Microsoft

Time Travel Debugging

Root Causing Bugs in Commercial Scale Software

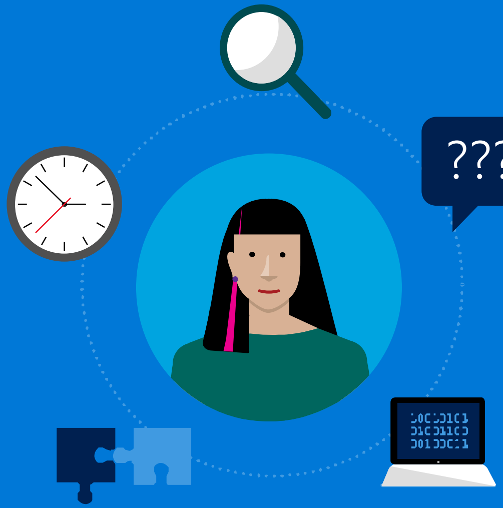
Ken Sykes
Principal Software Engineer
ken.sykes@microsoft.com

Jordi Mola
Principal Software Engineering Lead
jordim@microsoft.com

James McNellis
Senior Software Engineer
james.mcnellis@microsoft.com

What is
Time Travel Debugging?

Overview



TTD is a reverse debugging solution

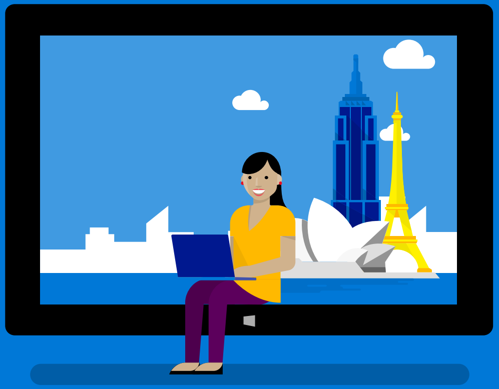
- Record code execution
- Replay it forwards and backwards
- Search trace using queries

Problems to solve

- Debugging is time consuming
- Debugging is hard
- Debugging is complex
- Debugging may require multiple repros

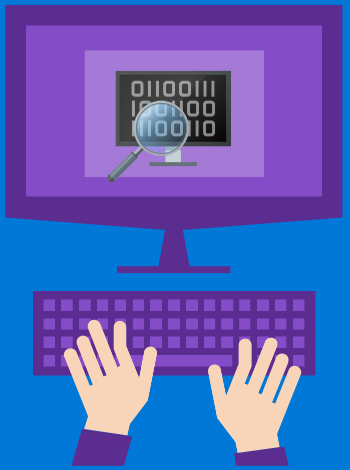
Key Features

- Based upon a Microsoft Research project
- Ready for public TTD preview today
- Written in C++ of many kinds
- Multi-threaded & multi-core recorder
- Shared memory and async I/O support
- Ability to query trace data
- Three step: record, index, replay

