Advanced Developers Conference
Development for Professionals!

Debugging Tools For Windows

Jochen Kalmbach
Microsoft MVP VC++

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Advanced Developers Conference
Development for Professionals!

Agenda

Jochen Kalmbach
Microsoft MVP VC++

- Grundlegendes
  - „Where to get it?“
  - Kurzvorstellung der verschiedenen Tools
  - Grundlegender Umgang mit WinDbg
- Symbole / Symbol-Server / Source-Server
- Fehlersuche
  - (Crash-) Dump Analyse
  - Handle- / Memory-Leaks
  - Memory Corruptions
  - Silent Process Exits
- Automatisierung

Partner: intel Software, embarcadero, IncrediBuild BEYOND ACCELERATION, ppedv

Relativ viel Praxis…

Advanced Developers Conference
Development for Professionals!

„Where to get it?“

Jochen Kalmbach
Microsoft MVP VC++

- Debugging Tools For Windows
  - Integriert in WDK
    - WDK 8.1 Update (für Win7 und höher)
      - http://www.microsoft.com/en-us/download/details.aspx?id=42273
    - WDK 7.1 (für XP und höher)
      - http://www.microsoft.com/en-us/download/confirmation.aspx?id=11800
  - Integriert in Windows SDK
    - Windows 8.1 SDK (für Win7 und höher)
      - http://msdn.microsoft.com/en-us/windows/desktop/bg162891.aspx
    - Windows 7.1 SDK (für XP und höher)
      - http://www.microsoft.com/en-us/download/details.aspx?id=8279

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Früher gab es dies als separaten Download…

DebuggingTools: http://msdn.microsoft.com/en-us/library/windows/hardware/ff551063
WDK: http://msdn.microsoft.com/en-US/windows/hardware/gg454513

Beim SDK/WDK Download kann man aber auch „Download“ auswählen, und so kann man es später „offline“ installieren

Advanced Developers Conference
Development for Professionals!

Mitgelieferte Tools

Jochen Kalmbach
Microsoft MVP VC++

- Debugger: windbg, cdb, ntsd, kd
  - div. Extensions / Remote-Debugger
- gflags (setzt im OS-Kernel „Flags“)
- symstore, symcheck, dbh (Verwaltung von Symbolen)
- Umdh (Memory Leak Analyse)
- Logger, logviewer (API Logging)
- tlist, kill, rtlist, adplus, u.v.a.

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Es werden 4 Debugger mitgeliefert

WinDbg (GUI based, erlaubt User- und Kernel-Mode debugging)
Cdb (Console-based, nur User-Mode)
Ntsd (GUI-based ohne GUI ;))
Kd (Console-based, Kernel-Mode Debugger)

Advanced Developers Conference
Development for Professionals!

Warum WinDbg?

Jochen Kalmbach
Microsoft MVP VC++

- Kostenlos
- „xcopy“ Deployment / Relativ klein
  - Einfach auf Remote-Systemen übertragbar/verwendbar
    - C:\Program Files (x86)\Windows Kits\8.1\Debuggers
- Funktioniert immer, auch wenn andere Debugger versagen
- Kann automatisiert werden
- Viele nützliche Tools (gflags, symstore, usw.)

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Advanced Developers Conference
Development for Professionals!

WinDbg - Befehlsübersicht

Jochen Kalmbach
Microsoft MVP VC++

- Normale Commands (beginnen mit einem Buchstaben)
  - Um den Prozess zu debuggen
  - z.B. „k“, „lm“, „g“, „d“, usw.
- Meta-Commands (beginnen mit einem „.“)
  - Um den Debugger zu beeinflussen
  - z.B. „.sympath“, „.cls“, „.lastevent“, „.if“, „.logopen“, usw.
- Extension-Commands (beginnen mit einem „!“)
  - Als DLL implementiert
  - Dies macht WinDbg erst richtig „mächtig“
  - Vorinstallierte Extensions (exts.dll/uext.dll/ntsdexts.dll/kdexts.dll/logexts.dll) / sos.dll
  - API/SDK Verfügbar
  - z.B. „!analyze“, „!handle“, „!peb“, „!locks“, usw.

