

# What does it mean when my attempt to stop a Windows NT service fails with `ERROR_BROKEN_PIPE`?

[devblogs.microsoft.com/oldnewthing/20190405-00](https://devblogs.microsoft.com/oldnewthing/20190405-00)

April 5, 2019



Raymond Chen

A customer reported that they had a sporadic problem: Their product includes a Windows NT service, and when their client program tries to stop the service, it sometimes fails with `ERROR_BROKEN_PIPE`. Their client program is written in C#, so it uses the `ServiceController.Stop` method to stop the service, and the failure is reported in the form of an exception. In Win32, this turns into a call to the `ControlService` function with the `SERVICE_CONTROL_STOP` code.

Under what conditions would an attempt to stop a service result in the error `ERROR_BROKEN_PIPE`?

One of the developer support escalation engineers used psychic powers:

Does your service terminate itself before the call to its `HandlerEx` routine returns from the `SERVICE_CONTROL_STOP` request, or before the call to `StartServiceCtrlDispatcher` returns?

I'm guessing that the `ERROR_BROKEN_PIPE` arises because the service process terminated itself while the Service Control Manager was still talking to it, waiting for the service to report that it finished processing the `SERVICE_CONTROL_STOP` request. The error is `ERROR_BROKEN_PIPE` because the process on the other end of the pipe (the service) died.

The customer agreed that this was a possibility: When the service receives the `SERVICE_CONTROL_STOP` request, it signals a helper thread to clean up, and that helper thread may finish its cleanup and terminate the service process before the main thread can report a successful stop to the Service Control Manager.

A short time later, the customer reported back and confirmed that when they forced the race condition to occur, they indeed got the `ERROR_BROKEN_PIPE` error code.

I like this example of psychic debugging because it demonstrates how you can take something you know (`ERROR_BROKEN_PIPE` means that two processes were talking to each other over a pipe, and one side suddenly terminated), and think about how it could apply to

something you don't know (surmising that the Service Control Manager uses a pipe to talk to the service).

Raymond Chen

**Follow**