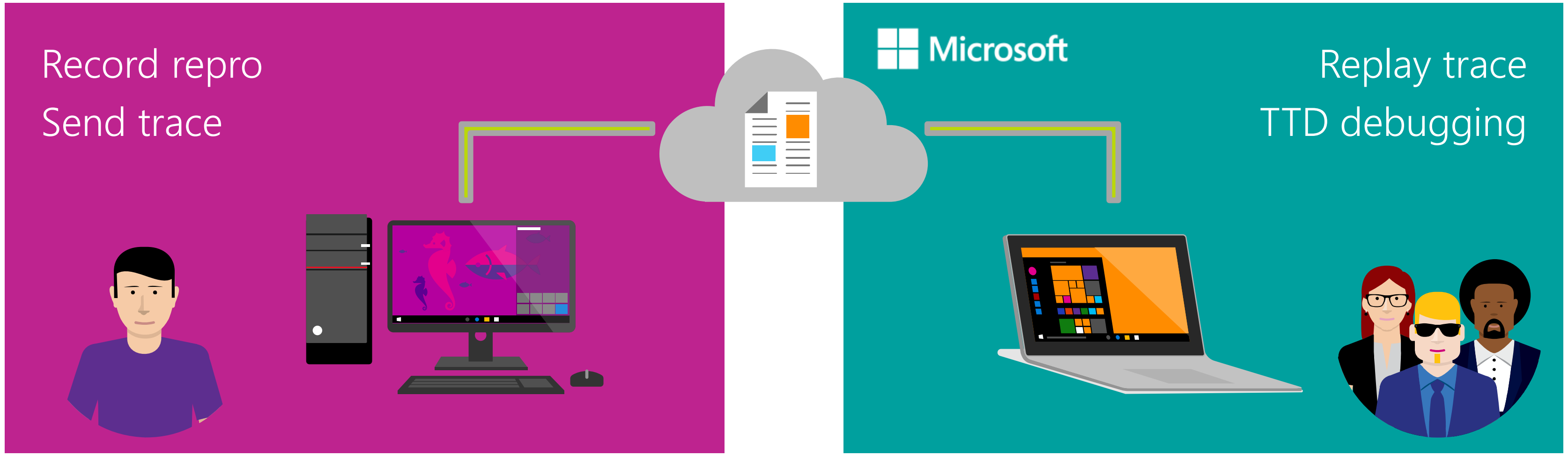


More details coming https://aka.ms/WinDbgBlog

How we use TTD



Microsoft engineers use TTD for solving tough customer problems



Record repro

Send trace

Replay trace

TTD debugging

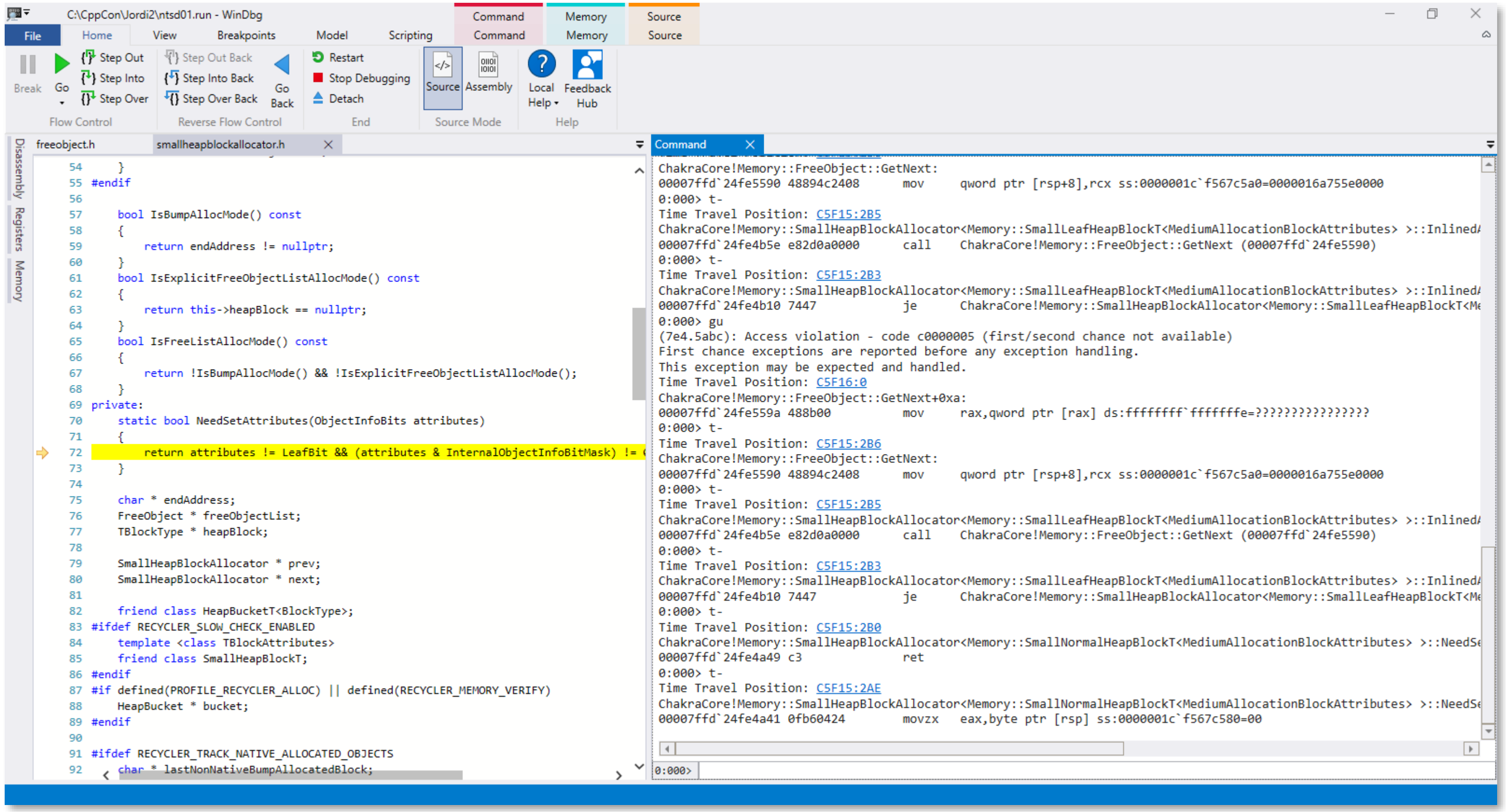
Types of Issues

- Difficult repro bugs like inconsistent crashes
- Corrupted memory, race conditions, resource leaks, improper API use, etc.

How to get TTD

You can start using TTD preview by installing WinDbg Preview from the Store

- https://aka.ms/BlogTTDPreview



```
    }
#endif

    bool IsBumpAllocMode() const
    {
        return endAddress != nullptr;
    }
    bool IsExplicitFreeObjectListAllocMode() const
    {
        return this->heapBlock == nullptr;
    }
    bool IsFreeListAllocMode() const
    {
        return !IsBumpAllocMode() && !IsExplicitFreeObjectListAllocMode();
    }
private:
    static bool NeedSetAttributes(ObjectInfoBits attributes)
    {
        return attributes != LeafBit && (attributes & InternalObjectInfoBitMask) !=
    }

    char * endAddress;
    FreeObject * freeObjectList;
    TBlockType * heapBlock;

    SmallHeapBlockAllocator * prev;
    SmallHeapBlockAllocator * next;

    friend class HeapBucketT<BlockType>;
