



eBook Gratuit

APPRENEZ debugging

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#debugging

Table des matières

À propos	1
Chapitre 1: Démarrer avec le débogage	2
Remarques.....	2
Exemples.....	2
Exemple de chaîne d'outils de débogage pour le débogage en mode utilisateur sous Windows.....	2
Chapitre 2: Comprendre les codes d'erreur / de sortie du système sous Windows	3
Remarques.....	3
Exemples.....	3
Conversion de codes de sortie en messages significatifs.....	3
Crédits	4

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [debugging](#)

It is an unofficial and free debugging 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 debugging.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

Chapitre 1: Démarrer avec le débogage

Remarques

Cette section fournit une vue d'ensemble de ce qu'est le débogage et des raisons pour lesquelles un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les sujets importants dans le débogage et établir un lien avec les sujets connexes. La documentation pour le débogage étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Exemple de chaîne d'outils de débogage pour le débogage en mode utilisateur sous Windows

- Visual Studio (IDE)
- [WPA](#) (analyseur de performance)
- [WinDbg](#) (débogueur)
- [IDA Pro](#) (démonteur)
- [dotPeek](#) (décompilateur pour .NET)
- [WinMerge](#) (outil de diff)
- [Éditeur HxD](#) ou [010](#) (éditeur hexadécimal)
- [Speedcrunch](#) (calculatrice)
- [Firefox](#) (navigateur)
- [Moniteur API Rohitab](#) (surveillance des appels API)
- SOS (une extension WinDbg pour .NET)
- [SOSex](#) (une autre extension pour WinDbg)
- [Agent Ransack](#) (recherche de contenu de fichier)
- [WSCC](#) (collection d'outils SysInternals et Nirsoft)
- [Debug Diag](#) (outil de diagnostic)
- [Vérificateur d'application](#) (outil d'analyse d'exécution)
- [Dependency Walker](#) (outil d'analyse des dépendances DLL)
- [Stud_PE](#) (outil d'analyse de fichier PE)

Lire [Démarrer avec le débogage en ligne](https://riptutorial.com/fr/debugging/topic/5312/demarrer-avec-le-debogage): <https://riptutorial.com/fr/debugging/topic/5312/demarrer-avec-le-debogage>

Chapitre 2: Comprendre les codes d'erreur / de sortie du système sous Windows

Remarques

Pour que ce processus fonctionne, le code d'erreur / de sortie d'origine doit commencer par `0x8007` ce qui `0x8007` généralement qu'il provient d'un processus Win32 valide.

Cependant, si aucun message n'apparaît, il ne provient probablement pas d'un processus Windows et devra donc être examiné en dehors des étapes mentionnées ci-dessus.

Exemples

Conversion de codes de sortie en messages significatifs

Être capable de comprendre les codes d'erreur / de sortie est une compétence fondamentale pour les développeurs sur la machine Windows. Cependant, pour beaucoup, le code hexadécimal crypté pouvant être produit sur une application sortant avec une erreur peut s'avérer long et fastidieux pour que le développeur puisse le retrouver et l'isoler.

Par exemple, sur SO, il y a plusieurs milliers de questions portant toutes sur la signification de ce que signifie un code d'erreur / de sortie particulier ... et par exemple, un code de sortie de ce type

Le programme '[4432] program.exe' est sorti avec le code -2147023895 (0x800703e9).

Par conséquent, pour identifier la cause du problème, nous devons convertir le code de sortie / erreur en quelque chose de plus significatif et nous pouvons le faire en procédant comme suit.

1. À partir du code d'erreur `0x800703e9`, prenez les 4 derniers caractères `03e9`
2. En utilisant un [convertisseur hexadécimal en décimal](#), convertissez `03e9` en son équivalent décimal, qui dans ce cas est `1001`
3. En utilisant `cmd`, tapez `net helpmsg 1001` ou la valeur décimale renvoyée à l'étape 3.
4. Un message d'erreur convivial devrait apparaître pour aider à identifier la cause du problème. Dans ce cas, l'erreur renvoyée était `Recursion too deep; the stack overflowed.`

Lire [Comprendre les codes d'erreur / de sortie du système sous Windows en ligne](#):

<https://riptutorial.com/fr/debugging/topic/6146/comprendre-les-codes-d-erreur---de-sortie-du-systeme-sous-windows>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec le débogage	Community , Daniel Nugent , Thomas Weller
2	Comprendre les codes d'erreur / de sortie du système sous Windows	timkly