Subject: B605-dev1: Assist++: Incorrect evaluation
Posted by Werner on Sun, 28 May 2006 13:50:02 GMT
View Forum Message <> Reply to Message

I'm using B605-dev1 under Windows XP.

There are some problems with Assist++. Please, look at the following code:

// code begins here

```
class demo_class;
void global_function();
int global_friend_function(demo_class x);

class demo_class
{
private:
   int static static_private_data;
   int non_static_private_data;

   void static static_private_class_function() {} // class function
   void non_static_private_member_function() {} // instance method

protected:
   int static static_protected_data;
   int non_static_protected_data;

   void static static_protected_class_function() {} // class function
   void non_static_protected_member_function() {} // instance method

public:
   int static static_public_data;
   int non_static_public_data;

   demo_class() {} // constructor
   ~demo_class() {} // destructor
   void static static_public_class_function() {} // class function
   void non_static_public_member_function() {} // instance method
   friend int global_friend_function(demo_class x);
};

void global_function() {}
int global_friend_function(demo_class x) {return x.non_static_private_data;}
```

// code ends here

Calling “Assist / Navigate in the file” yields the following wrong, missing, or misleading information:

First of all Assist++ lists all symbols twice, some of them even four times (compare below). Maybe this is partially the result of “- Assist++: Go to symbol is now case insensitive, listing full matches first and case insensitive matches next“. But in this case there should be an option to switch this feature off. In most cases it just inflates the listing.

Line 03:
Assist++ does not recognize the global friend function as friend although the respective class has been declared beforehand. Yes, I know – no definition so far! But please compare my comment about line 31.

Line 08:
Assist++ tells a wrong name: “static_public_class_function” instead of “static_private_data”. This wrong line is repeated.

Line 11:
Assist++ shows the data symbol instead of the code symbol and tells a wrong name: “static_public_class_function” instead of “static_private_class_function”. This wrong line is repeated.

Line 14:
Assist++ tells a wrong name: “static_public_class_function” instead of “static_protected_data”. This wrong line is repeated.

Line 17:
Assist++ shows the data symbol instead of the code symbol and tells a wrong name: “static_public_class_function” instead of “static_protected_class_function”. This wrong line is repeated.

Line 20:
Assist++ tells a wrong name: “static_public_class_function” instead of “static_public_data”. This wrong line is repeated.

Line 25:
Assist++ shows the data symbol instead of the code symbol. This wrong line is repeated.

Line 27:
Assist++ does not list the declaration of the global friend function at all.

Line 31:
Assist++ does not recognize the global friend function as friend, although it has already been declared as such.

Last but not least “TheIde help / Assist++ / Graphical symbols used by Assist++” should be completed with the “{}” symbol to tell apart declarations from definitions.

Werner

Subject: Re: B605-dev1: Assist++: Incorrect evaluation
Posted by mirek on Sun, 28 May 2006 22:44:59 GMT
View Forum Message <> Reply to Message

Werner wrote on Sun, 28 May 2006 09:50I'm using B605-dev1 under Windows XP.

There are some problems with Assist++. Please, look at the following code:

// code begins here

```
class demo_class;
void global_function();
int global_friend_function(demo_class x);

class demo_class
{
private:
   int static static_private_data;
   int non_static_private_data;

   void static static_private_class_function() {} // class function
   void non_static_private_member_function() {} // instance method

protected:
   int static static_protected_data;
   int non_static_protected_data;

   void static static_protected_class_function() {} // class function
   void non_static_protected_member_function() {} // instance method

public:
   int static static_public_data;
   int non_static_public_data;

   demo_class() {} // constructor
   ~demo_class() {} // destructor
   void static static_public_class_function() {} // class function
   void non_static_public_member_function() {} // instance method
   friend int global_friend_function(demo_class x);
};

void global_function() {}
int global_friend_function(demo_class x) {return x.non_static_private_data;}
```

// code ends here

Calling “Assist / Navigate in the file” yields the following wrong, missing, or misleading information:

First of all Assist++ lists all symbols twice, some of them even four times (compare below). Maybe this is partially the result of “- Assist++: Go to symbol is now case insensitive, listing full matches first and case insensitive matches next“. But in this case there should be an option to switch this feature off. In most cases it just inflates the listing.

Line 03:
Assist++ does not recognize the global friend function as friend although the respective class has been declared beforehand. Yes, I know – no definition so far! But please compare my comment about line 31.

Line 08:
Assist++ tells a wrong name: “static_public_class_function” instead of “static_private_data”. This wrong line is repeated.

Line 11:
Assist++ shows the data symbol instead of the code symbol and tells a wrong name: “static_public_class_function” instead of “static_private_class_function”. This wrong line is repeated.

Line 14:
Assist++ tells a wrong name: “static_public_class_function” instead of “static_protected_data”. This wrong line is repeated.

Line 17:
Assist++ shows the data symbol instead of the code symbol and tells a wrong name: “static_public_class_function” instead of “static_protected_class_function”. This wrong line is repeated.

Line 20:
Assist++ tells a wrong name: “static_public_class_function” instead of “static_public_data”. This wrong line is repeated.

Line 25:
Assist++ shows the data symbol instead of the code symbol. This wrong line is repeated.

Line 27:
Assist++ does not list the declaration of the global friend function at all.

Line 31:
Assist++ does not recognize the global friend function as friend, although it has already been declared as such.

Last but not least “TheIde help / Assist++ / Graphical symbols used by Assist++” should be completed with the “{}” symbol to tell apart declarations from definitions.

Werner

Thanks, this will help.

Assist++ is object of continual improvement. Is it semi-heurestic parser, because full parsing would be too slow, which makes it intrinsically tricky....

Mirek

Subject: Re: B605-dev1: Assist++: Incorrect evaluation
Posted by captainc on Tue, 13 Feb 2007 20:21:21 GMT
View Forum Message <> Reply to Message

I am having a similar issue. After I create a new class and use public and private keywords, the listing is never correct in assist++ auto-complete list.

Example of a similar header file as I define it:

Example.hpp

```
#ifndef EXAMPLE_H
#define EXAMPLE_H

#include <somelib>

using namespace of_some_lib;

class Example{
   private:
      std::vector<std::string> some_vector;
      void myPrivFunc();

   public:
      Example();
      Example(std::string init);
      void myFunc1(std::string str);
};

#endif
```

Note: I get the incorrect assist++ listing in any other file such as Example.cpp .

Also, this is using U++ 2007.1beta
