

Processes and Threads - Win32 apps

By Karl-Bridge-Microsoft

Archived: 2026-04-05 14:06:50 UTC

An application consists of one or more processes. A *process*, in the simplest terms, is an executing program. One or more threads run in the context of the process. A *thread* is the basic unit to which the operating system allocates processor time. A thread can execute any part of the process code, including parts currently being executed by another thread.

A *job object* allows groups of processes to be managed as a unit. Job objects are namable, securable, sharable objects that control attributes of the processes associated with them. Operations performed on the job object affect all processes associated with the job object.

A *thread pool* is a collection of worker threads that efficiently execute asynchronous callbacks on behalf of the application. The thread pool is primarily used to reduce the number of application threads and provide management of the worker threads.

A *fiber* is a unit of execution that must be manually scheduled by the application. Fibers run in the context of the threads that schedule them.

User-mode scheduling (UMS) is a lightweight mechanism that applications can use to schedule their own threads. UMS threads differ from [fibers](#) in that each UMS thread has its own thread context instead of sharing the thread context of a single thread.

- [What's New in Processes and Threads](#)
- [About Processes and Threads](#)
- [Using Processes and Threads](#)
- [Process and Thread Reference](#)

Source: <https://docs.microsoft.com/en-us/windows/win32/procthread/processes-and-threads>