Kostenloses eBook

LERNEN WinDbg

Free unaffiliated eBook created from
Stack Overflow contributors.

#windbg

Inhaltsverzeichnis

Über

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: windbg

It is an unofficial and free WinDbg ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official WinDbg.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit WinDbg

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was windbg ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in windbg erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für windbg neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Versionen

Wichtige Versionen von WinDbg für unterstützte Versionen von WinDbg. Siehe auch eine detaillierte Liste mit historischen Versionen online.

Es ist wichtig zu beachten, dass das Versionsschema von älterer Version 6.12 auf die neuere Version 6.1 geändert wird. Die älteren Versionen haben niedrige Zahlen (<100) an dritter Stelle, während neuere Versionen hohe Nummern (> 6000) haben.

In vielen Fällen funktionieren WinDbg-Versionen für neuere Windows-Versionen weiterhin unter älteren Versionen von Windows, z. B. kann Version 10 von WinDbg weiterhin unter Windows 7 verwendet werden. Allerdings können einige Befehle API-Aufrufe verwenden, die nicht verfügbar sind und daher fehlschlagen. Daher ist es gut, mehrere Versionen von WinDbg zur Verfügung zu haben.

Ausführung	Beschreibung	Veröffentlichungsdatum
6.12.0002.633	für Windows 7 und .NET Framework 4 bereitgestellt	2010-05-21
6.1.7600.16385		2009-07-24
6.2.8400.0	Update für Windows 8 (?)	2012-06-23
6.2.9200.16384	für Windows 8 und .NET Framework 4.5 bereitgestellt	2012-11-15
6.3.9600.16384	bereitgestellt für Windows 8.1	2013-10-17
10.0.10075.9	bereitgestellt für Windows 10	2015-04-29
10.0.10586.567	bereitgestellt seit Windows 10, Build 1511	2015-10-30
10.0.14321.1024	bereitgestellt seit Windows 10, Build 1607	2016-07-29

Examples

Installation oder Setup

Microsoft beschreibt drei Möglichkeiten , WinDbg zu installieren:

- als Teil des WDK (Windows Driver Kit)
- als Teil des SDK (Software Development Kit)
- mit dem Installer des SDK und der Auswahl aller anderen Elemente außer "Debugging Tools for Windows"

Um das Installationsprogramm zu erhalten, besuchen Sie WDK, WinDbg und die zugehörigen Tools herunterladen und scrollen Sie zu einem Abschnitt mit dem Titel "Debugging-Tools abrufen ".

Eine bekannte und praktische, aber inoffizielle Quelle ist Codemachine, wo Sie auch ältere Versionen der Debugging-Tools direkt herunterladen können.

Das Setup selbst ist unkompliziert. Klicken Sie durch das Installationsprogramm, bis der Vorgang abgeschlossen ist.

Debugger

WinDbg wird häufig als Abkürzung für "Debugging-Tools für Windows" verwendet. Es enthält verschiedene Debugger:

Debugger	Beschreibung
WinDbg	der Debugger mit einer grafischen Benutzeroberfläche
CDB	**c** onsole **d** e **b** ugger, Benutzermodus-Debugger, der in der aktuell geöffneten Konsole ausgeführt wird
NTSD	**n** ew tw **e** n **d** **e** m m **e** m **e** m **e** m **e** nb **e** n **d** **e** n **d** e **n d** e **n d** e **n d** e **n d** e **n d** e **n d** e **n d** e m **e** m **e** m **e** m **e** m **e** m **e** m bole, ein Benutzermodus-Debugger, der ein neues Terminal (Konsole) öffnet, wie der Name vermuten lässt
KD	Der **K d** Ernel ebugger, die in der offenen currrently Konsole läuft
NTKD	**n** ew **t k** erminal Ernel **d** ebugger, öffnet sich ein neues Terminal

Die Befehle sind identisch, außer dass möglicherweise GUI-bezogene Befehle vorhanden sind, die in den Konsolenversionen nicht funktionieren.

Erste Schritte mit WinDbg online lesen: https://riptutorial.com/de/windbg/topic/1833/erste-schritte-mit-windbg

Kapitel 2: Benutzermodus / Anwendungs-Debugging

Examples

Wichtige Befehle

Ihre Arbeit dokumentieren

Denken Sie daran, was Sie getan haben, und behalten Sie lange Ausgaben bei, die nicht im Puffer von WinDbg gespeichert werden können. Es ist immer gut, ein Protokoll für die Reproduktion von Debugging-Schritten zur Verfügung zu haben, z. B. um Fragen zu Stack Overflow zu stellen.