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

➔ Wir beschränken uns hier nur auf Applikations-Debugging / kein Kernel-Debugging

!exts.help =>Allgemein
!uext.help => UserMode (non-OS-Specific)
!ntsdexts.help => UserMode (OS-Specific)
!kdexts.help => Kernel-Mode
!logexts.help => API Logger
!sos.help => Managed Debugging (muss zuerst geladen werden)

Meta-Befehle:
- Hier kann man auch komplexe Abläufe „programmieren“, falls man automatisiert bestimmte Datenstrukturen durchsuchen oder ausgeben will / oder gcroot-Analyse

Extensions:
- Bekannnteste ist wohl „sos.dll“

Wichtig: HILFE zeigen!

Advanced Developers Conference
Development for Professionals!

WinDbg – Dump-Analyse

Jochen Kalmbach
Microsoft MVP VC++

- Warum Dumps?
  - Sichert den aktuellen Zustand
  - Erlaubt späteres Debuggen dieses Zustandes (Symbole müssen nicht ausgeliefert werden!)
- Woher kommen die Dumps?
  - WER (Windows Error Reporting)
  - WER (local dumps ➔ Empfohlener Weg!)
  - AeDebug-Registry-Key
  - Eigener UnhandledExceptionFilter Handler
  - Von Hand erzeugt via TaskManager / cdb (.dump)
    - .dump /ma /u <Name> (a = All ➔ Full Dump)
    - .dump /m /u <MiniDump> (Nur Callstacks)

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Wenn ein Fehler aufgetreten ist, was dann???

Advanced Developers Conference
Development for Professionals!

WinDbg – Dump-Analyse

Jochen Kalmbach
Microsoft MVP VC++

- WER- Local dumps
  - WER erlaubt auch das schreiben von „lokalen Dumps“ (ab Vista SP1!)
  - Registry-Key:
    - KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps
  - Globale-Settings und App-Settings möglich
  - Jede Applikation kann sich hier registrieren mit:
    - <AppName.exe>
      - DumpFolder, REG_EXPAND_SZ, Default: %LOCALAPPDATA%\CrashDumps
      - DumpCount, REG_DWORD, Default: 10
      - DumpType, REG_DWORD (0:Custom, 1:Mini, 2:Full), Default: 1
      - CustomDumpFlags, REG_DWORD (MINIDUMP_TYPE values)

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Leider sind Admin-Rechte zum installieren erforderlich; es gibt keine Unterstützung in HKCU!

PS: Es wird nicht empfohlen die DumpFolder zu setzen, da hier das System je nach Typ das passende wählt, wo auch schreibzugriff möglich ist!

- %WINDIR%\System32\Config\SystemProfile
- %WINDIR%\ServiceProfiles
- %LOCALAPPDATA%\CrashDumps

Advanced Developers Conference
Development for Professionals!

WER - local

Jochen Kalmbach
Microsoft MVP VC++

- DEMO

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Demo: WER_UserModeDumps

Advanced Developers Conference
Development for Professionals!

WinDbg – Dump-Analyse

Jochen Kalmbach
Microsoft MVP VC++

- VS
  - Open File…
- WinDbg
  - Geht für OS und App-Dumps
- Automatische Analyse von fast alles Crashes
  - .symfix+
  - .sympath+ …
  - .reload
  - !analyze -v
- Von Hand (bei Exceptions)
  - .ecxr ➔ Setzt den Context auf die aktuelle Exception
- Für sinnvolle Analyse werden PDB-Dateien benötigt

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

CrashDumpLocal01 verwenden
- VS öffnen; Compilieren / kurze Erklärung für den InProcess-Dump
- Prozess starten
- Fehlermeldung wird ausgegeben
- Nachschauen ob oben der Symbol-Pfad gesetzt ist!
- !analyze –v

