Subject: Assist++ typedef struct analysis problem
Posted by Xemuth on Sat, 26 Mar 2022 23:49:19 GMT
View Forum Message <> Reply to Message

Hello U++,

I noticed that Assist++ fail to parse some struct. Lets take a look at thoses 4 structs def:

test.h:

```
#ifndef _test_assist_test_h_
#define _test_assist_test_h_

typedef struct StructToResolve {
    uint32_t        engineVersion;
    uint32_t        apiVersion;
} StructToResolve;

typedef struct StructToResolve2 {
    uint32_t        engineVersion;
    uint32_t        apiVersion;
};

//Illegal but we test Assist++ here !
struct StructToResolve3 {
    uint32_t        engineVersion;
    uint32_t        apiVersion;
} StructToResolve3;

struct StructToResolve4 {
    uint32_t        engineVersion;
    uint32_t        apiVersion;
} test;

#endif
```

All four struct are found by assist++. However, auto completion don't work on the first.
Moreover having a struct having a name at begining and variable declaration with the same name are shown has two different struct by Assist++.

test.cpp

```
#include "test.h"
int main(int argc, const char *argv[])
{
 StructToResolve str;
```

```
 str. // Assist++ don't find anything. It don't even open

 StructToResolve2 str2;
 str2.engineVersion; // work fine


 StructToResolve3 str3;
 str3.apiVersion; // Work fine BUT doing a Ctrl+ Space during writting of this type result in 2
distinct StructToResolve3 type. See the screenshot

 StructToResolve4 str4;
 str4.engineVersion; // Work fine BUT ...

 return 0;
}
```

this bug is problematique when working with lib definition a huge amount of the structure which have the same declaration as the first one.
