



We use optional cookies to improve your experience on our websites, such as through social media connections, and to display personalized advertising based on your online activity. If you reject optional cookies, only cookies necessary to provide you the services will be used. You may change your selection by clicking "Manage Cookies" at the bottom of the page. [Privacy Statement](#) [Third-Party Cookies](#)

Accept

Reject

Manage cookies



Collecting User-Mode Dumps


Article • 07/19/2024 • 11 contributors [Feedback](#)

Starting with **Windows Server 2008** and **Windows Vista with Service Pack 1 (SP1)**, Windows Error Reporting (WER) can be configured so that full user-mode dumps are collected and stored locally after a user-mode application crashes. Applications that do their own custom crash reporting are not supported by this feature.

This feature is not enabled by default. Enabling the feature requires administrator privileges. To enable and configure the feature, use the following registry values under the **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps** key.

Expand table

Value	Description	Type
DumpFolder	The path where the dump files are to be stored. If you do not use the default path, then make sure that the folder contains ACLs that allow the crashing process to write data to the folder. For service crashes, the dump is written to service-specific profile folders depending on the service account used. For example, the profile folder for System services is %WINDIR%\System32\Config\SystemProfile. For Network and Local Services, the folder is %WINDIR%\ServiceProfiles.	REG_EXPAND_SZ
DumpCount	The maximum number of dump files in the folder. When the maximum value is exceeded, the oldest dump file in the folder will be replaced with the new dump file.	REG_DWORD
DumpType	Specify one of the following dump types: <ul style="list-style-type: none">• 0: Custom dump• 1: Mini dump• 2: Full dump	REG_DWORD
CustomDumpFlags	The custom dump options to be used. This value is used only when DumpType is set to 0. The options are a bitwise combination of the MINIDUMP_TYPE enumeration values.	REG_DWORD 0x00000012 (MiniDumpWriteMemory) MiniDumpWriteMemory MiniDumpWriteMemory == 0x00000012 0x00000010

 **Note**

A crash dump is not collected when you set [automatic debugging for application crashes](#).

These registry values represent the global settings. You can also provide per-application settings that override the global settings. To create a

per-application setting, create a new key for your application under **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Windows Error Reporting\LocalDumps**

(for example, **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Windows Error Reporting\LocalDumps\MyApplication.exe**). Add your dump settings under the **MyApplication.exe** key. If your application crashes, WER will first read the global settings and then override any of the settings with your application-specific settings.

After an application crashes and prior to its termination, the system will check the registry settings to determine whether a local dump is to be collected. After the dump collection has been completed, the application will be allowed to terminate normally. If the application supports recovery, the local dump is collected before the recovery callback is called.

These dumps are configured and controlled independently of the rest of the WER infrastructure. You can make use of the local dump collection even if WER is disabled or if the user cancels WER reporting. The local dump can be different than the dump sent to Microsoft.

More information

Task Manager enhancements

Windows 11 includes a new feature in Task Manager that allows users to create live memory dumps for both kernel and user-mode processes. This can be done by navigating to the Processes or Details tab, right-clicking the desired process, and selecting **Create live memory dump file**. This feature simplifies the process of capturing memory dumps directly from the Task Manager interface. See [Task Manager live memory dump](#) for more information.

ProcDump improvements

The Sysinternals ProCDump utility has been enhanced to support various new options for creating dumps, such as triggering dumps on thread creation or exit, using specific performance counters, or capturing dumps of hung windows. ProCDump in Windows 11 supports all trigger types introduced in Windows 8.1 and later. For more details, see [ProCDump v11.0](#).

Debugging enhancements

Windows 11 supports advanced debugging features with tools like WinDbg and CDB, which allow for detailed analysis of both full and minidump files. These tools have been updated to better handle the nuances of user-mode dumps in Windows 11, including the ability to read dump files directly from CAB files and to analyze multiple dump files simultaneously. Learn more: [Analyze crash dump files by using WinDbg](#).

 **Note:** The author created this article with assistance from AI. [Learn more](#)


Feedback

Was this page helpful?



 Yes

 No

[Provide product feedback](#)  | [Get help at Microsoft Q&A](#)

 English (United States)

  Your Privacy Choices


 Theme 

[Manage cookies](#)


[Previous Versions](#)

[Blog](#) 

[Contribute](#)

[Privacy](#) 

[Terms of Use](#)

[Trademarks](#) 

© Microsoft 2024