CreateAEVT

Err:=CreateAEVT(Class;ID;Addr;theEvent)

Creates an AppleEvent of the specified class and ID with the specified target address. This doesn't add any parameters to the event and doesn't actually send it.

Example:

```
$err:=CreateAEVT("misc";"dosc";theTarget;$myEvent)
if ($err=0)
 $err:=PutTextParam($myEvent;"----";"message(“Hi there!”)")
 if ($err=0)
   $err:=SendAppleEvent($myEvent;$myReply; kAENoReply+CanInteract ;-1)
   $err:=DisposeDesc($myReply)
 end if
 $err:=DisposeDesc($myEvent)
end if
```

DisposeDesc

Err:=DisposeDesc(desc)

Disposes an AppleEvent or Target Address descriptor. This should always be called for an AppleEvent created with CreateAEVT and the reply created when the event is sent as well as a target address that is no longer needed.

GetAliasParam

GetAliasParam(theEvent;theKey;aFileName)

Extracts a file alias parameter from an AppleEvent. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle. The alias will be converted to a string representing the full pathname of the file.

GetList

Err:=GetList(anEvent;theKey;anArray)

Extracts a descriptor list from an AppleEvent into a 4D array. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle. All items in the list must be of the same type. The array will be created automatically of the correct type to hold the contents of the list. A list of lists or AERecords will be returned as an array of long integers, with each element containing a handle to a descriptor. The type of an existing array won’t be changed, and if you’re using an existing array, a list of strings requires a TEXT array. In a compiled database, the array must be previously defined and you must know the data type in advance.

GetLongParam

Err:=GetLongParam(anEvent;keyWord;actualType;Value)

Extracts a long integer parameter from an AppleEvent. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle.

GetPicParam

Err:=GetPicParam(anEvent;theKey;convert;aPicture)

Extracts a picture parameter from an AppleEvent. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle. If the picture came from another 4D application, 'convert' should be 0, otherwise it should be non-zero and will cause the picture to be converted from standard Macintosh format to 4D's internal picture format.

GetRealParam

Err:=GetRealParam(anEvent;theKey;Value)

Extracts a floating point number parameter from an AppleEvent. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle. The value can be any numeric type and will be coerced into 4D's 10-byte extended format.

GetShortParam

Err:=GetShortParam(anEvent;keyWord;actualType;Value)

Extracts a short integer parameter from an AppleEvent. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle.

GetTextParam

Err:=GetTextParam(anEvent;theKey;Value)

Extracts a text parameter from an AppleEvent. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle.

Obj

ospec:=Obj(class;container;value)

Creates an object specifier identified by a numeric value. Class should be the 4-letter class ID. Container can be either another object specifier or 0 to specify a null container. The container will automatically be disposed of to simplify creating nested containers. Value should be numeric. For example, to create the specifier 'Word 1 of Paragraph 2 of Document "Foo"' you would use:
Obj("cwor";Obj("cpar";ObjNamed("docu";0;"Foo");2);1)

ObjNamed

ospec:=ObjNamed(Class;Container;Name)

Creates an object specifier identified by a name. Class should be the 4-letter class ID. Container can be either another object specifier or 0 to specify a null container. The container will automatically be disposed of to simplify creating nested containers. Name should be a string. For example, to create the specifier 'Word 1 of Paragraph 2 of Document "Foo"' you would use:
Obj("cwor"; Obj("cpar"; ObjNamed("docu";0;"Foo");2);1)

Property

ospec:=Property(Name;Container)

Creates an object specifier for a property. Container should be the object specifier the property applies to. The container will automatically be disposed of to simplify creating nested containers. Name should be the 4-letter property ID. For example, to create the specifer "font of word 1" you would use: Property("font";Obj("cwor";0;1))

PutAliasParam

Err:=PutAliasParam(anEvent;theKey;fileName)

Adds a file alias to an AppleEvent. The parameter is identified by a keyword. The value should be a string representing a file name which will be converted to an alias record. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

New in V3.5: To specify an alias of a file on a remote volume, you can use a string such as “[Server@MyZone]HD80:A File”. The zone name can be eliminated if its in the same zone. You can also use a string like “[]HD80:some file” to create a minimal alias containing only the full path name. This is useful when sending an AppleEvent to a remote machine.

Example:

```
$err:=CreateAEVT("misc";"dosc";theTarget;$myEvent)
if ($err=0)
 $err:=PutAliasParam($myEvent;"----";"HD80:Scripts:Do this")
 if ($err=0)
```

```
   $err:=SendAppleEvent($myEvent;$myReply; kAENoReply ;-1)
   $err:=DisposeDesc($myReply)
 end if
 $err:=DisposeDesc($myEvent)
end if
```

PutList

Err:=PutList(anEvent;key;SpecialType;anArray)

Adds a descriptor list built from a 4D array to an AppleEvent. SpecialType can specify a data type other than the default implied by the type of the list. The only conversions are from text representing file names to a list of aliases and from long integer to AERecord or object specifiers. In all other cases the data format will remain unchanged. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

To specify text to alias conversion, SpecialType should be "alis".

To send a list of AERecords or object specifiers, SpecialType should be “reco” and the long integer array should contain descriptors created with Obj, ObjNamed, or CreateAERec. In this case, each element of the array will automatically be disposed when you call PutList.

New in Version 3.5: You can now use a 2-dimensional array with PutList.