#ifdef RECYCLER_SLOW_CHECK_ENABLED
    template <class TBlockAttributes>
    friend class SmallHeapBlockT;
#endif
#if defined(PROFILE_RECYCLER_ALLOC) || defined(RECYCLER_MEMORY_VERIFY)
    HeapBucket * bucket;
#endif

#ifdef RECYCLER_TRACK_NATIVE_ALLOCATED_OBJECTS
    char * lastNonNativeBumpAllocatedBlock;
```

```
ChakraCore!Memory::FreeObject::GetNext:
00007ffd`24fe5590 48894c2408      mov     qword ptr [rsp+8],rcx ss:0000001c`f567c5a0=0000016a755e0000
0:000> t-
Time Travel Position: C5F15:2B5
ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallLeafHeapBlockT<MediumAllocationBlockAttributes> >::Inlined
00007ffd`24fe4b5e e82d0a0000      call    ChakraCore!Memory::FreeObject::GetNext (00007ffd`24fe5590)
0:000> t-
Time Travel Position: C5F15:2B3
ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallLeafHeapBlockT<MediumAllocationBlockAttributes> >::Inlined
00007ffd`24fe4b10 7447            je      ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallLeafHeapBlockT<Me
0:000> gu
(7e4.5abc): Access violation - code c0000005 (first/second chance not available)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
Time Travel Position: C5F16:0
ChakraCore!Memory::FreeObject::GetNext+0xa:
