EBook Gratuito

APPRENDIMENTO WinDbg

Free unaffiliated eBook created from
Stack Overflow contributors.

#windbg

Sommario

Di

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

Capitolo 1: Iniziare con WinDbg

Osservazioni

Questa sezione fornisce una panoramica di cosa è windbg e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di windbg e collegarsi agli argomenti correlati. Poiché la documentazione di windbg è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Versioni

Versioni importanti di WinDbg, per le versioni supportate di WinDbg. Vedi anche un elenco dettagliato con versioni storiche online.

È importante notare che c'è una modifica dello schema di versione dalla versione 6.12 alla versione 6.1 più recente. Le versioni precedenti hanno numeri bassi (<100) in terzo luogo mentre le versioni più recenti hanno numeri alti (> 6000).

In molti casi, le versioni di WinDbg fornite per le versioni di Windows più recenti funzionano ancora su versioni precedenti di Windows, ad esempio la versione 10 di WinDbg può ancora essere utilizzata su Windows 7. Tuttavia, alcuni comandi possono utilizzare chiamate API che non sono disponibili e quindi non riescono. Pertanto è bene avere diverse versioni di WinDbg disponibili.

Versione	Descrizione	Data di rilascio
6.12.0002.633	fornito per Windows 7 e .NET Framework 4	2010-05-21
6.1.7600.16385		2009-07-24
6.2.8400.0	aggiornamento per Windows 8 (?)	2012-06-23
6.2.9200.16384	fornito per Windows 8 e .NET Framework 4.5	2012/11/15
6.3.9600.16384	fornito per Windows 8.1	2013/10/17
10.0.10075.9	fornito per Windows 10	2015/04/29
10.0.10586.567	fornito da Windows 10, build 1511	2015/10/30
10.0.14321.1024	fornito da Windows 10, build 1607	2016/07/29

Examples

Installazione o configurazione

Microsoft descrive 3 modi di installare WinDbg:

- come parte del WDK (Windows Driver Kit)
- come parte dell'SDK (Software Development Kit)
- con l'installer dell'SDK e deselezionando tutto tranne "Debugging Tools for Windows"

Per ottenere il programma di installazione, visita Scarica il WDK, WinDbg e gli strumenti associati e scorri verso il basso fino a una sezione chiamata "Ottieni strumenti di debug".

Una fonte ben nota e comoda ma non ufficiale è Codemachine in cui è anche possibile scaricare direttamente le versioni precedenti degli strumenti di debug.

L'installazione stessa è diretta. Fare clic sul programma di installazione fino al termine.

debugger

WinDbg viene spesso utilizzato come abbreviazione di "Debugging tools for Windows". Contiene diversi debugger:

Debugger	Descrizione
WinDbg	il debugger con un'interfaccia grafica utente
CDB	**c** onsole **d** e **b** ugger, debugger in modalità utente che viene eseguito nella console attualmente aperta
NTSD	**n** ew **t** erminal **s** ymbolic **d** ebugger, debugger in modalità utente che apre un nuovo terminale (console) come suggerisce il nome
KD	il **k** ernel **d** ebugger, che viene eseguito nella console open currrently
NTKD	**n** ew **t** erminal **k** ernel **d** ebugger, apre un nuovo terminale

I comandi sono identici, tranne che potrebbero esserci comandi relativi alla GUI che non funzionano nelle versioni della console.

Leggi Iniziare con WinDbg online: https://riptutorial.com/it/windbg/topic/1833/iniziare-con-windbg

Capitolo 2: Analisi di crash

Examples

Analisi di crash in modalità utente di base

`.exr -1` fornisce dettagli sull'ultima eccezione generata.

`!analyze -v` solito fa anche un buon lavoro.

Per .NET, il comando `!pe` dell'estensione SOS mostra i dettagli sull'eccezione .NET che è stata generata.

Leggi Analisi di crash online: https://riptutorial.com/it/windbg/topic/5389/analisi-di-crash

Capitolo 3: Debug del kernel

Examples

Comandi importanti

- ! process - elenca i processi in modalità utente
- .processo - imposta il contesto del processo
- ! peb - mostra il blocco dell'ambiente di processo
- ! teb - mostra il blocco dell'ambiente del thread
- ! lock - analisi deadlock
- .dump: salva un file di dettagli di arresto anomalo sul disco

Leggi Debug del kernel online: https://riptutorial.com/it/windbg/topic/6076/debug-del-kernel

Capitolo 4: Debug di modalità utente / applicazione

Examples

Comandi importanti

Documentare il tuo lavoro

Ricorda cosa hai fatto e mantieni output lunghi che non possono essere tenuti nel buffer di WinDbg. È sempre utile disporre di un registro per la riproduzione dei passaggi di debug, ad esempio per porre domande su Stack Overflow.

Comando	Scopo
`.logopen`	crea un file di registro
`.logclose`	chiudi il file di registro
`.dump`	salva il file di crash dump (snapshot della sessione di debug corrente)

Lavorare con i simboli

Senza o con simboli errati, potresti ricevere informazioni errate e essere fuorviato. Assicurati di avere familiarità con questi comandi prima di iniziare a lavorare su WinDbg. Vedi anche Come configurare i simboli in WinDbg .