Befehl	Zweck
`.logopen`	Erstellen Sie eine Protokolldatei
`.logclose`	Schließen Sie die Protokolldatei
`.dump`	Speicherabbilddatei speichern (Momentaufnahme der aktuellen Debugsitzung)

Mit Symbolen arbeiten

Ohne oder mit falschen Symbolen erhalten Sie möglicherweise falsche Informationen und werden in die Irre geführt. Stellen Sie sicher, dass Sie mit diesen Befehlen vertraut sind, bevor Sie mit der Arbeit in WinDbg beginnen. Siehe auch So richten Sie Symbole in WinDbg ein .

Befehl	Zweck
`.symfix`	Festlegen oder Hinzufügen von Symbolen zum offiziellen Microsoft-Symbolpfad
`.sympath`	eigene Symbole oder Symbole von Drittanbietern setzen oder hinzufügen
`.reload`	Symbole neu laden
`.symopt`	Definieren Sie Optionen für die Symbolverarbeitung
`!sym`	Steuersymbol laden
`x`	Untersuche die Symbole
`ln`	Liste der nächstgelegenen Symbole

Crash-Analyse

Finden Sie heraus, was passiert ist (in Absturzabbildern) und wie Ereignisse behandelt werden sollen (im Live-Debugging).

Befehl	Zweck
`.exr`	Ausnahme-Datensatz anzeigen
`.lastevent`	letztes Ereignis anzeigen
`sx`	Ausnahmebehandlung definieren
`!analyze`	einen Absturz analysieren oder hängen lassen
`!avrf`	Anwendungsüberprüfung

Die Umgebung

Überprüfen Sie den Prozessnamen und die Versionsinformationen.

Befehl	Zweck
| (Rohr)	Prozessinformationen
`lm`	Modulliste

Threads, Call Stacks, Register und Speicher

Überprüfen Sie die Details.

Befehl	Zweck
`~`	Thread-Liste
`r`	registriert
`k`	Aufrufstack
`d` *	Speicher anzeigen
`e` *	Speicher bearbeiten
`s`	Speicher suchen
`.formats`	zwischen Zahlenformaten konvertieren

Befehl	Zweck
`?`	Ausdruck bewerten
`u` *	zerlegen
`a`	montieren
`!address`	Speicherinformationen

Ziel kontrollieren

Übernehmen Sie im Live-Debugging die Ausführung.

Befehl	Zweck
`g`	weiter / weiter
`gu`	geh hinauf
`p`	Einzelner Schritt
`t`	Trace (Einzelschritt- und Ausgangsregister)
`bp`	Haltepunkt setzen
`bl`	Haltepunktliste

Mit Erweiterungen arbeiten

Erweiterungen können erhebliche Vorteile und Verbesserungen bieten.

Befehl	Zweck
`.load`	Ladeerweiterung (vollständiger Pfad)
`.loadby`	Lastverlängerung relativ zum Modul
`.chain`	geladene Erweiterungen anzeigen
`.unload`	Erweiterung entladen

Stoppen Sie das Debuggen

Befehl	Zweck
`q`	Anwendung beenden und beenden

Befehl	Zweck
`qd`	lösen und beenden

Anbringen und abnehmen

Befehl	Zweck
`.tlist`	Prozessliste
`.attach`	an Prozess anhängen
`.create`	Einen Prozess erstellen und anhängen
`.childdbg`	Debugging-Verhalten für untergeordnete Prozesse definieren
`.detach`	Trennen Sie sich von einem Prozess
`.kill`	töte einen Prozess
`.restart`	Starten Sie den Prozess erneut

Verhalten von WinDbg

Befehl	Zweck
`.prefer_dml`	Debugger-Markup-Sprachverarbeitung festlegen
`.effmach`	die bitness wechseln

Usability-Befehle

Befehl	Zweck
`.cmdtree`	Lädt eine Textdatei mit vordefinierten Befehlen in einem separaten Fenster

Hilfe bekommen

Befehl	Zweck
`.hh`	Zeigt das Hilfehandbuch für WinDbg-Befehle an

Benutzerdefiniertes Befehlsfenster in Windbg erstellen

Mit `.cmdtree` Befehl `.cmdtree` können Sie eine `.txt` Datei mit vordefinierten Befehlen öffnen, die Sie einfach per Doppelklick ausführen können.

Wie erstellt man eine Befehlsdatei?

