

# Using DCOM Wrapper Objects with Web Scripting Languages

Objects generated by the DCOM Wrapper can be used on Windows platforms in Web scripting languages that support ActiveX automation servers. They can be used both for client-side scripting (for example in Microsoft's Internet Explorer) or for server-side scripting (for example in Microsoft's Active Server Pages). Wrapper objects work with Microsoft's scripting engines for VBScript and JScript.

This chapter covers the following topics:

- Hints
  - Examples
- 

## Hints

### Use VBScript as Scripting Language

The DCOM Wrapper creates objects that support VARIANT references. Scripting languages such as VBScript pass output parameters by VARIANT references, not as the exactly defined type. For example, when a method of a COM interface has an out parameter of type string, VBScript passes a reference to a VARIANT to get the out parameter. DCOM Wrapper objects try to convert these references to the required reference type. VBScript supports VARIANT references.

### Using DCOM Wrapper with VBScript and in ASP

VBScript and ASP allow only data type *variant*. Therefore you cannot define the `Function_result` for these application languages as an array. Only scalar data types can be used as `Function_result`, because otherwise you would not be able to access the data from VBScript and ASP. From VBScript and ASP applications you can access OUT parameters defined as an array in the IDL only if these parameters are defined correctly as array data types in the VBScript and ASP.

### Sample IDL File

```
PROGRAM 'AR3_AV' is
  DEFINE DATA PARAMETER
    1 iparm          (AV/2,2,2) IN
    1 ioparm         (AV/2,2,2) IN OUT
    1 oparm          (AV/2,2,2) OUT      <- must be defined as array in VBScript
    1 Function_Result (AV) OUT          <- permitted
  END-DEFINE
```

### VBScript

```
Dim oparm()      <- define as Array
```

## Examples

For scripting languages where automation type references can be passed for out parameters, such as VBScript, usage is no longer restricted to functional methods (only input parameters and one function result output parameter).

### IDL File

```
Program 'Calc'; Is
  Define Data Parameter
    1 Operator_ (A1) In
    1 Operand_1 (I4) In
    1 Operand_2 (I4) In
    1 Function_Result (I4) Out
  End-Define
```

### Example of Active Server Pages

For Active Server Pages, this example would look as follows:

```
<% set oo=server.createobject("eol.example")%>
<%= oo.userid %>
<br>
<%= oo.calc(' - ',200,8) %>
<br>
```

### Known Problem and Solution

If VBScript is used in ASP to transfer arrays between the client and the server application, memory leaks can occur. To avoid this, do not use an array definition in level 1 of the IDL file. Instead, move your array definition inside a group and use the index access to array members.

Example:

```
library 'ASPLib' is
program 'AspProg' is
define data parameter
1 myArray      (A105/10)
end-define
```

change to:

```
library 'ASPLib' is
program 'AspProg' is
define data parameter
1 myEnvelope
2 myArray      (A105/10)
end-define
```

set/get item in array (Visual Basic)

```
' set
' <object name>.<access to group>.<array name>_indexAccess(<index list>) = <value>
' <object name>.myEnvelope.myArray_indexAccess( index ) = <value>

' get
' <variable> = <object name>.<access to group>.<array name>_indexAccess(<index list>)
' var = <object name>.myEnvelope.myArray_indexAccess( index )
```