, *oEd explorer* runs fully fonctionnal but display sometimes a reminder telling that it is in demo mode.

*oEd Explorer* contains 2 routines. One launches an explorer in a floating window, the other one has no interface and returns a reference to a hierarchical list containing the methods.

*Notice : oEd Explorer works only in interpreted environnement.*

## I. With interface : oEdExpl\_Launch

The method oEdExpl\_Launch opens a floating window showing the methods in their groups :



To launch this Explorer, you call the protected method « OedExpl\_launch ».

oEdExpl\_launch expects at least 2 parameters parameters :

\$1 and \$2 are alpha/text. These parameters are the license numbers for a « developer » version of DynamicStructure, or the serial number of *oEd Explorer*. \$1 is the Mac OS platform license number; \$2 is the Windows one. Pass "" for the license you don't own. To test the component, use this licence number : « oEx\*DEMO ».

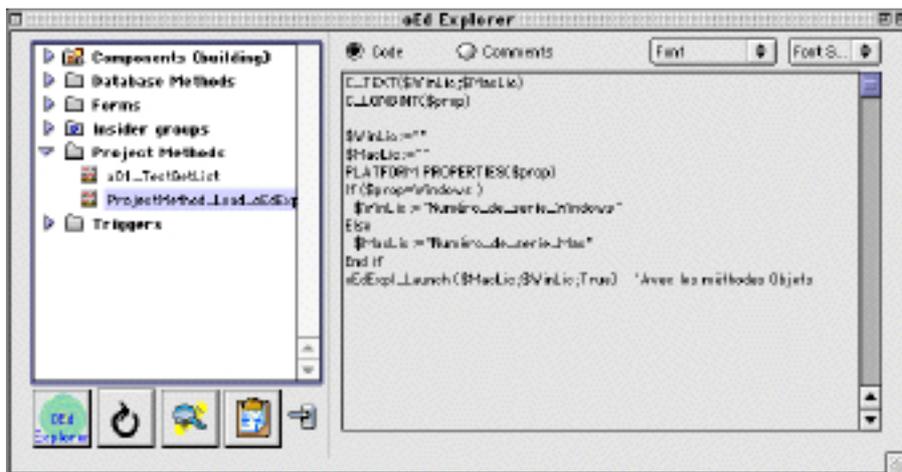
\$3 is a boolean (optionnal). If it is true, all the objects methods are also loaded is the list, they are put in a sublist whose name is the name of the form in the "Forms" group.

Example :

```
oEdExpl_Launch ($MacLic;$WinLic) `Without Object Methods
oEdExpl_Launch ($MacLic;$WinLic;True) `With Object Methods
```

Notice: - if you have a huge number of methods, filling the list may take a few seconds. A message indicates you have to wait.

- To see the code of a method , select it (single click) and choose the «Code» radio button.



- To open a method in Design mode, make a double-click on its name.
- To see the comment of a group or a method, choose the «Comment» radio button. Note that we are talking about the Insider comment, or the comment set in the 4D regular Explorer.
- To find a method in the list, click on the magnifying glass. « @@ can be used. Alt-clic : « find next ».



- To iconize the window, click on the logo « oEd Explorer »



- To show the iconized window, click again on the logo « oEd Explorer »



- To refresh the list of methods (after having created, deleted or renamed some methods), click on the refresh button.



- A clic on the button « Clipboard » copies the selected line in the clipboard. It is also possible to drag&drop the preview area into the icon of this button : this will put the code/comment in the clipboard.



## II. With interface : oEdExpl\_GetList

The developer can get a reference to a hierarchical list so that he/she can create his/her own interface, using the routines of DynamicStructure (ds\_MethodToText, ds\_TextToMethod, ...). To do that, the developer must call the protected method «oEdExpl\_GetList», that returns the reference of the created list. Note that the developer must dispose of it once he/she has finished to work with it.

Each item of the list has a unique reference. The reference is the ID of the object, or it is a number containing the ID. Once you have the ID of the selected objet (method, group, component), you can use it with the differents routines of DynamicStructure. The parameters are the same as for oEdExpl\_launch :

```
refList:=oEdExpl_GetList ("macLicense";"winLicense" ;{showObjectMethods})
```

Each element of the list has a unique reference built with the object ID using the following model :

- \* Method (project, object, form, ....) : the unique reference is the method ID.
- \* Component/Group : a value is added to the group ID. This value depends on the kind of the object (Insider group, component under construction, installed component). It is put in the two High Bytes of the unique reference. For more detail see below.

To know if the selected object is a method or a group and load its code or comment, you can use the following code. (It is important to declare the variables as long integer to be able to manipulate them with Bit operators).

```
C_LONGINT($L_reference;$L_tag;$L_groupID)
GET LIST ITEM (TheList;Selected list item(TheList);$L_reference;$T_label)
$L_tag:=($_reference & 0xFFFF0000)
If ($L_tag=0)
  ^ It is a method. $L_reference = method ID.
  ^ Get the code :
  $L_error:=ds_MethodToText($_reference;$T_code)
  ^ Get the comment :
  $L_error:=ds_GetObjectComments(kds_ObjectComment;$_reference;0;$T_comment)
Else
  ^ $L_tag is #0=> It is a group/Component.
  ^ Get group ID:
  $L_groupID:=($L_reference & 0xFFFF)
  ^ Get the global comment :
  $L_error:=ds_GetObjectComments(kds_GroupComment;$L_groupID;0;$T_comment)
End if
```

### **Deep inside**

For groups/components, the unique reference is built applying a mask to the group ID :

0x02000000 : Insider group

0x03000000 : folder of component under construction.

0x04000000 : sub-folder (Public/Protected/Private) of component under construction.

0x05000000 : Installed component folder

0x06000000 : sub-folder (Public/Protected/Private) of installed component

So, in order to know exactly to wich group belongs the selected « folder », you have to write something like :

```
$L_whichOne :=($L_reference & 0xFF000000)
` $L_whichOne is 0x02000000, 0x03000000, etc.
```

Sub-Lists of the main level used by oEdExplorer have the following unique reference :

- \* Folder "Components (building)" : 0x60000000
- \* Folder " Components (installed)" : 0x70000000
- \* Folder "Form" : 0x10000000
  - > Sub-Folder "table name"
    - (table number << 16)
    - To get the table number :
    - $\$i\_tableNum := (\$L\_reference \gg 16)$
  - > Sub-Folder "Form name"
    - (table number << 16) | Form\_ID
    - To get the form ID:
    - $\$i\_tableNum := (\$L\_reference \gg 16)$
    - $\$i\_formID := (\$L\_reference \& 0xFFFF)$
- \* Folder "Insider Groups " : 0x50000000
- \* Folder "Database methods" : 0x30000000
- \* Folder "Project methods" : 0x40000000
- \* Folder "Triggers" : 0x20000000