



EBook Gratis

APRENDIZAJE debugging

Free unaffiliated eBook created from
Stack Overflow contributors.

#debugging

Tabla de contenido

Acerca de.....	1
Capítulo 1: Comenzando con la depuración.....	2
Observaciones.....	2
Examples.....	2
Ejemplo de cadena de herramientas de depuración para la depuración en modo usuario en Wind.....	2
Capítulo 2: Entendiendo los códigos de error / salida del sistema en Windows.....	3
Observaciones.....	3
Examples.....	3
Convertir códigos de salida en mensajes significativos.....	3
Creditos.....	4

Acerca de

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

Capítulo 1: Comenzando con la depuración

Observaciones

Esta sección proporciona una descripción general de qué es la depuración y por qué un desarrollador puede querer usarla.

También debe mencionar cualquier tema importante dentro de la depuración y vincular a los temas relacionados. Dado que la Documentación para la depuración es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Ejemplo de cadena de herramientas de depuración para la depuración en modo usuario en Windows

- Visual Studio (IDE)
- [WPA](#) (analizador de rendimiento)
- [WinDbg](#) (depurador)
- [IDA Pro](#) (desensamblador)
- [dotPeek](#) (descompilador para .NET)
- [WinMerge](#) (herramienta de diferencias)
- [Editor HxD](#) o [010](#) (editor hexadecimal)
- [Speedcrunch](#) (calculadora)
- [Firefox](#) (navegador)
- [Monitor API de Rohitab](#) (monitoreo de llamadas API)
- SOS (una extensión de WinDbg para .NET)
- [SOSex](#) (otra extensión para WinDbg)
- [Agente Ransack](#) (búsqueda de contenido de archivo)
- [WSCC](#) (colección de herramientas SysInternals y Nirsoft)
- [Debug Diag](#) (herramienta de diagnóstico)
- [Verificador de aplicación](#) (herramienta de análisis de tiempo de ejecución)
- [Caminante de dependencia](#) (herramienta de análisis de dependencia de DLL)
- [Stud_PE](#) (herramienta de análisis de archivos PE)

Lea [Comenzando con la depuración en línea](#):

<https://riptutorial.com/es/debugging/topic/5312/comenzando-con-la-depuracion>

Capítulo 2: Entendiendo los códigos de error / salida del sistema en Windows

Observaciones

Para que este proceso funcione, el código de error / salida original debe comenzar con `0x8007` que generalmente es una indicación de que se originó a partir de un proceso Win32 válido.

Sin embargo, si no aparece ningún mensaje, probablemente no se originó en un proceso de Windows y, por lo tanto, será necesario examinarlo más allá de los pasos mencionados anteriormente.

Examples

Convertir códigos de salida en mensajes significativos.

Ser capaz de comprender los códigos de error / salida es una habilidad fundamental para los desarrolladores en la máquina de Windows. Sin embargo, para muchos, el código hexadecimal críptico que se puede producir en una aplicación que se produce con un error puede resultar lento y un proceso minucioso para que el desarrollador pueda localizarlo y aislarlo.

Por ejemplo, en SO, hay varios miles de preguntas que preguntan sobre el significado de un código de error / salida en particular ... y como ejemplo, a continuación se encuentra uno de esos códigos de salida.

El programa '[4432] program.exe' ha salido con el código -2147023895 (0x800703e9).

Por lo tanto, para identificar la causa del problema, necesitamos convertir el código de salida / error en algo más significativo, y podemos hacerlo realizando el siguiente proceso.

1. Del código de error `0x800703e9`, tome los últimos 4 caracteres `03e9`
2. Usando un [convertidor de hexadecimal a decimal](#), convierta `03e9` a su contraparte decimal, que en este caso es `1001`
3. Usando `cmd`, escriba `net helpmsg 1001` o cualquier valor decimal que se devuelva desde el paso 3.
4. Debe aparecer un mensaje de error amistoso que puede ayudar a identificar la causa del problema, que en este caso, el error devuelto fue `Recursion too deep; the stack overflowed.`

Lea [Entendiendo los códigos de error / salida del sistema en Windows en línea](#):

<https://riptutorial.com/es/debugging/topic/6146/entendiendo-los-codigos-de-error---salida-del-sistema-en-windows>

Creditos

S. No	Capítulos	Contributors
1	Comenzando con la depuración	Community , Daniel Nugent , Thomas Weller
2	Entendiendo los códigos de error / salida del sistema en Windows	timkly