00007ffd`24fe559a 488b00          mov     rax,qword ptr [rax] ds:ffffffff`fffffffe=????????????????
0:000> t-
Time Travel Position: C5F15:2B6
ChakraCore!Memory::FreeObject::GetNext:
00007ffd`24fe5590 48894c2408      mov     qword ptr [rsp+8],rcx ss:0000001c`f567c5a0=0000016a755e0000
0:000> t-
Time Travel Position: C5F15:2B5
ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallLeafHeapBlockT<MediumAllocationBlockAttributes> >::Inlined
00007ffd`24fe4b5e e82d0a0000      call    ChakraCore!Memory::FreeObject::GetNext (00007ffd`24fe5590)
0:000> t-
Time Travel Position: C5F15:2B3
ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallLeafHeapBlockT<MediumAllocationBlockAttributes> >::Inlined
00007ffd`24fe4b10 7447            je      ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallLeafHeapBlockT<Me
0:000> t-
Time Travel Position: C5F15:2B0
ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallNormalHeapBlockT<MediumAllocationBlockAttributes> >::NeedSe
00007ffd`24fe4a49 c3              ret
0:000> t-
Time Travel Position: C5F15:2AE
ChakraCore!Memory::SmallHeapBlockAllocator<Memory::SmallNormalHeapBlockT<MediumAllocationBlockAttributes> >::NeedSe
00007ffd`24fe4a41 0fb60424        movzx   eax,byte ptr [rsp] ss:0000001c`f567c580=00

0:000>
```

