Writing and Installing Coercion Handlers

When your application extracts data from a parameter, it can request that the **Apple Event Manager** return the data using a descriptor type that is different from the original descriptor type. For example, when extracting data from the direct parameter of the Open Documents event, you can request that the alias records be returned as file system specification records. The **Apple Event Manager** can automatically coerce many different types of data from one to another.

You can also provide your own routines, referred to as coercion handlers, to coerce other descriptor types. To install your own coercion handlers, use the **AEInstallCoercionHandler** function. You specify as parameters to this function

- the descriptor type of the data coerced by the handler
- the descriptor type of the resulting data
- the address of the coercion handler for this descriptor type
- a reference constant
- a Boolean value that indicates whether your coercion handler expects the data to be specified as a descriptor record or as a pointer to the actual data
- a Boolean value that indicates whether your coercion handler should be added to the application coercion table or the system coercion table.

You can provide a coercion handler that expects to receive the data in a descriptor record or a buffer referred to by a pointer. When you install your coercion handler, you specify how your handler wishes to receive the data. It's more efficient for the **Apple Event Manager** to provide your coercion handler with a pointer to the data so, whenever possible, you should write your coercion handler so that it can accept a pointer to the data.

A coercion handler that accepts a pointer to data must be a function with the following syntax:

```
OSErr MyCoercePtr (DescType typeCode, Ptr dataPtr, Size dataSize,
                   DescType toType, long handlerRefcon,
                   AEDesc *result);
```

The *typeCode* parameter is the descriptor type of the original data. The *dataPtr* parameter is a pointer to the data to coerce; the *dataSize* parameter is the length, in bytes, of the data. The toType is the desired descriptor type of the resulting data. The *handlerRefcon* parameter is a reference constant that is stored in the coercion table entry for the handler and passed to the handler by the **Apple Event Manager** whenever the handler is called. The *result* parameter is the resulting descriptor record returned by your coercion handler.

Your coercion handler should coerce the data to the desired descriptor type and return the resulting data in the descriptor record specified by the *result* parameter. Your handler should return the noErr result code if your handler successfully performs the coercion, and a nonzero result code otherwise.

A coercion handler that accepts a descriptor record must be a function with the following syntax:

```
OSErr MyCoerceDesc (AEDesc *theAEDesc, DescType toType,
                    long handlerRefcon, AEDesc *result);
```

The parameter *theAEDesc* is the descriptor record that contains the data to be coerced. The *toType* parameter is the descriptor type of the resulting data. The *handlerRefcon* parameter is a reference constant that is stored in the coercion table entry for the handler and passed to the handler by the **Apple Event Manager** whenever the handler is called. The *result* parameter is the resulting descriptor record.

Your coercion handler should coerce the data in the descriptor record to the desired descriptor type and return the resulting data in the descriptor record specified by the *result* parameter. Your handler should return an appropriate result code.

Note: For many **Apple Event Manager** functions, the **Apple Event Manager** attempts to coerce data to the descriptor type you specify even if the result is no longer meaningful. To ensure that no coercion is performed and that the descriptor type of the result is of the same descriptor type as the original, specify typeWildCard for the desired type.