➔ Kurze Erklärung der Ausgabe…

Ausgabe von Hand mit Maus ;)
- .ecxr
- kM

➔ Klick auf „04“ ➔ Source-Anzeige
- Menü „Debugger“ aufmachen und weitere Dinge zeigen

Zeigen, dass es so auch möglich ist Dumps von kritischen Punkten zu erstellen, dazu das Beispiel im Unterverzeichnis „Automatisch“ öffnen! (siehe Automation\CrashDumpLocal01)

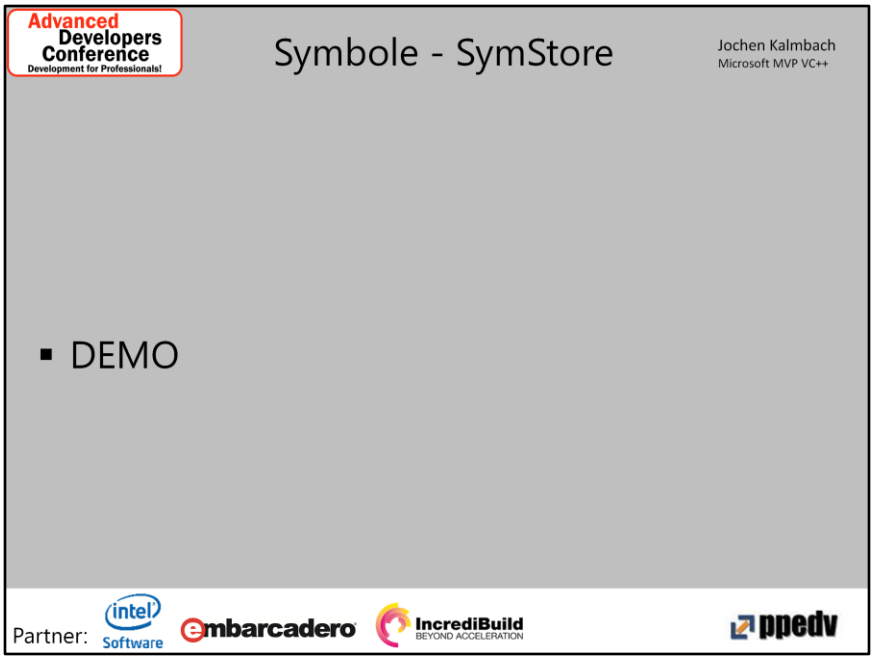
Advanced Developers Conference
Development for Professionals!

Symbole - SymStore

Jochen Kalmbach
Microsoft MVP VC++

- Verwaltung von Symbolen
  - Problem:
    - Wo liegen die EXE/DLL/PDB Dateien um einen Crash-Dump zu analysieren?
  - Lösung:
    - Zentrale Ablage der Dateien und automatisches finden dieser ;)
    - Benötigt nur ein ganz normales Dateisystem
    - Symstore add /r /f <SourcePath> /s <StorePath> /t <Product>

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv



CrashDumpLocal01 verwenden

1. CrashDumpLocal01 übersetzen;
2. Die Symbole in den Store legen! (StoreSymbols.cmd) ➔ Ausführlich die „StoreSymbols.cmd“ zeigen!!!

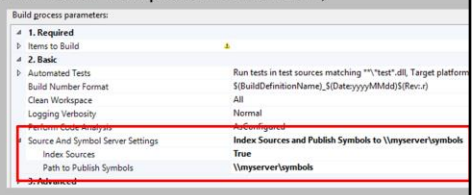
%symstore% add /r /f .\Crash*.pdb /s ..\symbols /t "CrashDumpLocal01“
%symstore% add /r /f .\Crash*.exe /s ..\symbols /t "CrashDumpLocal01"
2. Starten und abstürzen lassen
3. EXE/PDB-Dateien löschen
3. Dump debuggen (hier mal mit VS!)
.sympath+ d:\adc2014\symbols
.reload
.ecxr
kM
➔ Verifizieren mit !lmi CrashDumpLocal01! Achtung: Cache, falls symfix+ gemacht wurde…

Advanced Developers Conference
Development for Professionals!

