

## Create Publisher Event

If your application supports publish and subscribe capabilities, it should also handle the **Create Publisher** event.

-----	
<b>Create Publisher event</b>	create a publisher
Event class	<u>keyAEMiscStdSuite</u>
Event ID	<u>keyAECreatePublisher</u>
Required parameter	None
Keyword	<u>keyDirectObject</u>
<u>Descriptor type</u>	<u>typeObjectSpecifier</u>
Data	The part of the <u>document</u> to publish. If this parameter is omitted, publish the current selection.
Keyword	<u>keyAEEditionFileLoc</u>
<u>Descriptor type</u>	<u>typeAlias</u>
Data	An <u>alias record</u> that contains the location of the <u>edition</u> container to create. If this parameter is omitted, use the default <u>edition container</u> .
Requested action	Create a publisher for the specified data using the specified location for the <u>edition container</u> . If the data isn't specified, publish the current selection. If the location of the <u>edition</u> isn't specified, use the default location.
-----	

When your application receives the **Create Publisher event**, it should create a publisher by writing the publisher's data to an edition. The data of the publisher, and the location and name of the edition, are defined by the Apple event. If the **Create Publisher event** includes a keyDirectObject parameter, then your application should publish the data contained in the parameter. If the keyDirectObject parameter is missing, then your application should publish the current selection. If the document doesn't have a current selection, your handler for the event should return a nonzero result code.

If the **Create Publisher event** includes a keyAEEditionFileLoc parameter, then your application should use the location and name contained in the parameter as the default location and name of the edition. If the keyAEEditionFileLoc parameter is missing, then your application should use the default location and name your application normally uses to specify the edition container.

The following program shows a handler for the **Create Publisher event**. This handler checks for the keyDirectObject parameter and the keyAEEditionFileLoc parameter. If either of these is not specified, the handler uses default values. The handler uses the DoNewPublisher function, an application-defined function, to create the publisher and its edition, create a section record, and update other data structures associated with the document.

Note that the handler uses the AEInteractWithUser function to determine

whether user interaction is allowed. If user interaction is allowed, the handler sets the *promptForDialog* variable to TRUE, indicating that the DoNewPublisher function should display the publisher dialog box. If user interaction is not allowed, the handler sets the *promptForDialog* variable to FALSE, and the DoNewPublisher function does not prompt the user for the location or name of the edition.

```
// A handler for the Create Publisher event
```

```
// Assuming inclusion of <MacHeaders>
```

```
#include <Script.h>
#include <Editions.h>
#include <AERegistry.h>
#include <AEObjects.h>
#include <AppleEvents.h>
```

```
pascal OSErr MyHandleCreatePublisherEvent (AppleEvent *theAppleEvent,
                                           AppleEvent *reply, long handlerRefcon);
void    MyGetDocumentPtr (MyDocumentInfoPtr thisDocument);
Handle MyGetPreviewForSelection (MyDocumentInfoPtr
                                  thisDocument);
void    MySetInfoForPreview(AEDesc *thePublisherDataDesc,
                           MyDocumentInfoPtr thisDocument,
                           Handle preview, FormatType previewFormat);
void    DoError (OSErr myErr);
OSErr MyGetDefaultEditionSpec(MyDocumentInfoPtr thisDocument,
                              EditionContainerSpec defaultLocation);
OSErr MyGotRequiredParams (AppleEvent *theAppleEvent);
Boolean MyIdleFunction (EventRecord *theEventRecord,
                          long *sleepTime, RgnHandle *mouseRgn);
OSErr DoNewPublisher(MyDocumentInfoPtr thisDocument,
                     Boolean promptForDialog,
                     Handle preview,
                     FormatType previewFormat,
                     EditionContainerSpec defaultLocation);
```

```
pascal OSErr MyHandleCreatePublisherEvent (AppleEvent *theAppleEvent,
                                           AppleEvent *reply, long handlerRefcon)
{
    OSErr    myErr;
    DescType returnedType;
    AEDesc thePublisherDataDesc;
    long    actualSize;
    Boolean  promptForDialog;
    MyDocumentInfoPtr  thisDocument;
    Handle preview;
    FormatType  previewFormat;
    EditionContainerSpec defaultLocation;

    MyGetDocumentPtr(thisDocument);

    myErr = AEGetParamDesc(theAppleEvent, keyDirectObject,
```

```

        typeObjectSpecifier, &thePublisherDataDesc);
switch (myErr) {
case errAEDescNotFound:
    // use the current selection as the publisher and set up
    // info for later when DoNewPublisher displays preview
    preview = MyGetPreviewForSelection(thisDocument);
    previewFormat = 'TEXT';
    break;
case noErr:
    // use the data in keyDirectObject parameter as the
    // publisher (which is returned in the
    // thePublisherDataDesc variable), and set up info for
    // later when DoNewPublisher displays preview

    MySetInfoForPreview(&thePublisherDataDesc, thisDocument,
        preview, previewFormat);
    break;
default:
    DoError(myErr);
    break;
}

myErr = AEGGetParamPtr(theAppleEvent, keyAEEditionFileLoc,
                        typeFSS, &returnedType,
                        (Ptr)&defaultLocation.theFile,
                        sizeof(FSSpec), &actualSize);

switch (myErr) {
case errAEDescNotFound:
    // use the default location as the edition container
    myErr = MyGetDefaultEditionSpec(thisDocument,
        defaultLocation);

    break;
case noErr:
    // the keyAEEditionFileLoc parameter
    // contained a default location
    defaultLocation.thePart = kPartsNotUsed;
    defaultLocation.theFileScript = smSystemScript;
    break;
default:
    DoError(myErr);
    break;
}

myErr = MyGotRequiredParams(theAppleEvent);
if (myErr) {
    // handle the error appropriately
    return myErr;
}

myErr = AEInteractWithUser(kAEDefaultTimeout, nil,
                            &MyIdleFunction);

if (myErr == noErr)
    promptForDialog = TRUE;
else

```

```
promptForDialog = FALSE;  
  
myErr = DoNewPublisher(thisDocument, promptForDialog,  
    preview, previewFormat, defaultLocation);  
  
// add keyErrorNumber and keyErrorString parameters if desired  
return myErr;  
}
```

You can find a listing of **MyGotRequiredParams** may be found under **Writing Apple Event Handlers**. For more information on writing the **MyIdleFucntion**, see **Writing an Idle Function**.