Interfacing with Claris Resolve®

In addition to the required AppleEvents and DoScript, Claris Resolve® Supports the following custom events:

```
      CLRS
GVAL     Extracts data from the current worksheet.
  CLRS
PVAL     Puts data into the current worksheet.
```

Both of these events require a special record which must be created with the low-level functions. This record consists of top left and bottom right points, and a return type (for GVAL) or return value (for GVAL reply or PVAL) and is contained in another record.

```
  TABL     Record with the following fields:
    TLPT     typeInteger   Top left cell in the range


BRPT     typeInteger   Bottom Right cell of the range


VAL      typeEnum      Return type (should be “TEXT”)
```

The code to create and send such a record would look something like this:

```
` 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)
```

Interfacing with Microsoft Excel 4.0®

Excel® 4.0 supports most of the AppleEvent Registry (See Appendix D) and also allows any command or macro to be executed with the DoScript command.The most useful events are Set

Data, Get Data, and Create Element. You can use these events to create new worksheets or charts, fill-in a range of cells, extract the contents of a range of cells, produce various kinds of charts, and copy a chart into 4D® as a picture field.

NOTE: Commands sent using DoScript will be executed asynchronously in Excel® while the 4D procedure continues to run. If 4D tries to send additional commands while Excel is still executing the last command, it can cause unexpected results. To make sure the 4D procedure and Excel are properly synchronized, use SendWithReply(Excel;"misc";"dosc"...) instead of DoScript.

To execute an Excel macro via DoScript, you must send it as a function call, for example DoScript($excel;"'Macro1'!MyMacro()"). If you send only the macro name it will not be executed.

Example 1: Filling a range of cells from a 4D array

```
$err:=CreateAEVT("core";"setd";Excel;$aevt)
if ($err=0)

$err:=PutObject($aevt;"----"ObjNamed("crng";0;"R1C1:R10C1"))

$err:=PutList($aevt;"data";ArrayOfValues)

$err:=SendAppleEvent($aevt;$reply;kAEWaitReply+CanInteract;-1)

$err:=DisposeDesc($reply)

$err:=DisposeDesc($aevt)
end if
```

Example 2: Creating a chart

```
$err:=DoScript(Excel;"Select("+char(34)+"R1C1:R10C2"+char(34)+")")
$err:=CreateAEVT("core";"crel";Excel;$aevt)
if ($err=0)

$err:=PutLongParam($aevt;"kocl";"type";Long("chrt"))
$err:=SendAppleEvent($aevt;$reply;kAEWaitReply+CanInteract;-1)

$err:=DisposeDesc($reply)

$err:=DisposeDesc($aevt)
end if
```

Example 3: Copying chart to 4D

```
$err:=CreateAEVT("core";"getd";Excel;$aevt)
if ($err=0)

$err:=PutObject($aevt;"----";Obj("chrt";0;1))

$err:=PutLongParam($aevt;"rtyp";"type";Long("SPIC"))

$err:=SendAppleEvent($aevt;$reply;kAEWaitReply+CanInteract;-1)
```

```
if ($err=0)


$err:=GetPicParam($reply;”----”;1;aPicture)

End if

$err:=DisposeDesc($reply)

$err:=DisposeDesc($aevt)
End if
```
