Registry-Free Profiler Startup and Attach

By Archiveddocs

Archived: 2026-04-05 23:46:21 UTC

Updated: January 2012

Starting with the .NET Framework version 4, you can attach profilers both at application startup time (startup loading) and while applications are running (attach loading). (Before the .NET Framework 4, startup loading was the only way to load a profiler.) Both options provide the ability to start up profilers that have not been registered as COM components.

Startup loading and attach loading use different protocols for starting profilers and are discussed in the following sections.

A startup-load profiler is loaded when the application to be profiled starts. The profiler is registered through the value of the following environment variable:

- COR_ENABLE_PROFILING=1

Starting with the .NET Framework 4, you use either the COR_PROFILER or the COR_PROFILER_PATH environment variable to specify the location of the profiler. (Only COR_PROFILER is available in earlier versions of the .NET Framework.)

- COR_PROFILER={CLSID of profiler}
- COR_PROFILER_PATH=full path of the profiler DLL

If COR_PROFILER_PATH is not present, the common language runtime (CLR) uses the CLSID from COR_PROFILER to locate the profiler in the HKEY_CLASSES_ROOT of the registry. If COR_PROFILER_PATH is present, the CLR uses its value to locate the profiler and skips registry lookup. (However, you still have to set COR_PROFILER, as discussed in the following list of rules.)

Starting with the .NET Framework 4, the rules for loading a profiler at startup are as follows:

- You must set COR_ENABLE_PROFILING to 1. If COR_ENABLE_PROFILING is absent or is set to 0 (zero), the runtime system will not connect the process to the profiler.
- You must set the COR_PROFILER environment variable to the CLSID of the profiler, even if you specify COR_PROFILER_PATH, because the runtime still has to pass the CLSID of the profiler to the CreateInstance call for the class factory.
- If you specify COR_PROFILER_PATH, the CLR uses the path you specify, even if it is invalid, and does not check the registry to find the path. If you specify an invalid path, the profiler is not loaded.

- If COR_PROFILER_PATH is not present, the CLR uses the CLSID specified by COR_PROFILER to find the full path to your profiler DLL in the registry. As with any COM server DLL, the CLR looks for the CLSID under HKEY_CLASSES_ROOT, which merges the classes from HKEY_LOCALE_MACHINE and HKEY_CURRENT_USER.

Starting with the .NET Framework 4, you can attach a profiler to a running application. For more information about this functionality, see Profiler Attach and Detach. An attach-load profiler uses the wszProfilerPath parameter of the ICLRProfiling::AttachProfiler method to find the location of the profiler DLL file. If wszProfilerPath is null, the runtime tries to locate the profiler by looking in the registry for the CLSID that is specified in the pClsidProfiler parameter.

Profiling Overview

Profiling (Unmanaged API Reference)

Unmanaged API Reference

Date	History	Reason
January 2012	Clarified information.	Information enhancement.
