

Get_Repos_Entities Function

Description

Returns a list of all entities classified under the specified application, type, and class.

Syntax

entlistt = **Get_Repos_Entities** (*appID*, *typeID*, *classID*, *Integer* [*getflags*] [, *expand*] [, *remote*] [, *moreinfo*])

Parameters

The Get_Repos_Entities function has the following parameters.

Parameter	Description
<i>AppID</i>	Application identifier. If the parameter is null, the current application and its inheritance chain are included.

TypeID	Types defined for OpenInsight.	
	TypeID	Description
	APPNOTE	QuickHelp entities.
	APPROW	This is a catch-all for rows in the application database tables which need to be tracked for impact analysis and/or deployment reasons.
	CONTEXT MENU	Window controls Context Menus.
	DATASOURCE	A collection of DBComponents.
	DBCOLUMN	Database table column.
	DBCOMPONENT	A logical collection of properties that define a view of user data. OpenInsight views Lotus Notes forms and views as DBComponents. Datasets created using the C/S Workspace are also DBComponents.
	DBCOMPONENTEXE	The compiled versions of the Dataset DBComponent.
	DBTABLE	Database table.
	DLL	Windows .DLL file.
	DOC	Document file, in one of specified classes.
	IMAGE	File of image data (.BMP, .ICO, etc.).
	MSG	Prepared application system message.
	NOTESMEDIA	Lotus provided tools used to generate multimedia records. There are no classes. This type behaves exactly as the WINEXE entity type.
	NOTESSMART	Lotus provided SmartSuite tools used to generate documents (AmiPro), spreadsheets (Lotus 1-2-3), etc. There are no classes. This type behaves exactly as the WINEXE entity type.
	OCX	OLE Custom Controls.
	OIEVENT	OpenInsight Event Handler source code. The class id for each OIEVENT are the Event name. e.g. CLICK, OPTIONS, READ, WRITE.
	OIEVENTDBG	Symbol table, for event handlers. The class id for each OIEVENT are the Event name. e.g. CLICK, OPTIONS, READ, WRITE.
	OIEVENTEXE	Executable version of an event handler. The class id for each OIEVENT are the Event name. e.g. CLICK, OPTIONS, READ, WRITE.
	OIREPORT	Report from OpenInsight Report Builder.
	OIWIN	Code form of an OpenInsight window.
	OIWINEXE	Executable form of an OpenInsight window.
	OSFILE	Operating system file.
	POPUP	Executable form of an OpenInsight popup.
	STPROC	Stored procedure, code.
	STPROCDBG	Symbol table, for stored procedure.
	STPROCEXE	Executable stored procedure.
	STPROCINS	Insert table, for stored procedure code.
	WINEXE	Windows .EXE file.
	WINHELP	Windows Help file.
	XML	

