

+ 25

MSVC C++ Dynamic Debugging

How We Enabled Full Debuggability of Optimized Code

ERIC BRUMER



Cppcon
The C++ Conference

20
25



September 13 - 19

/dynamicdeopt

<https://aka.ms/vcdd>

<https://aka.ms/dynamicdebugging>

Demo!

<https://aka.ms/vcdd>
<https://aka.ms/dynamicdebugging>

Source code
(test.cpp)

```
int two() {  
    return 2;  
}
```

```
int func() {  
    int x = 1;  
    x += two();  
    x += 3;  
    return x;  
}
```

Optimized asm
(test.exe)

```
0000000140001000: <16 bytes of padding>  
?func@@YAHXZ:  
    0000000140001010: mov eax, 6  
    0000000140001015: ret
```

Source code
(test.cpp)

```
int two() {  
    return 2;  
}
```

```
int func() {  
    int x = 1;  
    x += two();  
    x += 3;  
    return x;  
}
```

Optimized asm
(test.exe)



```
0000000140001000: jmp qword ptr [rip+0x2]  
0000000140001006: cc cc  
0000000140001008: 0x0007777140001010  
?func@@YAHXZ:  
0000000140001010: jmp rip-0x10  
0000000140001015: ret
```

Source code
(test.cpp)

```
int two() {  
    return 2;  
}  
  
int func() {  
    int x = 1;  
    x += two();  
    x += 3;  
    return x;  
}
```

Optimized asm
(test.exe)

```
0000000140001000: jmp qword ptr [rip+0x2]  
0000000140001006: CC CC  
0000000140001008: 0x0007777140001010  
?func@@YAHXZ:  
0000000140001010: jmp rip-0x10  
0000000140001015: ret
```

10 byte
mov instr

Unoptimized asm
(test.alt.exe)

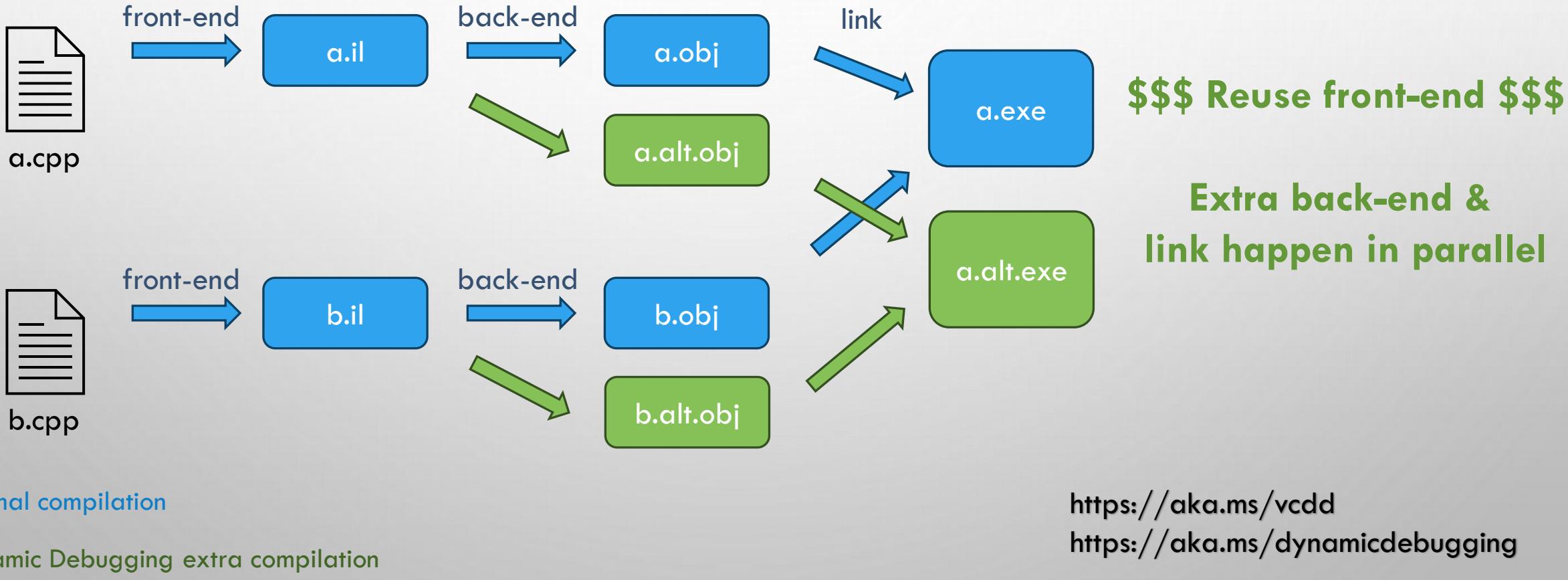
```
?func@@YAHXZ:  
0007777140001010: sub    rsp,38h  
0007777140001014: mov     dword ptr [rsp+20h],1  
000777714000101C: mov     rax, 0x0000000140001030  
0007777140001026: call    rax  
0007777140001028: mov     ecx, dword ptr [rsp+20h]  
000777714000102C: add     ecx, eax  
000777714000102E: mov     eax, ecx  
0007777140001030: mov     dword ptr [rsp+20h], eax  
0007777140001034: mov     eax, dword ptr [rsp+20h]  
0007777140001038: add     eax, 3  
000777714000103B: mov     dword ptr [rsp+20h], eax  
000777714000103F: mov     eax, dword ptr [rsp+20h]  
0007777140001043: add     rsp, 38h  
0007777140001047: ret
```