Symbole - SrcSrv

Jochen Kalmbach
Microsoft MVP VC++

- Weiteres Problem: Jetzt hab ich zwar meine EXE/DLL/PDB aber wo ist der passende Source?
- Source-Server
  - Verknüpfung von Symbolen mit Source-Code aus CVS-Systemen (Source-Indexing)
  - Unterstützt werden
    - Ab TFS2010 (per Default im Build-Template vorhanden)
    - Perforce
    - Visual Source Safe
    - CVS
    - SVN
    - (erweiterbar)
  - Siehe dazu auch „srcsrv\srcsrv.doc“ im WinDbg Verzeichnis



Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Advanced Developers Conference
Development for Professionals!

WinDbg – Handle Leaks

Jochen Kalmbach
Microsoft MVP VC++

- !htrace -enable
  - Aktiviert das Handle Logging und erstellt einen ersten Snapshot
- !htrace -snapshot
  - Erstellt einen neuen Snapshot
- !htrace -diff
  - Gibt den Unterschied zum letzten Snapshot aus

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

HandleLeak01:
➔ Open “Relase\HandleLeak01.exe”
g ➔ Damit die normalen Handles zuerst mal geladen werden
➔ Ansicht im Task-Manager (Handles; letzte Spalte)

BreakIn
!htrace –enable
Handle tracing enabled.
Handle tracing information snapshot successfully taken.
g
➔ Press key

➔ BreakIn
!htrace –diff
.open –a 0x0119a9e7 (addr) (➔ Muss man manchmal zweimal machen…)
➔ Datei wird geöffnet

Advanced Developers Conference
Development for Professionals!

gflags

Jochen Kalmbach
Microsoft MVP VC++

- Muss jeder kennen!
- Sehr nützlich
- Aktiviert „Flags“ im OS
- Mögliche Einstellungen
  - Erlaubt das setzen von „Flags“, welche im Windows abgefragt werden
  - System: Systemweit (auch nach dem Reboot)
  - Kernel: Nur im laufenden Kernel
  - Image: Nur für diesen Prozess(name)
  - Silent Process Exit

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

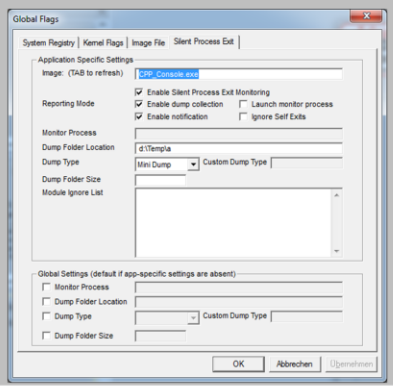
System: HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\GlobalFlag
Image: HKLM\SOFTWARE\Microsoft\Windows NT\Current Version\Image File Execution Options\<Image File Name>\GlobalFlag

Advanced Developers Conference
Development for Professionals!

