Installing and Activating Task Records

The following listing shows how to install and activate a **Time Manager** task. It assumes that the procedure MyTask has already been defined; see the next two examples after this one for examples of simple task definitions.

```
// Installing and activating a Time Manager task
// Assuming inclusion of <MacHeaders>

void InstallTMTask (void);
pascal void MyTask (void);

void InstallTMTask ()
{
    TMTask     myTMTask;     // an extended task record
    long     myDelay;   // delay value

    myDelay    = 2000;    // no. of milliseconds to delay
    myTMTask.tmAddr      = MyTask; // get task address
    myTMTask.tmWakeUp= 0;     // initialize tmWakeUp
    myTMTask.tmReserved    = 0;     // initialize tmReserved
    InsXTime((QElemPtr) &myTMTask);  // install the task record
    PrimeTime((QElemPtr)
    &myTMTask,   // activate the task record
    myDelay); //
}
```

In this example, InstallTMTask installs an extended task record into the **Time** Manager queue and then activates the task. After the specified delay has elapsed (in this case, 2000 milliseconds, or 2 seconds), the procedure MyTask executes.

In cases where no task is to run after the specified time delay has elapsed, you should set the tmAddr field to NIL. To determine if the time has expired, you can check the task-active bit in the qType field.

Calling **PrimeTime** on a **Time Manager** task record that has not yet expired yields unpredict-able results and should therefore be avoided. If a prior unexpired request exists in the **Time Manager** queue that you wish to reactivate for some different delay, you should call **RmvTime** to cancel the prior request, then call **InsTime** to reinstall the timer task, and finally call **PrimeTime** to reschedule the task. Note, however, that it is possible and sometimes desirable to call **PrimeTime** in a **Time Manager** task that you want to reactivate, because the timer will have expired before the task is called.