Example:

```
Selection To Array([file1]field1;anArray)
$err:=CreateAEVT("send";"data";theTarget;$myEvent)
if ($err=0)
 $err:=PutList($myEvent;"----";"";anArray)
 if ($err=0)
   $err:=SendAppleEvent($myEvent;$myReply;kAEWaitReply;-1)
   $err:=DisposeDesc($myReply)
 end if
 $err:=DisposeDesc($myEvent)
end if
```

PutLongParam

Err:=PutLongParam(anEvent;key;actualType;value)

Adds a long integer parameter to an AppleEvent. The parameter is identified by a keyword and can be given a special type. (It defaults to typeLongInteger if 'actualType' is an empty string.) In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

PutObject

Err:=PutObject(anEvent;key;anObject)

Puts an object specifier in an AppleEvent. This command will dispose the object specifier after adding it to the AppleEvent. This command should be used in conjunction with Obj and/or ObjNamed to create the specifier. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

Example:

```
$err:=MakeAddress("DFB0";$addr)
if ($err=0)
 $err:=CreateAEVT("core";"getd";$addr;$myEvent)
 if ($err=0)
   $err:=PutObj($myEvent;"----";objNamed("cMPV";0;"theDate"))
   if ($err=0)
    $err:=SendAppleEvent($myEvent;$myReply;kAEWaitReply;-1)
    $err:=GetTextParam($myReply;"----";theResult)
    $err:=DisposeDesc($myReply)
   end if
   $err:=DisposeDesc($myEvent)
 end if
 $err:=DisposeDesc($addr)
end if
```

PutPicParam

Err:=PutPicParam(anEvent;theKey;convert;aPicture)

Adds a picture parameter to an AppleEvent. It will be identified by a keyword and will always be of type 'PICT'. If ‘convert’ is non-zero, the picture will be converted from 4D’s internal format to a standard Macintosh® picture. In 4D® version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

PutRealParam

Err:=PutRealParam(anEvent;theKey;Value)

Adds a floating point number to an AppleEvent. The number will be in 4D’s internal 10-byte floating point format and the descriptor type will be ‘exte’. Most applications will be able to convert it to their desired floating point type. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

PutShortParam

Err:=PutShortParam(anEvent;key;actualType;value)

Adds a short integer parameter to an AppleEvent. The parameter is identified by a keyword and can be given a special type. (It defaults to typeShortInteger if 'actualType' is an empty string.) In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

PutTextParam

Err:=PutTextParam(anEvent;key;value)

Adds a text parameter to an AppleEvent. The parameter is identified by a keyword and will always be of type 'typeText'. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

SendAppleEvent

Err:=SendAppleEvent(theEvent;theReply;send mode;time out)

Sends an AppleEvent previously created by CreateAEVT. SendMode can be any of the constants kAEWaitReply, kAENoReply, AlwaysInteract, CanInteract, and/or NeverInteract. TimeOut specifies the time (in 1/60ths of a second) to wait for a reply. Specify -1 for default time-out or -2 for no time- out.

CreateAERec

Err:=CreateAERec(theRecord)

Creates an AE Record, which is a list of keyword-specified parameters within an AppleEvent. All commands which add and extract parameters from an AppleEvent can be applied the same way to an AE Record. The AE Record must be disposed when you finish using it. Adding it to an AppleEvent or another AE Record with PutAERecord will automatically dispose it.

Example:

```
` this is how to send a request to Claris® Resolve™
` create the request range table
$Err:=CreateAERec($tabl)
$Err:=PutLongParam($tabl;"TLPT";"";100)
$Err:=PutLongParam($tabl;"BRPT";"";200)
$Err:=PutKeyword($tabl;"RTRN";"TEXT")
` create the record which holds the table
$Err:=CreateAERec($reqList)
$Err:=PutAERecord($reqlist;"TABL";$tabl)
` create the actual AppleEvent and add the record
$Err:=CreateAppleEvent("CLRS";"GVAL";$Resolve;$GetValue)
$Err:=PutAERecord($GetValue;"----";$reqList)
$Err:=SendAppleEvent($GetValue;$Reply;kAEWaitReply;-1)
` extract the data from the reply record
$Err:=GetAERecord($Reply;"----";$reqList)
$Err:=GetAERecord($reqList;"TABL";$tabl)
$Err:=GetTextParam($tabl;"VAL ";$resultString)
` clean up everything we allocated
```

```
$Err:=DisposeDesc($tabl)
$Err:=DisposeDesc($reqList)
$Err:=DisposeDesc($Reply)
$Err:=DisposeDesc($GetValue)
```

GetAERecord

Err:=GetAERecord(anEvent;Key;theRecord)

Extracts an AE Record from an AppleEvent or from another AE Record. Any commands which apply to AppleEvents for adding and/or extracting data can also be used on AE Records. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle.

PutAERecord

Err:=PutAERecord(anEvent;Key;theRecord)

Adds an AE Record to an AppleEvent or to another AE Record. The AE Record will automatically be disposed after this call. Note that function this is really the same as PutObject, since an object specifier is simply a special kind of AE Record. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

PutKeyword

Err:=PutKeyword(anEvent;key;value)

Adds a 4-character keyword (enumerated type) to an AppleEvent or AE Record. If you need to pass a type rather than an enumeration, use PutLongParam(aevt; key; ”type”; Long(value)) instead. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.

GetKeyword

Err:=GetKeyword(anEvent;key;value)

Extracts a 4-character enumerated type keyword from an AppleEvent or AE Record. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle.

GetBoolean

Err:=GetBoolean(anEvent;key;value)

Extracts a boolean value from an AppleEvent or AE Record. To reference the last event received in a procedure installed via HandleAEVT, pass 0 for the event handle.

PutBoolean

```
Err:=PutBoolean(anEvent;key;value)
```

Adds a boolean value to an AppleEvent or AE Record. In 4D version 3.0.3 or later, you can specify -1 for the AppleEvent to add an item to the reply event in a procedure called from HandleAEVT.