Enough talk
Time for demos

Demo: The basics

James McNellis

Key takeaways

Here is what you just saw

- How to record a process
- How to step forward and backward
- Replay is read-only

- Debugger data model
- How to find events easily
- How to use memory breakpoints

Demo: Real commercial scale software - Chakra

Jordi Mola

Key takeaways

Here is what you just saw

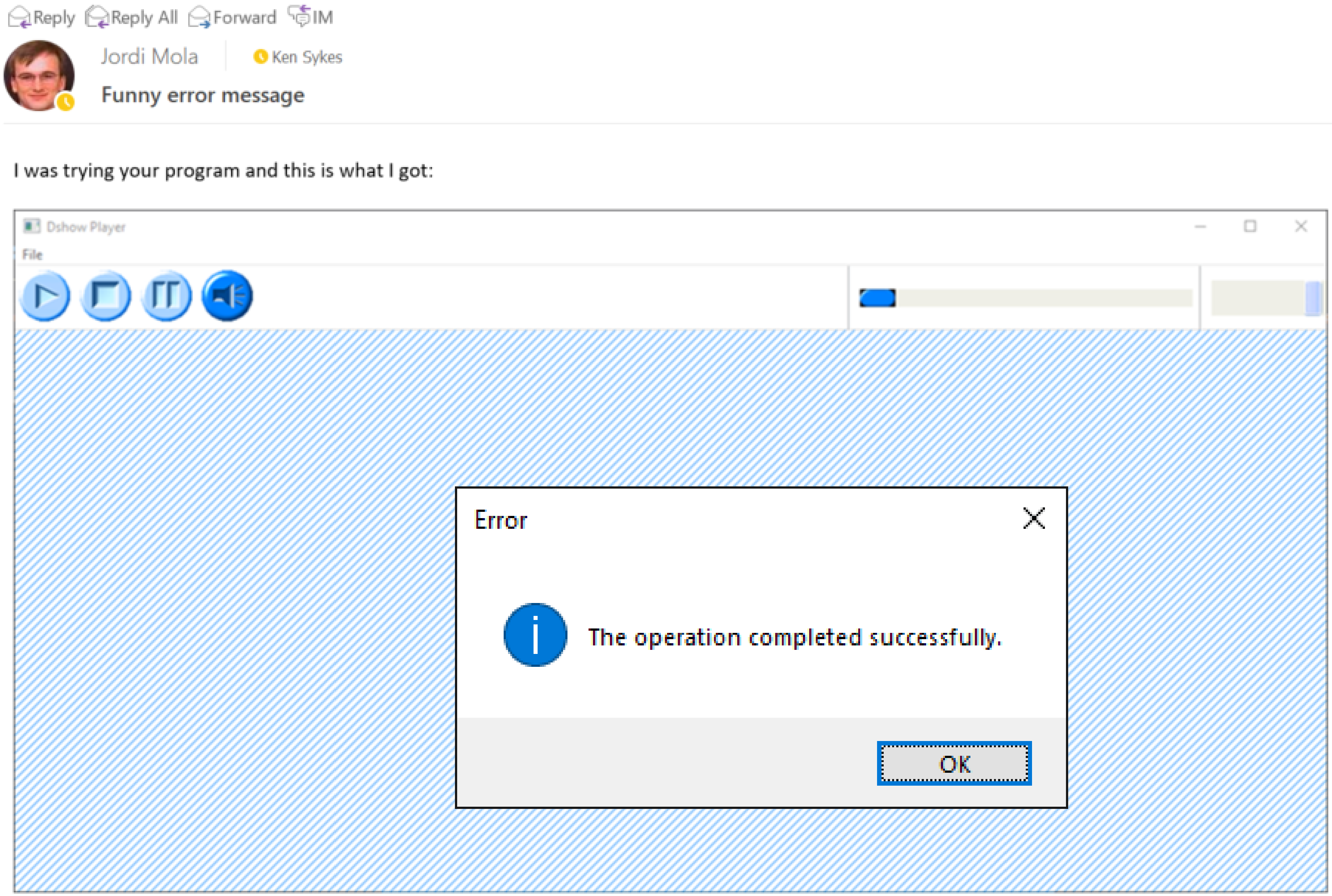
- There is overhead when recording
- Hard problems become easy
- One trace instead of a dozen restarts

Demo: Where did that error code come from?

Ken Sykes

Querying for error codes

When you get an email from your manager at 4pm on a Friday...



Jordi Mola | Ken Sykes

Funny error message

I was trying your program and this is what I got:

Error

The operation completed successfully.

Key takeaways

Here is what you just saw

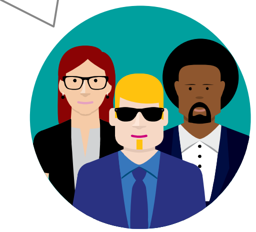
- Send me a trace not a repro
- Power of queries
- C# concepts at CppCon!!!

What's next in TTD

Future areas of work

- Improve recording experience
- More powerful query support
- Add dynamic code analysis
- Kernel tracing support for VMs

Want to help?
We are hiring!!!



Key takeaways

Here is what you just saw

- Ken saves the day … again
- The debugger is scriptable
- JavaScript concepts at CppCon!!!

Microsoft

Q&A

Have more questions … find someone with this shirt

Microsoft

Other Microsoft talks at CppCon

- Today @ 3:15pm: What's New and Upcoming in Visual Studio 2017 Updates (Steve Carroll and Daniel Moth)
- Today @ 3:15pm: C++ Development with Visual Studio Code (Rong Lu)
- Tuesday @ 9:00am: Everything You Ever Wanted to Know about DLLs (James McNellis)
- Thursday @ 3:15pm: Naked coroutines live (with networking) (Gor Nishanov)
- Friday @ 4:15pm: C++/WinRT and the Future of C++ on Windows (Kenny Kerr and Scott Jones)

\0