Inside the Microsoft STL: The std::exception_ptr

devblogs.microsoft.com/oldnewthing/20200820-00

August 20, 2020

Raymond Chen

When debugging, you may find yourself staring at a `std::exception_ptr` and want to know what exception is inside it.

What you see in the MSVC header file is that a `std::exception_ptr` is a class that consists of two pointers enigmatically named `_Data1` and `_Data2`.

The dirty secret is that a `std::exception_ptr` is a `std::shared_ptr` in disguise.

Prerequisite: Advanced STL, part 1: `shared_ptr` by Stephan T. Lavavej.

The `_Data1` acts as the `_Ptr` and points to the shared object. The `_Data2` acts as the `_Rep` and points to the control block.

For debugging purposes, you can ignore the `_Data2` and focus on the `_Data1`, which is a pointer to an `EXCEPTION_RECORD`.

Once you have the `EXCEPTION_RECORD`, you can use the `.exr` command to view it, and then use the existing cookbook for extracting the thrown object and its type information.

In practice, you don't usually need to go through the whole cookbook. The `Parameter[1]` points to the object that was thrown, and that object usually contains enough information to let you figure out what it is.

We'll look at some of the possibilities next time.

Raymond Chen

Follow