Data Type Binary with the DCOM Wrapper

Note:
Mapping and handling of data type binary (IDL file format B) changed in version 5.2.1.6 of the DCOM Wrapper. To avoid conversions of any binary data, the mapping of data type binary was changed from automation type BSTR to automation type unsigned char.

This chapter covers the following topics:

- Mapping Data Type Binary
- Index Order
- Example

Mapping Data Type Binary

Version 5.3.1 and above

Software AG IDL		Visual Basic	C++	Note
B1		Byte	unsigned char	
B *n*	1 GB >= *n* > 1	Byte()	SAFEARRAY(unsigned char)	1
B *n* /i1,i2,i3	1 GB >= *n* >= 1	Byte()	SAFEARRAY(unsigned char)	1, 2
BV1		Byte	unsigned char	
BV *n*	1 GB >= *n* > 1	Byte()	SAFEARRAY(unsigned char)	1
BV *n* /i1,i2,i3	1 GB >= *n* >= 1	Byte()	SAFEARRAY(unsigned char)	1,2
BV	Variable size <=1 GB	Byte()	SAFEARRAY(unsigned char)	1
BV/i1,i2,i3	Variable size <=1 GB	Byte()	SAFEARRAY(unsigned char)	1

Notes

1. The maximum length you can specify depends on your hardware configuration and your software environment apart from EntireX. There is, however, an absolute limit (1 GB) that cannot be exceeded.
2. Depending on *n*, the SAFEARRAY has 3 dimensions (*n* = 1) or 4 dimensions (*n* > 1).

Index Order

The *n* (*n* > 1) length value is used as the rightmost index, for *n* = 1 the index *n* is dropped.

Example:

```
Bn / i1,i2,i3       -->  i1,i2,i3,n
B1 / i1,i2,i3       -->  i1,i2,i3
```

Note:
If the IDL file contains data of type binary, the code generated by the EntireX DCOM Wrapper 5.2.1.6 is not compatible to the code generated by older versions of the EntireX DCOM Wrapper. If the IDL file does not contain data of type binary, the codes are compatible.

Example

```
Library 'Bintest' Is
Program 'bintest1':'binary1' Is
 Define Data Parameter
  1  LittleBin  (B1)
  1  MaxBin (B126)
  1  BigBin (B16/20)
  1  VeryBigBin (B10/3,4,5)
 End-Define

Program 'bintest2':'binary2' Is
 Define Data Parameter
  1  Group
  2    BigBin    (B126/20)
  1  PerGroup    (/2,3)
  2    AnotherBin (B30/10)
 End-Define
```

Visual Basic

```
Dim obj As Object

...

Set obj = CreateObject("Eol.BinTest")

...

' Init parameter of program "binary1"
Dim vbLittleBin As Byte
Dim vbMaxBin()  As Byte
Dim vbBigBen()  As Byte
Dim vbVeryBigBin() As Byte

' Parameter of program "binary2"
' --

' others
Dim checksum As Long


' redim Parameter of program "binary1"
ReDim vbMaxBin(1 To 126)
ReDim vbBigBin(1 To 20, 1 To 16)
ReDim vbVeryBigBin(1 To 3, 1 To 4, 1 To 5, 1 To 10)


'''' binary1 ''''
```

```
' Fill in binary data
vbLittleBin = &H0


For i1 = 1 To 126
    vbMaxBin(i1) = i1
Next i1

For i1 = 1 To 20
    For i2 = 1 To 16
         vbBigBin(i1, i2) = 130 + i2 + i1
    Next i2
Next i1

For i1 = 1 To 3
    For i2 = 1 To 4
        For i3 = 1 To 5
            For i4 = 1 To 10
               vbVeryBigBin(i1, i2, i3, i4) = 130 + i2 + i1
            Next i4
        Next i3
    Next i2
Next i1

obj.binary1 vbLittleBin, vbMaxBin, vbBigBin, vbVeryBigBin

' Reading of binary data
' here for example:
' takes all binary data values and add them to variable checksum
checksum = 0

For i1 = 1 To 3
    For i2 = 1 To 4
        For i3 = 1 To 5
            For i4 = 1 To 10
               checksum = checksum + vbVeryBigBin(i1, i2, i3, i4)
            Next i4
        Next i3
    Next i2
Next i1

...

'''' binary2 ''''
Dim barray20_126() As Byte
Dim barray10_30() As Byte

ReDim barray20_126(19, 125)
ReDim barray10_30(9, 29)

For i1 = 0 To 19
   For i2 = 0 To 125
      ' fill in some values
      barray20_126(i1, i2) = i1 + i2
   Next i2
Next i1

obj.binary2_group.bigbin = barray20_126  '(B126/20)


For i1 = 0 To 1
   For i2 = 0 To 2
```

```
      For i3 = 0 To 9
         For i4 = 0 To 29
    ' fill in some values
            barray10_30(i3, i4) = 255 - i1 - i2
         Next i4
      Next i3

      obj.binary2_pergroup(i1, i2).anotherbin = barray10_30 ' (B30/10)
   Next i2
Next i1

obj.binary2 obj.binary2_group, obj.binary2_pergroup_all

' Reading of binary data
' here for example:
' takes all binary data values and add them to variable checksum

barray20_126 = obj.binary2_group.bigbin   '(B126/20)

checksum = 0
For i1 = 0 To 19
   For i2 = 0 To 125
      checksum = checksum + barray20_126(i1, i2)
   Next i2
Next i1

For i1 = 0 To 1
   For i2 = 0 To 2
      barray10_30 = obj.binary2_pergroup(i1, i2).anotherbin

      For i3 = 0 To 9
         For i4 = 0 To 29
            checksum = checksum + barray10_30(i3, i4)
         Next i4
      Next i3

   Next i2
Next i1
```