gflags: Silent Process Exit

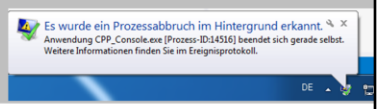
Jochen Kalmbach
Microsoft MVP VC++

- Loggt „unerwartetes“ Beenden von Prozessen mit



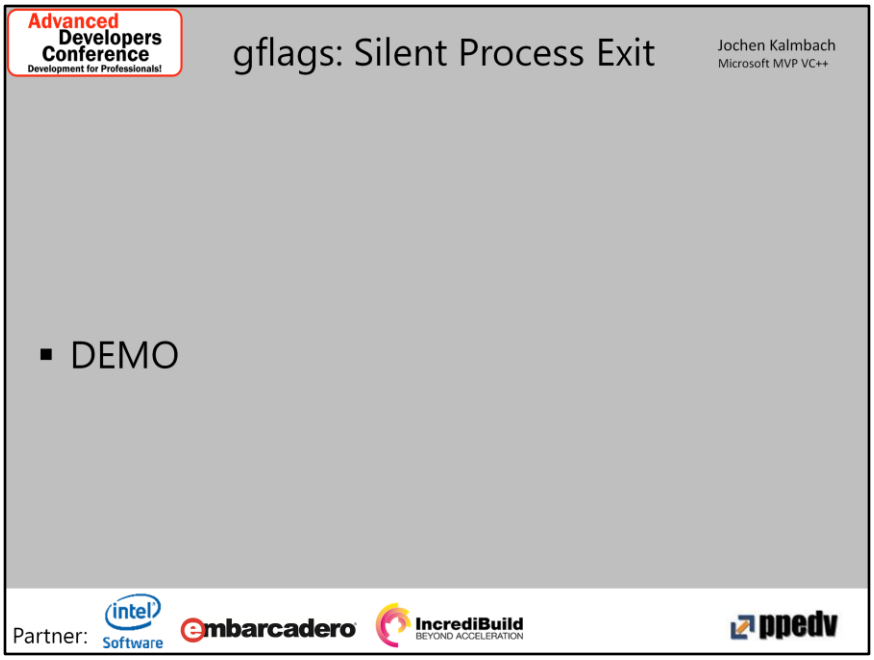
Benachrichtigung / Dump und Prozess-Start möglich

Bei „Cross-Process-Exits“ wird auch der 2. Prozess gedumpt



Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Beispiel: Fremd-DLL ruft irgendwo „ExitProcess“ auf…
„IgnoreSelfExist“: Wenn „an“, dann reagiert es nur auf „Cross-Process-Exists“!



SilentExit:

- Gflags konfigurieren "SilentExit.exe« / nur Enable und Dump
- Prozess starten /

Ausgabe in:

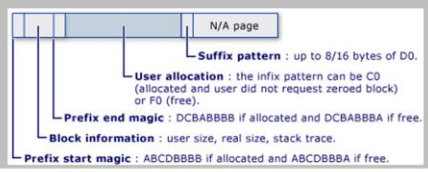
- C:\Users\ex7002t\AppData\Local\Temp\Silent Process Exit\
- Man kann auch Pfade angeben, dann muss man darauf achten, dass es auch dort Schreibberechtigungen gibt!

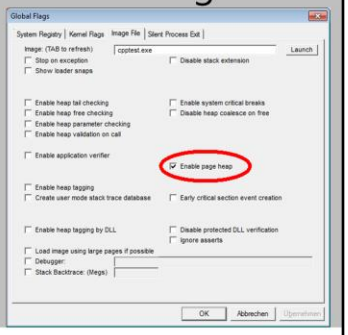
Advanced Developers Conference
Development for Professionals!

gflags: Heap-Overwrite

Jochen Kalmbach
Microsoft MVP VC++

- Einfache Anwendung via Gflags
- Aktivierung von PageHeap
- ➔ Alle Heap-Allocs werden an eine Page-Boundary gelegt
- ➔ Wesentlich mehr Speicherverbrauch





Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

The Structure of a Page Heap Block
http://msdn.microsoft.com/en-us/library/ms220938

HeapOverwrite01

- Erster Versuch: Ausführen als *Release* ➔ Fehler wird erst im Free festgestellt
- Dann Beenden und:
- Aktivieren in den gflags
    - UI: „Enable Page Heap“ / gflags /p /enable HeapOverwrite01.exe /full
- Ausführen in VS als *Release*! Hält im Debugger an der passenden Stelle!
- Anschauen des allokierten Blocks in VS-Memory-Window (padding / bytes usw.)
- Wenn nur die Padding-Bytes überschrieben werden, so wird dies (leider) auch erst beim free festgestellt.
- Hinweis: Wegen Alignment wird der Block i.d.R. passend allokiert; also es bleiben oft auch Padding-Bytes übrig

➔ Info: Einstellungen sind für x86/x64 verschieden!

Advanced Developers Conference
Development for Professionals!