Comando	Scopo
`.symfix`	imposta o aggiungi simboli al percorso dei simboli Microsoft ufficiale
`.sympath`	imposta o aggiungi simboli propri o di terze parti
`.reload`	ricarica i simboli
`.symopt`	definire le opzioni di gestione dei simboli
`!sym`	caricamento del simbolo di controllo
`x`	esaminare i simboli
`ln`	elenca i simboli più vicini

Analisi di crash

Scopri cosa è successo (nei crash dump) e come gestire gli eventi (in live debugging).

Comando	Scopo
`.exr`	mostra record di eccezioni
`.lastevent`	mostra l'ultimo evento
`sx`	definire la gestione delle eccezioni
`!analyze`	analizzare un arresto anomalo o bloccarsi
`!avrf`	verificatore dell'applicazione

L'ambiente

Controlla il nome del processo e le informazioni sulla versione.

Comando	Scopo
| (tubo)	informazioni sul processo
`lm`	lista dei moduli

Discussioni, pile di chiamate, registri e memoria

Ispezionare i dettagli

Comando	Scopo
`~`	elenco di thread
`r`	registri
`k`	pila di chiamate
`d` *	mostra memoria
`e` *	modifica memoria
`s`	ricerca di memoria
`.formats`	convertire tra i formati numerici
`?`	valutare l'espressione

Comando	Scopo
`u` *	smontare
`a`	montare
`!address`	informazioni sulla memoria

Controllare l'obiettivo

Nel live debug, prendi il controllo dell'esecuzione.

Comando	Scopo
`g`	vai / continua
`gu`	vai su
`p`	singolo passo
`t`	trace (single step e registri di output)
`bp`	impostare il punto di interruzione
`bl`	elenco di punti di interruzione

Lavorare con le estensioni

Le estensioni possono offrire vantaggi e miglioramenti significativi.

Comando	Scopo
`.load`	carica estensione (percorso completo)
`.loadby`	caricare l'estensione relativa al modulo
`.chain`	mostra le estensioni caricate
`.unload`	scarica estensione

Ferma il debug

Comando	Scopo
`q`	chiudere e terminare l'applicazione
`qd`	staccare e uscire

Attaccare e staccare

Comando	Scopo
`.tlist`	lista dei processi
`.attach`	allegare al processo
`.create`	creare un processo e allegare
`.childddbg`	definire il comportamento di debug del processo figlio
`.detach`	staccare da un processo
`.kill`	uccidere un processo
`.restart`	riavviare il processo

Comportamento di WinDbg

Comando	Scopo
`.prefer_dml`	imposta la gestione del linguaggio di markup del debugger
`.effmach`	cambia il testimone

Comandi di usabilità

Comando	Scopo
`.cmdtree`	Carica un file di testo con comandi predefiniti in una finestra separata

Ottenere aiuto

Comando	Scopo
`.hh`	Visualizza il manuale della guida per i comandi WinDbg

Crea una finestra di comando personalizzata in Windbg

Il comando `.cmdtree` consente di aprire un file `.txt` con comandi predefiniti che puoi semplicemente fare doppio clic per eseguire.

Come creare un file di comando

Crea il file usando questo modello

```
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"}
```

Cose da curare

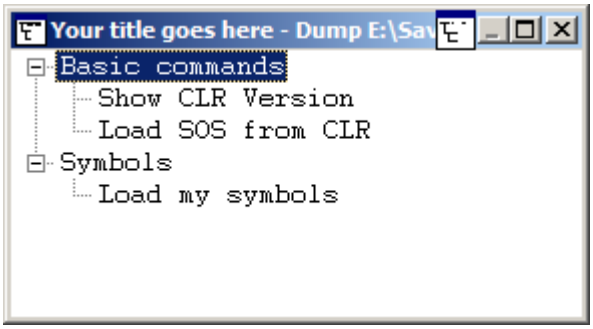
1. Il formato del modello deve essere seguito precisamente per aprire il file in Windbg.
2. La newline è richiesta dopo ogni `{Group Heading}` .
3. Ogni `{Name of command to display} {command}` coppia dovrebbe essere in una riga e dovrebbe essere seguito da una nuova riga.

Esempio di file di comando personalizzato

```
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"}
```

Come aprire l'interfaccia utente del comando dalla finestra di comando

Esegui `.cmdtree <path of your .txt file>` per aprire la finestra. Vedrai una finestra come questa



Fare doppio clic sul comando da eseguire.

