## HexRaysCodeXplorer v1.0

## Authors:

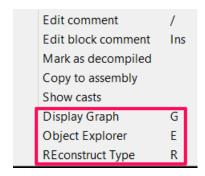
Aleksandr Matrosov (alex.matrosov@gmail.com)

Eugene Rodionov (radius.evgeniy@gmail.com)

**Github repository:** <a href="http://rehints.github.io/HexRaysCodeXplorer/">http://rehints.github.io/HexRaysCodeXplorer/</a>

Official website: <a href="http://rehints.com/">http://rehints.com/</a>

**HexRaysCodeXplorer** - Hex-Rays Decompiler plugin for easier code navigation. Right-click context menu in the Pseudocode window shows HexRaysCodeXplorer plugin commands:



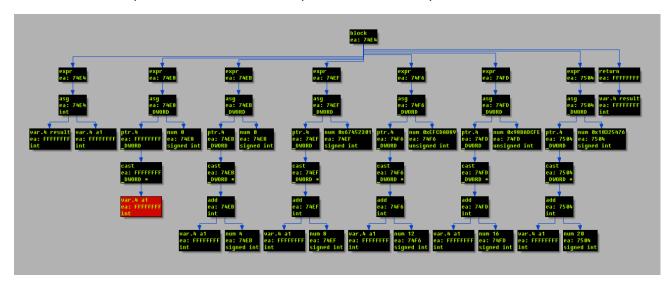
## Here are the main features of the plugin:

 Automatic type REconstruction for C++ objects. To be able to reconstruct a type using HexRaysCodeXplorer one needs to select the variable holding pointer to the instance of position independed code or to an object and by right-button mouse click select from the context menu «REconstruct Type» option:

```
28
22)
23
24
25
26
27
28
2->base64_encode = v4 + 0x388 - *v4;
a2->base64_decode = v4 + 0x46D - *v4;
a2->base64_decode = v4 + 0x46D - *v4;
a2->rotsc.proc = v4 - *v4 + 0x5782 - *v4;
a2->rnd_process_block = v4 + 0x5782 - *v4;
a2->rnd_fill_buffer = v4 + 0x6887;
a2->rind_process_block = v4 + 0x4887;
a2
```

The reconstructed structure is displayed in "Output window". Detailed information about type Reconstruction feature is provided in the blog post "Type REconstruction in HexRaysCodeXplorer".

• C-tree graph visualization – a special tree-like structure representing a decompiled routine in c\_itemt terms. Useful feature for understanding how the decompiler works. The highlighted graph node corresponds to the current cursor position in the HexRays Pseudocode window:



Navigation through virtual function calls in HexRays Pseudocode window. After representing C++
objects by C-structures this feature make possible navigation by mouse clicking to the virtual
function calls as structure fields:

• Object Explorer – useful interface for navigation through virtual tables (VTBL) structures. Object Explorer outputs VTBL information into IDA custom view window. The output window is shown by choosing «Object Explorer» option in right-button mouse click context menu:

## **Object Explorer**

```
0x10256aa0 - 0x10256afc:
                           VECTOR_DATA_2_VTABLE method count: 23
                           FILE_MAPPING_1_VTABLE method count: 10
0x10256bb0 - 0x10256bd8:
0x10256bd8 - 0x10256bf0:
                           GLOBAL EVENT 1 VTABLE method count: 6
                           PROCESS_HANDLE_VTABLE method count: 20
0x102679a0 - 0x102679f0:
0x10267a90 - 0x10267acc:
                           THREAD_HANDLE_VTABLE method count: 15
                          FILE_VTABLE_0 method count: 29
EVENT_VTABLE method count: 6
0x10267b08 - 0x10267b7c:
0x10267bc0 - 0x10267bd8:
                           PROCESS_HANDLE_VTABLE_0 method count: 20
0x10267df0 - 0x10267e40:
0x10267e40 - 0x10267e80:
                           EVENTGLOBAL_HZ_VTABLE method count: 16
0x10267e90 - 0x10267eb0:
                          KASPER_EVENT_ENTRY_VTABLE method count: 8
0x10267f10 - 0x10267f34:
                          TOKEN_HANDLE_VTABLE method count: 9
0x10268118 - 0x10268120:
                          USTRING_REG_PATH_VTABLE method count: 2
                          FILE_1_vTable method count: 31
0x10268128 - 0x102681a4:
                          ENC_2_VTABLE method count: 14
0x10268260 - 0x10268298:
0x10268478 - 0x102684a8:
                           ZLIB_HLPR_VTABLE method count: 12
                          ENC_3_VTABLE method count: 23
SYSTEM_HANDLE_INFO_VTABLE method count: 9
0x102684e0 - 0x1026853c:
0x1026856c - 0x10268590:
                           DICT_1_VTABLE method count: 13
0x10268688 - 0x102686bc:
0x10268d78 - 0x10268dd4:
                           MAIN_VECT_3_VTABLE method count: 23
                           CONCOL HANDLER VTABLE method count: 26
0x10268f80 - 0x10268fe8:
0x102693c0 - 0x102693d0:
                           CMD EXECUTER VIPER VTABLE method count: 4
0x10269490 - 0x102694ec: MAIN_VECT_1_VTABLE method count: 23
```