Gflags / umdh: Leaks

Jochen Kalmbach
Microsoft MVP VC++

- UMDH: user-mode dump heap utility
- Gibt alle Allokationen im Heap aus
- Benötigt bestimmte „glflags“
- Muss aktiviert werden
  1. gflags aktivieren (gflags -i Name.exe +ust)
  2. Prozess starten
  3. Erster „Snapshot“ erstellen (umdh -pn:Name.exe -f:1st.txt)
  4. (Prozess erzeugt leaks)
  5. Zweiter Snapshot erstellen (umdh -pn:Name.exe -f:2nd.txt)
  6. Differenz bilden (umdh 1st.txt 2nd.txt -f:Diff.txt)
     - Erst hier werden Debug-Symbole benötigt!

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Demo:
0. Global Flags (x86) srtarten / gflags -i HeapLeak01.exe +ust ➔ UI: Image-Tag: HeapLeak01.exe "Create user mode stack trace database"!
3. cmd starten (auf dem Desktop), dann nach „D:" wechseln, damit es schreibbar ist!
4. HeapLeak01 starten
4. Erster Dump: c:umdh -pn:HeapLeak01.exe -f:First.txt
5. Taste im Programm drücken
6. Zweiter Dump: c:umdh -pn:HeapLeak01.exe –f:Second.txt
7. Symbole aufsetzen
set _NT_SYMBOL_PATH=SRV*C:\Users\ex7002t\AppData\Local\Temp\SymbolCache*http://msdl.microsoft.com/download/symbols;D:\ADC2014\HeapLeak01\Release
8. Vergleichen: c:umdh First.txt Second.txt -f:Diff.txt
- Datei „diff.txt" anschauen
9. Reset settings: gflags -i HeapLeak01.exe FFFFFFFF

Advanced Developers Conference
Development for Professionals!

Logger / LogViewer / .logexts

Jochen Kalmbach
Microsoft MVP VC++

- Logger
  - GUI App um den Logger zu starten
  - Empfohlener Weg
- .logexts
  - Ermöglich das Logging innerhalb von WinDbg
  - Kann sehr viele Ausgaben erzeugen und das System zum erliegen bringen
- LogViewer
  - Zum Anzeigen der Logdateien (.lgv)

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Notepad nehmen…
- Cmd auf dem Dektop öffnen
- „logger notepad.exe“ aufrufen
- „Start“ drücken

Ausgabe landet immer im „Desktop\LogExts“!
- Programm wieder beenden
- Logviewer in der Cmd starten
- Datei vom Desktop öffnen!

WinDbg: !logexts.help / !loge

Automatisierung

Jochen Kalmbach
Microsoft MVP VC++

- Alle Dinge lassen sich auch mit „cdb“ (Kommandozeilen-Debugger) ausführen
- Kann auch in „Batch-Dateien“ verwendet werden
- Beispiele:
  - Automatisches Auswertung vom Dump-Dateien
  - Dumps bei „AppHangs“ erzeugen

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

D:\ADC2014\Automation

D:\ADC2014\Automation\CLRApp

- Im „Windows SDK“ enthalten
- HiVersionLie verwenden:

- Zuerst die HiVersionLie aktivieren im AppVerify (Add App…);

- Dann auch noch „Dangerous functions“! ➔ Im Debugger ausführen…

Advanced Developers Conference
Development for Professionals!

Ausblick

Jochen Kalmbach
Microsoft MVP VC++

- Nützliche Tools
  - Application Verifier (auch im Windows SDK enthalten)
  - PowerDbg: PowerShell-Schnittstelle für WinDbg
    - http://powerdbg.codeplex.com/
  - SOS: Managed Debugging in WinDbg
    - Im .NET enthalten
  - SxsTrace: Side-By-Side Tracing
    - Im Windows enthalten

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Advanced Developers Conference
Development for Professionals!

FRAGEN?

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv

Advanced Developers Conference
Development for Professionals!

Ich freue mich auf Ihr Feedback!

Vortrag und Demos unter:
http://blog.kalmbach-software.de/

Vielen Dank!

Jochen Kalmbach

Partner: intel Software embarcadero IncrediBuild BEYOND ACCELERATION ppedv