

Launching & Quitting

```
Err:=Launch(Signature;Document Name)
Err:=LaunchBehind(Creator;Document)
Err:=PrintDoc(Signature;Document Name)
Err:=QuitApp(Signature)
```

Note: A launch command doesn't complete until 4D calls the event manager. In pre-3.0 versions, the procedure must either end or call ProcessAEVT to complete the launch.

Misc. Utilities

```
Err:=BringToFront(Signature)
```

Note: BringToFront only requests that the application be brought to the front. It won't actually happen until 4D calls the event manager. In pre-3.0 versions, the procedure must either end or call ProcessAEVT to allow another application to be brought to the front.

```
Err:=FindAppName(Signature;Name)
Err:=FindCreator(File Name;Creator ID)
Result:=IsRunning(Creator ID)
Result:=Long(TypeString)
Err:=ProcessList(Visible;StringArray)
Version:=S7Version(Serialization String)
SetTimeout(Ticks)
Result:=System7
Result:=AE Process ID
```

AE Process ID will always return a non-zero value if you're running 4D v3.0 or later and will return 0 if you're running an earlier version.

SetTimeOut applies to all high-level scripting commands & high-level commands which send AppleEvents.

Target Addresses

```
Err:=AddrToString(Target;ATLocation;Name;Port)
Err:=DisposeAddress(Target)
Err:=MakeAddress(Signature;Target)
Err:=SelectAddress(Prompt;Signature;Target)
Err:=StringToAddr(ATLocation;Name;Port;Target)
```

Scripting

```
Err:=DoScript(Target;Command)
Err:=Evaluate(Target;Command;Result)
```

```
Err:=Frontier(Command;Result)
Err:=QuicKeys(Macro Name)
```

Sending AppleEvents

```
Err:=AESend(Target;Class;ID;Message)
Err:=AESendPict(target;class;id;aPicture)
Err:=SendWithReply(Target;Class;ID;message;reply)
```

AppleEvent Handling

```
AllowAccess(Flag)
GetAEMessage(Msg)
Err:=GetAEPict(aPicture)
GetAEType(theType)
GetReturnAddr(Target)
Err:=HandleAEVT(Class;ID;Command)
Err:=IgnoreAEVT(Class;ID)
ProcessAEVT [not needed in 4D v3.0]
```

Low Level Interface

```
Err:=CreateAEVT(Class;ID;Addr;theEvent)
Err:=DisposeDesc(desc)
Err:=GetAliasParam(theEvent;theKey;aFileName)
Err:=GetList(anEvent;theKey;anArray)
Err:=GetLongParam(anEvent;keyWord;actualType;Value)
Err:=GetPicParam(anEvent;theKey;convert;aPicture)
Err:=GetRealParam(anEvent;theKey;Value)
Err:=GetShortParam(anEvent;keyWord;actualType;Value)
Err:=GetTextParam(anEvent;theKey;Value)
ospec:=Obj(class;container;value)
ospec:=ObjNamed(Class;Container;Name)
ospec:=Property(Name;Container)
Err:=PutAliasParam(anEvent;theKey;fileName)
Err:=PutList(anEvent;key;SpecialType;anArray)
Err:=PutLongParam(anEvent;key;actualType;value)
Err:=PutObject(anEvent;key;anObject)
Err:=PutPicParam(anEvent;theKey;convert;aPicture)
Err:=PutRealParam(anEvent;key;value)
Err:=PutShortParam(anEvent;key;actualType;value)
Err:=PutTextParam(anEvent;key;value)
Err:=SendAppleEvent(theEvent;theReply;send mode;time out)
Err:=CreateAERec(theRecord)
Err:=GetAERecord(anEvent;Key;theRecord)
Err:=PutAERecord(anEvent;Key;theRecord)
Err:=PutKeyword(anEvent;key;value)
Err:=GetKeyword(anEvent;key;value)
```

To extract a boolean value, use GetShortParam(...;...;"bool";...)

To send a list of file aliases, use `PutList(...;...;"alis";...)`
To send a list of AE Records or object specifiers, use
`PutList(...;...;"reco";...)`

`GetList` will automatically convert a list of file aliases to a text array containing the file names or a list of `AERecords`, object specifiers or records to an array of long integers which represent descriptors.

Constants

`AlwaysInteract`
`CanInteract`
`NeverInteract`
`kAENoReply`
`kAEWaitReply`

These constants can be used to specify the send mode for `SendAppleEvent`. Use an interaction value + a reply value.

Advanced Commands

`Err:=CreateXAEVT(Class;id;addr;returnID;TransID;aevt)`
`Result:=CopyDesc(aevt)`
`Result:=GetTransactionID(aevt)`
`Result:=GetReturnID(aevt)`
`Err:=GetAEInfo(Descriptor;Size;Keys;Types;Lengths)`
`Err:=GetNthDesc(aevt,index,key,type,result)`
`Err:=GetNthItem(aevt,index,key,type,textVar)`

Use `CopyDesc(0)` to obtain the current event being handled.

Transaction IDs and Return IDs can be used to associate a queued reply with a specific event that was sent.

`GetAEInfo`, `GetNthDesc`, and `GetNthItem` can be used to examine an `AppleEvent` to determine what keywords are present.