Erstellen Sie die Datei mit dieser Vorlage

```
windbg ANSI Command Tree 1.0
title {"Window title"}
body
{"Group Heading"}
 {"Name of command to display"} {"command"}
 {"Name of command to display"} {"command"}
{"Group Heading"}
 {"Name of command to display"} {"command"}
```

Dinge zu beachten

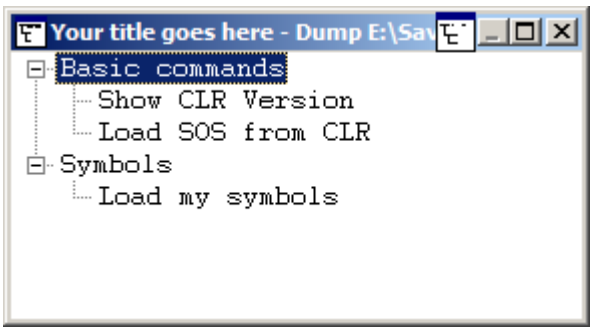
1. Das Vorlagenformat sollte genau eingehalten werden, um die Datei in Windbg zu öffnen.
2. Die Newline ist nach jedem `{Group Heading}` erforderlich.
3. Jedes `{Name of command to display} {command}` sollte sich in einer Zeile befinden und von einer neuen Zeile gefolgt werden.

Beispiel für eine benutzerdefinierte Befehlsdatei

```
windbg ANSI Command Tree 1.0
title {"Your title goes here"}
body
{"Basic commands"}
 {"Show CLR Version"} {"lmv m clr"}
 {"Load SOS from CLR"} {".loadby sos clr "}
{"Symbols"}
 {"Load my symbols"} {".sympath+ "c:\DebugSymbols" ; .reload"}
```

So öffnen Sie die Befehlsoberfläche aus dem Befehlsfenster

`.cmdtree <path of your .txt file>` , um das Fenster zu öffnen. Sie werden ein solches Fenster sehen



Doppelklicken Sie auf den auszuführenden Befehl.

Benutzermodus / Anwendungs-Debugging online lesen:
https://riptutorial.com/de/windbg/topic/5384/benutzermodus---anwendungs-debugging

Kapitel 3: Crash-Analyse

Examples

Grundlegende Absturzanalyse im Benutzermodus

`.exr -1` enthält Details zu der zuletzt ausgelösten Ausnahme.

`!analyze -v` normalerweise auch gute Arbeit.

Für .NET zeigt der Befehl `!pe` der SOS-Erweiterung Details zu der geworfenen .NET-Ausnahme.

Crash-Analyse online lesen: https://riptutorial.com/de/windbg/topic/5389/crash-analyse

Kapitel 4: DML (Debugger Mark Language)

Examples

An / Ausschalten

.prefer_dml 1 dmlformat-Ausgabe aktivieren

.prefer_dml 0 Deaktiviert die Ausgabe im dmlformat

DML (Debugger Mark Language) online lesen: https://riptutorial.com/de/windbg/topic/7987/dml--debugger-mark-language-

Kapitel 5: Erweiterungen

Examples

SOS

SOS (Son of Strike) ist die offizielle WinDbg-Erweiterung von Microsoft für .NET. Es wird als Teil des .NET-Frameworks installiert und ist daher standardmäßig verfügbar.

Wie jede Erweiterung kann es mit `.load x:\full\path\to\sos.dll` , es gibt jedoch einfachere Möglichkeiten. Abhängig von der .NET-Version befindet sich die Erweiterung neben `mscorwks.dll` (.NET CLR 2), `clr.dll` (.NET CLR 4) oder `coreclr.dll` (Silverlight- und Universal-Apps) Folgende Befehle sollten funktionieren:

```
.loadby sos clr
.loadby sos coreclr
.loadby sos mscorwks
```

Eine Liste der verfügbaren Befehle finden Sie in `!help` .

SOSex

SOSex ist eine Erweiterung von SOS, geschrieben von Steve Johnson , einem Mitarbeiter von Microsoft. Er stellt SOSex kostenlos zum Download zur Verfügung, ist jedoch nicht Open Source.

Normalerweise ist die Erweiterung nicht neben einer anderen DLL verfügbar. `.load x:\full\path\to\sosex.dll` wird sie normalerweise mit `.load x:\full\path\to\sosex.dll` .

Der Befehl `!dlk` vereinfacht nicht nur das Debuggen von .NET, `!dlk` kann auch in nativen Umgebungen zum Überprüfen von Deadlocks kritischer Abschnitte verwendet werden.

Eine Liste der verfügbaren Befehle finden Sie in der `!help` von SOSex `!help`

PyKD

PyKD ist eine WinDbg-Erweiterung, mit der Sie Python-Skripts schreiben können. Es ist Open Source.

Normalerweise ist die Erweiterung nicht neben einer anderen DLL verfügbar. `.load x:\full\path\to\pykd.pyd` wird sie normalerweise mit `.load x:\full\path\to\pykd.pyd` , wobei PYD die Erweiterung für eine Python-DLL ist. Sie können sie jedoch umbenennen es zu DLL wenn Sie mögen.

