

Using Broker ActiveX Control with Active Server Pages

Microsoft's Active Server Page (ASP) is an HTML page that includes one or more scripts and reusable ActiveX server components to create dynamic Web pages. The scripts and ActiveX components are processed on a Microsoft Web server before the page is sent to the user.

This chapter covers the following topics:

- Prerequisites
 - Designing a Web Page with ASP and Broker ActiveX Control
 - Using Broker ActiveX Control in Multiple Pages
-

Prerequisites

Installation prerequisites for all EntireX components are described centrally. See *Prerequisites*.

To use Broker ActiveX Control with ASP, you must have a running Web server.

Designing a Web Page with ASP and Broker ActiveX Control

Creating an Instance of the ActiveX Control and the Transaction Object

```
<%  
Set EBX      = server.Createobject("EntireX.Broker.ACI")  
  
Set torobj = EBX.CreateTransObject("calc.tor")  
  
or  
  
Set torobj = EBX.CreateTransObjectSA("calc.tor") (if returnvalue contains array)  
>%
```

Calling a TOR Method

```
Set retobj = torobj.calc(op,op1,op2)
```

Accessing the Data

Scalars

```
<% string = retobj.result %>
```

Structures

```
<% string = retobj.result.str %>
```

Arrays

You can have access to array elements:

```
<%string = retobj.retarr(0) %>
```

or

```
<%  
return = retobj.retarr  
string = return(0)  
%>
```

or

```
<%  
For Each element in retobj.retarr  
    string = element  
Next  
%>
```

Records

You can have access to record elements:

```
<%string = retobj.retrec(0).str %>
```

or

```
<%  
Set return = retobj.retrec(3)  
Response.Write return.str  
%>
```

or

```
<%  
For Each struct in retobj.retrec  
    string = struct.str  
Next  
%>
```

or

```
<%  
Array_Value = retobj.retrec  
For I = LBound(Array_Value) To UBound(Array_Value)  
    string = Array_Value(I).str  
Next  
%>
```

Using Broker ActiveX Control in Multiple Pages

Objects created by `Server.CreateObject` or `CreateTransObject` have page scope. They will be destroyed automatically when the current ASP page is finished.

To create an object with session or application scope, you can either use the `<OBJECT>` tag and set the `SCOPE` parameter to `SESSION` or `APPLICATION`, or store the object in a session or application variable.

For example, an object stored in a session variable, as shown in the following script, is destroyed when the Session object is destroyed. That is, when the session times out, or the `Abandon` method is called.

```
<% Set Session("torobj") = EBX.CreateTransObject("calc.tor")%>
```

You can destroy the object by setting the variable to "Nothing" or setting the variable to a new value.

```
<% Session("torobj") = Nothing %>
```