Data Access Manager Routines

The **Data Access Manager** has high-level routines, low-level routines, and routines that manipulate result handlers. All of the low-level routines and some of the high-level routines have as a parameter a pointer to an asynchronous parameter block. If you specify a nonzero value for this parameter, the database extension executes the function asynchronously-that is, it returns control to the **Data Access Manager** before the routine has completed execution, and the **Data Access Manager** returns control to your application. If you specify NIL for the pointer to the asynchronous parameter block, the database extension does not return control to your application until the routine has finished execution.Your application must call the **WaitNextEvent** function periodically to allow an asynchronous routine to complete execution. The **WaitNextEvent** function is described in the **Event Manager** .

You can tell when an asynchronous routine has completed execution and check the result code by looking at values in the asynchronous parameter block. The asynchronous parameter block is described in the section, **Asynchronous Execution of Routines**.

> **Note:** A noErr result code returned by a routine that has been called asynchronously indicates only that the routine *began* execution successfully. You must check the result field of the asynchronous parameter block for the final result of the routine.

> **Assembly-language note:** You can invoke each of the **Data Access Manager** routines with a macro that has the same name as the routine, but preceded with an underscore; for example, the macro for the **DBInit** function is named _DBInit. Each of these macros places a routine selector in the D0 register and calls the trap _Pack13.