Erste Schritte mit PyKd

PyKD bietet keine `!help` -Hilfe an, also lesen Sie die Dokumentation bei Codeplex. Viele

Entwickler scheinen aus Russland zu stammen und die aktuellste und vollständigste Dokumentation ist wahrscheinlich in russischer Sprache. Der Google-Übersetzer macht einen anständigen Job.

Verwenden Sie wie andere Erweiterungen die korrekte Bitgröße der Erweiterung, die der von WinDbg entspricht. Außerdem müssen Sie Python mit der gleichen Bitness installiert haben.

`!py` führt einen REPL-Interpreter aus und `!py x:\path\to\script.py` führt ein Python-Skript aus. Skripte sollten verwenden

```
from pykd import *
```

als erste Zeile, um die PyKD-Funktionalität zu nutzen, während diese Zeile im REPL-Interpreter nicht benötigt wird. Der Interpreter kann mit `exit()` .

NetExt

NetExt ist eine Erweiterung für .NET, die zur Verfügung stellt

- LINQ-ähnliche Abfragen für Objekte auf dem Heap (`!wselect !wfrom`)
- Anzeigefunktionen für spezielle Objekte wie Wörterbücher und Hashtabellen (`!wdict !whash`)
- ASP.NET / HTTP-bezogene Befehle (`!wcookie !wruntime !whttp !wruntime !whttp`)
- mehrere andere netzwerkbezogene Befehle

Normalerweise ist die Erweiterung nicht neben einer anderen DLL verfügbar, daher wird sie normalerweise mit .load x: \ full \ path \ to \ netext.dll geladen

Erweiterungen im Überblick

Eine unvollständige Liste von WinDbg-Erweiterungen, die nicht mit WinDbg selbst installiert werden:

Erweiterung	Zweck
SOS	.NET (offizielle Microsoft-Erweiterung)
SOSex	.NET (Erweiterung für SOS)
CoSOS	.NET (Erweiterung für SOS)
NetExt	.NET (mit Fokus auf Vernetzung)
PyKD	Python-Skripting
PDE	Windows native und Store-Anwendungen (verstaute Ausnahmen)
PSSCOR	.NETZ
SDbgExt	.NETZ

Erweiterung	Zweck
MEX	.NETZ

CoSOS

Cosos (Cousin von SOS) ist ein Open - Source - Erweiterung für WinDbg wobei der Schwerpunkt auf .NET Speicherfragmentierung (`!gcview`) und Threadingprobleme (`!wfo` , `!tn`).

Normalerweise ist die Erweiterung nicht neben einer anderen DLL verfügbar. `.load x:\full\path\to\cosos.dll` wird sie normalerweise mit `.load x:\full\path\to\cosos.dll` . Es ist erforderlich, dass SOS geladen ist und derzeit nur mit 32-Bit-Anwendungen arbeitet.

Erweiterungen online lesen: https://riptutorial.com/de/windbg/topic/5391/erweiterungen

Kapitel 6: Kernel-Debugging

Examples

Wichtige Befehle

- ! process - liste Benutzermodusprozesse auf
- Prozess - Prozesskontext festlegen
- ! peb - Prozessumgebungsblock anzeigen
- ! teb - Thread-Umgebungsblock anzeigen
- ! Schlösser - Deadlock-Analyse
- .dump - Speichern Sie eine Crash-Dump-Datei auf der Festplatte

Kernel-Debugging online lesen: https://riptutorial.com/de/windbg/topic/6076/kernel-debugging

Kapitel 7: Remote-Debugging

Examples

Wichtige Befehle

- .server - Erstellen Sie einen Debug-Server
- .clients - Liste der Debugging-Clients, die mit dem Server verbunden sind
- .endsrv - Beendet einen Debug-Server
- .servers - Liste der Debugging-Serververbindungen
- .remote - Starten Sie einen remote.exe-Server
- .noshell - verhindert Shell-Befehle

Remote-Debugging online lesen: https://riptutorial.com/de/windbg/topic/5977/remote-debugging

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit WinDbg	Community, Thomas Weller
2	Benutzermodus / Anwendungs-Debugging	Piyush Parashar, Thomas Weller, X. Liu
3	Crash-Analyse	Thomas Weller
4	DML (Debugger Mark Language)	Wang Zhengzhang
5	Erweiterungen	Jason Evans, Lieven Keersmaekers, Thomas Weller
6	Kernel-Debugging	Thomas Weller
7	Remote-Debugging	Thomas Weller