



# Tech Info Library

## System 7.5: Thread Manager Overview (11/95)

Revised: 2/13/96  
Security: Everyone

System 7.5: Thread Manager Overview (11/95)

Article Created: 02 August 1994  
Article Reviewed/Updated: 29 November 1995

TOPIC -----

This article provides an overview of the Thread Manager component of System 7.5.

DISCUSSION -----

### Introduction

The Thread Manager is an addition to the Macintosh Toolbox that enables concurrent programming for true multitasking within an application. The Thread Manager supports cooperative and preemptive threads of execution at the application level, not at the system level.

The Thread Manager has the same rights and privileges as other services in the Macintosh Toolbox. The Thread Manager works on all Macintosh platforms running System 7 or later.

### Thread Manager Overview

The Thread Manager has two types of threads: cooperative and preemptive. Additionally, the Thread Manager provides the basic services for creating, scheduling, and deleting threads and for gathering thread status.

### Thread

A thread is a sequence of instructions executed in a program. A thread has a program counter and a stack to keep track of local variables and return addresses. A multithreaded process is associated with one or more threads.

Threads share process instructions and most of their data. Threads that share the process state have considerable interaction; it is possible for them to affect each other because there is no system-enforced protection between

threads.

Threads also share the open files; each thread sees the same open files. If one thread opens a file, another thread can read it. If shared data changes, it can be seen by other threads in the process.

#### Cooperative Threads

Cooperative threads allow cooperative multitasking. They are the most effective to use for scheduling and accessing Toolbox traps. Cooperative threads have the same capability as a standard application when working with memory allocation, file I/O, QuickDraw, and other resources. Operationally, cooperative threads yield to other cooperative threads only when the application explicitly makes one of the Thread Manager yield calls or changes the state of the current cooperative thread.

Cooperative threading in the Thread Manager works similar to the cooperative threading in the earlier Threads Package. The advantage of using the Thread Manager is that thread stacks are register-swapped during a context switch.

#### Preemptive Threads

Preemptive threads allow true multitasking at the application level. When an application gets control from the Process Manager, preemptive threads for that application are allowed to run.

Preemptive threads differ from cooperative threads because they can interrupt the currently executing thread at any time to resume execution. Preemptive threads can be designed to yield back to the just-preempted cooperative thread using a Thread Manager yield call or to wait for its time quantum to expire.

If the interrupted cooperative thread is in the stopped state, the next available preemptive thread is made to run. Preemptive threads then preempt each other in a round-robin fashion until the interrupted cooperative thread is made ready.

#### Data Type

Data type is an identifier within the Thread Manager that provides the identity and status of the thread. There are several data types in the Thread Manager. Data types determine the type of thread (cooperative or preemptive), the state of a thread (running, ready, or stopped), and a set of options for creating new threads.

#### The Main Thread

When an application launches with the Thread Manager installed, the Thread Manager automatically defines the application entry point as the main cooperative thread. The main cooperative thread is commonly referred to as the application thread or main thread.

#### End of the Thread

When threads finish executing, the Thread Manager automatically disposes of them. Preemptive threads are recycled to avoid moving memory. Recycling a thread causes the Thread Manager to clear out the internal data structure, reset it, and place the thread in the proper thread pool. (There are two pools: one for cooperative and one for preemptive threads.) Since recycling a thread doesn't move memory, preemptive threads can dispose of themselves and other threads safely.

#### Requirements

-----  
The Thread Manager is part of the Macintosh Toolbox. Thread Manager requires System 7 or later.

#### Article Change History:

29 Nov 1995 - Added keyword, reviewed for technical accuracy.

#### Support Information Services

Copyright 1994-95, Apple Computer, Inc.

Tech Info Library Article Number:15963