classID	<p>Valid classIDs are:</p> <table> <tr> <th>typeID</th><th>Description</th></tr> <tr> <td>DATASOURCE</td><td>CONNECTION is a set of attributes used to connect to Client/Server data. NOTESDB is a Lotus Notes database).</td></tr> <tr> <td>DBCOMPONENT</td><td>DATASET is a data model used to work with Client/Server Data. NOTESAGENT is a registered Lotus Notes V4.0 agent. NOTESFOLDER is a registered Lotus Notes V4.0 folder. NOTESFORM is a registered Lotus Notes form. NOTESMACRO is a registered Lotus Notes V3.0 macro. NOTESVIEW is a registered Lotus Notes database view.</td></tr> <tr> <td>DBCOMPONENTTEXT</td><td>DATASET</td></tr> <tr> <td>DOC</td><td> <p>The class id is the name of the Windows executable used to access the document. Provided class IDs are:</p> <p>AMIPRO</p> <p>CSS</p> <p>HTML</p> <p>NOTEPAD</p> <p>WINWORD</p> <p>WP</p> <p>WRITE</p> <p>New document classes can be added by simply adding to the SYSREPOSCLASSES table. The DOC type-level method logic will automatically call the related program.</p> </td></tr> <tr> <td>IMAGE</td><td>BMP (bitmap). ICO (icon).</td></tr> <tr> <td>OIREPORT</td><td> <p>BRWLAYOUT (Banded Report Writer layout).</p> <p>RBLAYOUT (Report Builder report layout).</p> <p>RBLABEL (Report Builder label layout).</p> <p>BRDLAYOUT (Report Designer report layout).</p> <p>RDQUERY (Report Designer query).</p> </td></tr> </table>	typeID	Description	DATASOURCE	CONNECTION is a set of attributes used to connect to Client/Server data. NOTESDB is a Lotus Notes database).	DBCOMPONENT	DATASET is a data model used to work with Client/Server Data. NOTESAGENT is a registered Lotus Notes V4.0 agent. NOTESFOLDER is a registered Lotus Notes V4.0 folder. NOTESFORM is a registered Lotus Notes form. NOTESMACRO is a registered Lotus Notes V3.0 macro. NOTESVIEW is a registered Lotus Notes database view.	DBCOMPONENTTEXT	DATASET	DOC	<p>The class id is the name of the Windows executable used to access the document. Provided class IDs are:</p> <p>AMIPRO</p> <p>CSS</p> <p>HTML</p> <p>NOTEPAD</p> <p>WINWORD</p> <p>WP</p> <p>WRITE</p> <p>New document classes can be added by simply adding to the SYSREPOSCLASSES table. The DOC type-level method logic will automatically call the related program.</p>	IMAGE	BMP (bitmap). ICO (icon).	OIREPORT	<p>BRWLAYOUT (Banded Report Writer layout).</p> <p>RBLAYOUT (Report Builder report layout).</p> <p>RBLABEL (Report Builder label layout).</p> <p>BRDLAYOUT (Report Designer report layout).</p> <p>RDQUERY (Report Designer query).</p>
typeID	Description														
DATASOURCE	CONNECTION is a set of attributes used to connect to Client/Server data. NOTESDB is a Lotus Notes database).														
DBCOMPONENT	DATASET is a data model used to work with Client/Server Data. NOTESAGENT is a registered Lotus Notes V4.0 agent. NOTESFOLDER is a registered Lotus Notes V4.0 folder. NOTESFORM is a registered Lotus Notes form. NOTESMACRO is a registered Lotus Notes V3.0 macro. NOTESVIEW is a registered Lotus Notes database view.														
DBCOMPONENTTEXT	DATASET														
DOC	<p>The class id is the name of the Windows executable used to access the document. Provided class IDs are:</p> <p>AMIPRO</p> <p>CSS</p> <p>HTML</p> <p>NOTEPAD</p> <p>WINWORD</p> <p>WP</p> <p>WRITE</p> <p>New document classes can be added by simply adding to the SYSREPOSCLASSES table. The DOC type-level method logic will automatically call the related program.</p>														
IMAGE	BMP (bitmap). ICO (icon).														
OIREPORT	<p>BRWLAYOUT (Banded Report Writer layout).</p> <p>RBLAYOUT (Report Builder report layout).</p> <p>RBLABEL (Report Builder label layout).</p> <p>BRDLAYOUT (Report Designer report layout).</p> <p>RDQUERY (Report Designer query).</p>														
getflags	<p>The default is false. If set to true, returns repository entities along with flags in an @SVM-delimited array of literals:</p> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>S1</td><td>State 1.</td></tr> <tr> <td>S0</td><td>State 0.</td></tr> <tr> <td>E</td><td>Evaluate.</td></tr> <tr> <td>R</td><td>Recompile.</td></tr> <tr> <td>C</td><td>Checked-out.</td></tr> </table>	Value	Description	S1	State 1.	S0	State 0.	E	Evaluate.	R	Recompile.	C	Checked-out.		
Value	Description														
S1	State 1.														
S0	State 0.														
E	Evaluate.														
R	Recompile.														
C	Checked-out.														
expand	The default is true. If set to false, returns only entity names (as opposed to fully qualified entity identifiers).														
remote	The default is false. If set to true uses remote (check-out) repository.														
moreinfo	Used when the list exceeds 64K. Set in calling procedure the first time call is made, then used internally by Get_Repos_Entities to control which 64K block of the list is retrieved. When the full list has been retrieved, set to null. (See the example below.)														

Note: This parameter is obsolete in OpenInsight versions 4 and above. Pass a null value to this parameter. The 64k limit has been removed.

See also

[Repository\(\)](#)

Example

```

* Count the number of OpenInsight windows in the current application.
Declare Function Get_Repos_Entities
AppID = @APPID<1>  ;* current application
TypeID = "OIWIN"  ;* OpenInsight Windows
ClassID = ""      ;* Not Applicable
MoreInfo = ""
SubList = Get_Repos_Entities(AppID, TypeID, ClassID, "", "", "", "")
numEntities = dCount(SubList, @fm)
* Count the number of OpenInsight windows in the current application,
* as well as the applications it has been inherited from.
AppID = ""
TypeID = "OIWIN"  ;* OpenInsight Windows
ClassID = ""      ;* Not Applicable
MoreInfo = ""
SubList = Get_Repos_Entities(AppID, TypeID, ClassID, "", "", "", "")
numEntities = dCount(SubList, @fm)

* For use with 16-bit systems
Declare Function Get_Repos_Entities
AppID = @APPID<1>  ;* current application
TypeID = "OIWIN"  ;* OpenInsight Windows
ClassID = ""      ;* Not Applicable
MoreInfo = 1
CNT = 0
Loop
    SubList = Get_Repos_Entities(AppID, TypeID, ClassID, |
                                "", "", "", MoreInfo)

    If Len(SubList) Then
        CNT = (SubList, @FM) +1
    End
While Len(MoreInfo)
Repeat

```