Subject: Some fixes in InVector.hpp and Vcont.cpp
Posted by kasome on Thu, 25 Sep 2014 11:56:34 GMT
View Forum Message <> Reply to Message

Hi, all

I try to compile U++ to the dynamic library (by using the .def file to generate the .dll with the .lib), and it can work on windows7 except some modification should be made.

In the InVector.hpp

change
```
extern thread__ int64 invector_cache_serial_;
extern thread__ int   invector_cache_blki_;
extern thread__ int   invector_cache_offset_;
extern thread__ int   invector_cache_end_;
```

to
```
#ifndef flagSO
extern thread__ int64 invector_cache_serial_;
extern thread__ int   invector_cache_blki_;
extern thread__ int   invector_cache_offset_;
extern thread__ int   invector_cache_end_;
#else
static thread__ int64 invector_cache_serial_;
static thread__ int   invector_cache_blki_;
static thread__ int   invector_cache_offset_;
static thread__ int   invector_cache_end_;
#endif
```

In the Vcont.cpp

change
```
thread__ int64 invector_cache_serial_;
thread__ int   invector_cache_blki_;
thread__ int   invector_cache_offset_;
thread__ int   invector_cache_end_;
```

to
```
#ifndef flagSO
thread__ int64 invector_cache_serial_;
thread__ int   invector_cache_blki_;
thread__ int   invector_cache_offset_;
thread__ int   invector_cache_end_;
#endif
```

Hope that helps, and the modified code was uploaded as attachment.

File Attachments
1) InVector.hpp, downloaded 538 times
2) Vcont.cpp, downloaded 512 times

Subject: Re: Some fixes in InVector.hpp and Vcont.cpp
Posted by mirek on Thu, 25 Sep 2014 13:29:46 GMT
View Forum Message <> Reply to Message

kasome wrote on Thu, 25 September 2014 13:56Hi, all

I try to compile U++ to the dynamic library (by using the .def file to generate the .dll with the .lib), and it can work on windows7 except some modification should be made.

In the InVector.hpp

change
```
extern thread__ int64 invector_cache_serial_;
extern thread__ int   invector_cache_blki_;
extern thread__ int   invector_cache_offset_;
extern thread__ int   invector_cache_end_;
```

to
```
#ifndef flagSO
extern thread__ int64 invector_cache_serial_;
extern thread__ int   invector_cache_blki_;
extern thread__ int   invector_cache_offset_;
extern thread__ int   invector_cache_end_;
#else
static thread__ int64 invector_cache_serial_;
static thread__ int   invector_cache_blki_;
static thread__ int   invector_cache_offset_;
static thread__ int   invector_cache_end_;
#endif
```

In the Vcont.cpp

change
```
thread__ int64 invector_cache_serial_;
thread__ int   invector_cache_blki_;
thread__ int   invector_cache_offset_;
thread__ int   invector_cache_end_;
```

to

```
#ifndef flagSO
thread__ int64 invector_cache_serial_;
thread__ int   invector_cache_blki_;
thread__ int   invector_cache_offset_;
thread__ int   invector_cache_end_;
#endif
```

Hope that helps, and the modified code was uploaded as attachment.

I doubt this change maintains the current behaviour.

What is the error reported?

Mirek

Subject: Re: Some fixes in InVector.hpp and Vcont.cpp
Posted by kasome on Sat, 27 Sep 2014 04:35:39 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks for the reply, I already upload the entire UppDLL project in the following download link, which can be compiled by using VS2012

http://www.mediafire.com/download/05ejdu5dm0uoddc/TestUppDLL .rar

Just use VS2012 to open the following two solutioin file and run on Windows 7

TestUppDLL\WithModification\UppLib_WithModification.sln ------> no crash

TestUppDLL\WithoutModification\UppLib_WithoutModification.sl n ------> crash

The only difference between "UppLib_WithModification" and "UppLib_WithoutModification" is the modification of InVector.hpp and Vcont.cpp as mentioned.

Notice:
Do not use or test the compiled U++ dll on the Windows XP or before Windows Vista, the reason is the follwoing:

Rules and Limitations for TLS

Quote:On Windows operating systems before Windows Vista, __declspec(thread) has some limitations. If a DLL declares any data or object as __declspec(thread), it can cause a protection fault if dynamically loaded. After the DLL is loaded with LoadLibrary, it causes system failure whenever the code references the __declspec(thread) data. Because the global variable space for a thread is allocated at run time, the size of this space is based on a calculation of the requirements of the application plus the requirements of all the DLLs that are statically linked.

When you use LoadLibrary, you cannot extend this space to allow for the thread local variables declared with __declspec(thread). Use the TLS APIs, such as TlsAlloc, in your DLL to allocate TLS if the DLL might be loaded with LoadLibrary.

Subject: Re: Some fixes in InVector.hpp and Vcont.cpp
Posted by mirek on Sat, 27 Sep 2014 10:01:04 GMT
View Forum Message <> Reply to Message

I am well aware of that; but that is only interesting if you are using "LoadLibrary" (instead of adding .dll with linker).

In that case thread variables indeed cannot work, but that would require much more changes than just this. And it would have serious impact on performance too.

But the real problem is that proposed patch is simply wrong, it just generates hard to spot bugs.

(OTOH, congratulation on the project).

Subject: Re: Some fixes in InVector.hpp and Vcont.cpp
Posted by mirek on Sat, 27 Sep 2014 16:22:15 GMT
View Forum Message <> Reply to Message

I wanted to test your code with VS 2013, but I am stuck with this error:

```
1>------ Build started: Project: UppLib, Configuration: Debug Win32 ------
1>  Xml.cpp
1>c:\program files (x86)\windows kits\8.1\include\um\rpcnsip.h(34): error C2146: syntax error : missing ';' before identifier 'LookupContext'
1>c:\program files (x86)\windows kits\8.1\include\um\rpcnsip.h(34): error C4430: missing type specifier - int assumed. Note: C++ does not support default-int
```

(The "Xml.cpp" file is (originally) Core/Rpc/Xml.cpp)

I am perhaps too unfamiliar with VS to find out what is going wrong here... Any ideas?

Mirek

Subject: Re: Some fixes in InVector.hpp and Vcont.cpp
Posted by kasome on Sat, 27 Sep 2014 18:32:51 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Performing the following steps:

VS2012:
modify
C:\Program Files (x86)\Windows Kits\8.0\Include\um\ObjIdl.h

change
#include "rpc.h"

to
#include "C:\Program Files (x86)\Windows Kits\8.0\Include\shared\rpc.h"

VS2013:
modify
C:\Program Files (x86)\Windows Kits\8.1\Include\um\ObjIdl.h

change
#include "rpc.h"

to
#include "C:\Program Files (x86)\Windows Kits\8.1\Include\shared\rpc.h"

Somehow, when VS2012/VS2013 read the line #include "rpc.h", it will try to include the wrong file MSVS_Ultimate++\upp\uppsrc\Core\Rpc\Rpc.h, and that cause error.