Address of two()
in opt binary

Compile time

<https://aka.ms/vcdd>
<https://aka.ms/dynamicdebugging>

Threadripper PRO 5975WX 32 cores	Release	Release + /dynamicdeopt	Slowdown
end-to-end	132s	139s	5%
iteration	8.8s	10.3s	17%

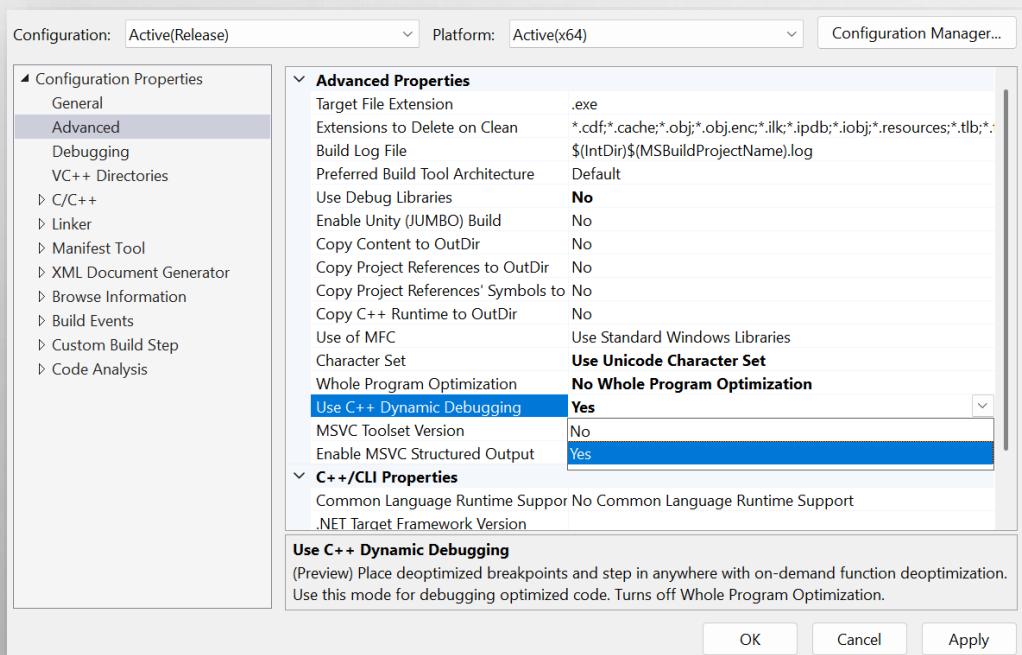


VISUAL STUDIO W/ MSVC ON WINDOWS ONLY

CURRENT (NOT NECESSARILY PERMANENT) LIMITATIONS:

- X64 ONLY
- NON-LTCG ONLY
- FULL LIST OF INCOMPATIBILITIES AT [HTTPS://AKA.MS/VCDD](https://aka.ms/vcdd)

BUILD SYSTEM INTEGRATION



Enable C++ Dynamic Debugging in Unreal Engine

Unreal Engine 5.6 supports C++ Dynamic Debugging for both Unreal Build Tool and Unreal Build Accelerator. There are two ways to enable it:

- Modify your project's `Target.cs` file to contain `WindowsPlatform.bDynamicDebugging = true`.
- Use the **Development Editor** configuration, and modify `BuildConfiguration.xml` to include:

```
XML Copy
<WindowsPlatform>
  <bDynamicDebugging>true</bDynamicDebugging>
</WindowsPlatform>
```

<https://aka.ms/vcdd>
<https://aka.ms/dynamicdebugging>

Undoing optimizations

<https://aka.ms/vcdd>
<https://aka.ms/dynamicdebugging>

MICROSOFT @ CPPCON

<https://aka.ms/cppcon/dd>



Tues.	14:00	What's new in Visual Studio for C++ Developers in 2025	Augustin Popa & David
	14:00	Back to Basics: Code Review	Glenn Lloyd, Paul Stoffregen
	14:00	LLMs in the Trenches: Boosting System Programming with LLMs	Ramneek Bakshi & Pranav Grade
	15:15	C++ Performance Tips: Cutting Down on Unnecessary Objects	
Wed.	15:50	Connecting C++ Tools to AI Agents Using the Model Context Protocol	Ben McMorran
	Welcome to v1.0 of the <code>meta::[[verse]]</code> !		
Thurs.	14:00	MSVC C++ Dynamic Debugging: How We Enabled Full Debuggability of Optimized Code	Eric Brumer
	16:45	It's Dangerous to Go Alone: A Game Developer Tutorial	Michael Price
Fri	9:00	Reflection-based JSON in C++ at Gigabytes per Second	Daniel Lemire & Francisco Geiman Thiesen
	13:30	Duck-Tape Chronicles: Rust/C++ Interop	Victor Ciura

**Send us your feedback.
Last chance for a prize!**