Subject: BUG Access violation when closing FileSel
Posted by Maginor on Thu, 22 Dec 2022 12:35:13 GMT
View Forum Message <> Reply to Message

Hi, I get the following bug when I click OK on a FileSel. It only seems to happen if the rest of the function after ExecuteOpen() takes a long time, and it does not happen consistently. ExcecuteOpen() is called inside a callback function attached to a toolbar button.

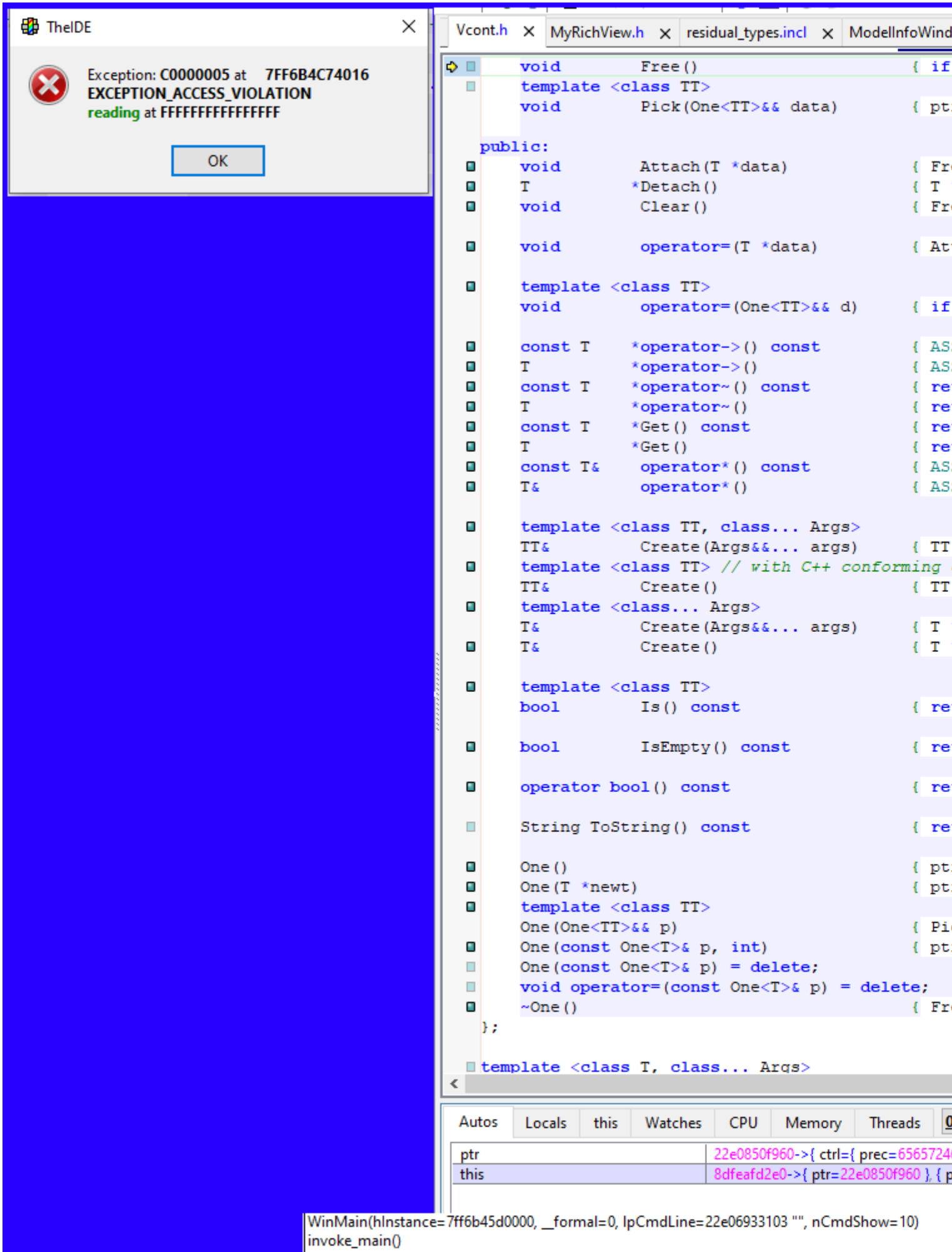
As you can see, the call stack does not involve any user code, and I get the same bug in two different ultimate++ applications that have fairly different user code.
If I make a PromptOK() call right after ExecuteOpen() and before the rest of the function is executed, the crash does not happen.

I am getting this both with ClangX64 and MSVCx64 compilation setups (Windows 10).

The lambda in the call stack is the following one inside DisplayPopup.cpp, DisplayPopup::Set(...)
popup->WhenClose << [=] { PostCallback([=] { popup.Clear(); }); };

File Attachments

1) bug.png, downloaded 718 times


TheIDE
Exception: C0000005 at 7FF6B4C74016
EXCEPTION_ACCESS_VIOLATION
reading at FFFFFFFFFFFFFFFF
OK
Vcont.h
MyRichView.h
residual_types.incl
ModelInfoWind
void Free() { if
template <class TT>
void Pick(One<TT>&& data) { pt
public:
void Attach(T *data) { Fr
T *Detach() { T
void Clear() { Fr
void operator=(T *data) { At
template <class TT>
void operator=(One<TT>&& d) { if
const T *operator->() const { AS
T *operator->() { AS
const T *operator~() const { re
T *operator~() { re
const T *Get() const { re
T *Get() { re
const T& operator*() const { AS
T& operator*() { AS
template <class TT, class... Args>
TT& Create(Args&&... args) { TT
template <class TT> // with C++ conforming
TT& Create() { TT
template <class... Args>
T& Create(Args&&... args) { T
T& Create() { T
template <class TT>
bool Is() const { re
bool IsEmpty() const { re
operator bool() const { re
String ToString() const { re
One() { pt
One(T *newt) { pt
template <class TT>
One(One<TT>&& p) { Pi
One(const One<T>& p, int) { pt
One(const One<T>& p) = delete;
void operator=(const One<T>& p) = delete;
~One() { Fr
};
template <class T, class... Args>
Autos
Locals
this
Watches
CPU
Memory
Threads
ptr
22e0850f960->{ ctrl={ prec=6565724
this
8dfeafd2e0->{ ptr=22e0850f960 }, { p
WinMain(hInstance=7ff6b45d0000, __formal=0, lpCmdLine=22e06933103 "", nCmdShow=10)
invoke_main()