Leggi Debug di modalità utente / applicazione online:
https://riptutorial.com/it/windbg/topic/5384/debug-di-modalita-utente---applicazione

Capitolo 5: Debug remoto

Examples

Comandi importanti

- .server: crea un server di debug
- .clients - elenca i client di debug collegati al server
- .endsrv - termina un server di debug
- .servers - elenca le connessioni al server di debugging
- .remote - avvia un server remote.exe
- .noshell - impedisce i comandi della shell

Leggi Debug remoto online: https://riptutorial.com/it/windbg/topic/5977/debug-remoto

Capitolo 6: DML (linguaggio dei segni del debugger)

Examples

Accendi / spegni

.prefer_dml 1 attiva l'output di dmlformat

.prefer_dml 0 disattiva l'output di dmlformat

Leggi DML (linguaggio dei segni del debugger) online:
https://riptutorial.com/it/windbg/topic/7987/dml--linguaggio-dei-segni-del-debugger-

Capitolo 7: estensioni

Examples

SOS

SOS (figlio di sciopero) è l'estensione ufficiale di WinDbg di Microsoft per .NET. Viene installato come parte del framework .NET e quindi è disponibile per impostazione predefinita.

Come qualsiasi estensione, può essere caricato usando `.load x:\full\path\to\sos.dll` , ma ci sono modi più semplici. A seconda della versione di .NET, l'estensione si trova affiancata a `mscorwks.dll` (.NET CLR 2), `clr.dll` (.NET CLR 4) o `coreclr.dll` (Silverlight e Universal app), quindi una delle i seguenti comandi dovrebbero funzionare:

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

Per un elenco di comandi disponibili, consultare la guida `!help`

SOSex

SOSex è un'estensione di SOS, scritta da Steve Johnson , un dipendente Microsoft. Fornisce SOSex per il download gratuitamente, ma non è open source.

In genere, l'estensione non è disponibile affiancata a qualsiasi altra DLL, quindi viene solitamente caricata con `.load x:\full\path\to\sosex.dll` .

Oltre a semplificare il debug di .NET, il comando `!dlk` può essere utilizzato anche in ambienti nativi per il controllo di deadlock di sezioni critiche.

Per un elenco di comandi disponibili, consultare l' `!help` di SOSex.

PyKD

PyKD è un'estensione WinDbg che ti consente di scrivere script Python. È open source.

In genere, l'estensione non è disponibile affiancata a qualsiasi altra DLL, quindi viene solitamente caricata con `.load x:\full\path\to\pykd.pyd` , dove PYD è l'estensione per una DLL python, ma è possibile rinominare a DLL se ti piace.

Iniziare con PyKd

PyKD non offre `!help` , quindi consulta la documentazione su Codeplex. Molti sviluppatori sembrano provenire dalla Russia e la documentazione più aggiornata e completa è probabilmente in russo. Il traduttore di Google fa un lavoro decente.

Come le altre estensioni, utilizza la versione corretta dell'estensione corrispondente a quella di WinDbg. In aggiunta a questo devi avere Python installato con lo stesso bitness.

`!py` esegue un interprete REPL e `!py x:\path\to\script.py` esegue uno script python. Gli script dovrebbero usare

```
from pykd import *
```

come prima linea per poter utilizzare le funzionalità di PyKD, mentre questa linea non è necessaria nell'interprete REPL. L'interprete può essere chiuso usando `exit()` .

NetExt

NetExt è un'estensione per .NET che fornisce

- Query tipo LINQ per oggetti sullo heap (`!wselect !wfrom`)
- visualizzare le funzionalità per oggetti speciali come dizionari e tabelle hash (`!wdict !whash`)
- Comandi relativi a ASP.NET / HTTP (`!wcookie !wruntime !whttp`)
- molti altri comandi relativi alla rete

In genere, l'estensione non è disponibile affiancata a qualsiasi altra DLL, quindi viene solitamente caricata con .load x: \ full \ path \ to \ netext.dll

Panoramica delle estensioni

Un elenco incompleto di estensioni WinDbg che non sono installate con WinDbg stesso:

Estensione	Scopo
sos	.NET (estensione Microsoft ufficiale)
SOSex	.NET (estensione per SOS)
Cosos	.NET (estensione per SOS)
NetExt	.NET (con particolare attenzione alla rete)
PyKD	Scripting Python
PDE	Windows native e store applications (eccezioni stivate)
PSSCOR	.NETTO
SDbgExt	.NETTO
MEX	.NETTO

Cosos

Cosos (cugino di SOS) è un'estensione open source per WinDbg concentrandosi sulla frammentazione di memoria .NET (`!gcview`) e problemi di threading (`!wfo` , `!tn`).

In genere, l'estensione non è disponibile affiancata a qualsiasi altra DLL, quindi viene solitamente caricata con `.load x:\full\path\to\cosos.dll` . Richiede che SOS sia caricato e attualmente funziona solo con applicazioni a 32 bit.

Leggi estensioni online: https://riptutorial.com/it/windbg/topic/5391/estensioni

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con WinDbg	Community, Thomas Weller
2	Analisi di crash	Thomas Weller
3	Debug del kernel	Thomas Weller
4	Debug di modalità utente / applicazione	Piyush Parashar, Thomas Weller, X. Liu
5	Debug remoto	Thomas Weller
6	DML (linguaggio dei segni del debugger)	Wang Zhengzhang
7	estensioni	Jason Evans, Lieven Keersmaekers, Thomas Weller