



## **Quick Reference Library**

### **ASP Quick Reference**

This sample shows the layout and design of the Quick References in a PDF format, but only contains a fraction of the content.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire content online.

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.

Welcome to the **DevGuru Built-in ASP Objects Quick Reference** guide. This is a valuable 71 page reference source that explains and provides comprehensive, working examples of code for all of the collections, events, methods, and properties belonging to the seven Built-in Active Server Pages objects.

The seven built-in objects are:

- [Application](#)
- **ASPError**
- **ObjectContext**
- **Request**
- **Response**
- **Server**
- **Session**

In the good old days (a few years ago!), most Web sites were created with HTML and simply displayed static pages. A few of the more adventurous programmers would use C or Perl to design a dynamic Web site utilizing the CGI technology. However, these techniques were plagued with security concerns and did not scale well to large sites. So, in general, dynamic sites remained relatively few in number and were time consuming and expensive to create. The introduction of the scripting languages, which could be embedded inside HTML code, opened new doors for dynamic site development. Active Server Pages (ASP), which managed to arrive just in time for the explosive growth of the World Wide Web, was a next logical step for Web-based application development. Introduced in 1996 by Microsoft, Active Server Pages proved to be an exciting, new technology that extended standard HTML by adding built-in objects, server-side scripts, access to databases, and ActiveX components. Another important development by Microsoft was to make the ASP scripting environment compliant with the Component Object Model (COM). COM created a standard communication mechanism between components. This step allowed non-vendor components, such as those offered by **DevGuru**, to share their properties, methods and events with other components in a process called OLE automation. Non-vendor components greatly extend the functionality of ASP applications.

The true power of ASP is the ease and rapidity with which developers can create and implement dynamic Web sites. Indeed, for today's modern Web commerce, a dynamic, database-driven, server-side application that interacts with the client is the norm.

ASP employs a scripting environment and VBScript is the default scripting language of choice. However, you can use other languages (such as JScript and Perl) as long as they have a scripting engine that is compatible with the ActiveX scripting standard.

Fortunately, you are not limited to just using Active Server Pages with Microsoft's Internet Information Server (IIS) and this has enhanced the popularity of ASP. For example, Chili!Soft is a proven industry leader in providing ASP engines for use with Web servers from FastTrack, Lotus, Netscape, O'Reilly, and many others. And Halcyon Software offers a brilliant Java-based implementation of the Microsoft ASP framework, allowing developers to deploy ASP applications on any platform.

Meanwhile, ASP continues to evolve. With the arrival of the millennium came the arrival of ASP version 3.0. Version 3.0 was released along with Internet Information Server (IIS) version 5.0 as part of the highly anticipated Microsoft Windows 2000. By far, the

most important new feature of version 3.0 is the addition of a seventh, intrinsic object called ASPError which should greatly simplify error handling. Other new features include the addition of three new methods to the Server object, and two new methods to both the [Application](#) object and the Session object.

Active Server Pages has ultimately proven to be of significant value to developers and fueled a revolution in the development of Web-based applications.

The future of ASP will continue to be very interesting.

## COLLECTIONS

Contents

[Application](#)

[Session](#)

[ClientCertificate](#)

Cookies

[Request](#)

[Response](#)

Form

[QueryString](#)

[ServerVariables](#)

StaticObjects

[Application](#)

[Session](#)

## EVENTS

OnEnd

[Application](#)

Session

OnStart

Application

Session

OnTransactionAbort

OnTransactionCommit

## METHODS

<a href="#">Abandon</a>	CreateObject
<a href="#">AddHeader</a>	End
<a href="#">AppendToLog</a>	Execute
BinaryRead	Flush
BinaryWrite	GetLastError
Clear	HTMLEncode
Contents.Remove	MapPath
<a href="#">Application</a>	Redirect
Session	SetAbort
Contents.RemoveAll	SetComplete
<a href="#">Application</a>	Transfer
Session	URLEncode
	Write

## OBJECTS

[Application](#)

[ASPError](#)

[ObjectContext](#)

[Request](#)

[Response](#)

[Server](#)

[Session](#)

## PROPERTIES

ASPCode	File
ASPDescription	IsClientConnected
Buffer	LCID
CacheControl	Line
Category	Number
Charset	Pics
CodePage	ScriptTimeout
Column	SessionID
ContentType	Source
Description	Status
Expires	Timeout
ExpiresAbsolute	

## INDEX

[Abandon](#)  
[AddHeader](#)  
[AppendToLog](#)  
[Application](#)  
ASPCode  
ASPDescription  
ASPErrors  
BinaryRead  
BinaryWrite  
Buffer  
CacheControl  
Category  
Charset  
Clear  
ClientCertificate  
CodePage  
Column  
Contents  
    [Application](#)  
    Session  
ContentType  
Cookies  
    Request  
    Response  
Contents.Remove  
    [Application](#)  
    Session  
Contents.RemoveAll  
    [Application](#)  
    Session  
CreateObject  
Description  
End  
Execute  
Expires  
ExpiresAbsolute  
File  
Flush  
Form  
GetLastError  
HTMLEncode  
IsClientConnected  
LCID  
Line  
MapPath  
Number  
ObjectContext  
OnEnd  
    [Application](#)  
    Session  
OnStart  
    [Application](#)  
    Session  
OnTransactionAbort  
OnTransactionCommit  
Pics  
QueryString  
Redirect  
Request  
Response  
ScriptTimeout  
Server  
ServerVariables  
Session  
SessionID  
SetAbort  
SetComplete  
Source  
StaticObjects  
    [Application](#)  
    Session  
Status  
Timeout  
Transfer  
URLEncode  
Write

## METHOD: Session.Abandon

---

### Session.Abandon

The **Abandon** method terminates a user session, destroys all the objects in the current Session object, and releases its resources. However, this deletion will not occur until all of the script is processed for the current page. When the session ends, the **OnEnd** event handler is called. By default, even if you do not call **Abandon**, the **Session** object is terminated after twenty minutes of idle time.

Code:

```
-----File1.asp-----
<%
Response.Write "Your SessionID is " & Session.SessionID
Session("Application") = "DevSite"
Session.Abandon
Response.Write "The Application name is " & Session("Application")
%>
```

```
-----File2.asp-----
<%
Response.Write "Your SessionID is " & Session.SessionID
%>
```

Output:

```
-----File1.asp-----
Your SessionID is 465107831
The Application name is DevSite

-----File2.asp-----
Your SessionID is 465107831
```

## METHOD: Response.AddHeader

---

### **Response.AddHeader** Name, Value

The **AddHeader** method adds a new named HTTP header with a specific value to the HTTP response. Note that once a header has been added, it cannot be removed. This method must be called before any output is sent to the client unless the `Response.Buffer` is set to true.

There are two mandatory arguments.

#### **Name**

The **Name** argument is the name of the new header variable. The name cannot contain any underscores ( `_` ).

#### **Value**

The **Value** argument is the initial value of the new header variable.

Code:

```
<%  
Response.AddHeader "MyHeader", "ERROR"  
%>
```

## METHOD: Response.AppendToLog

---

### **Response.AppendToLog(String)**

The **AppendToLog** method adds (appends) a string to the end of the entry of the Web server log for this request.

There is one mandatory argument.

### **String**

The **String** argument is the string to be appended. The string can have a maximum length of 80 characters. It cannot contain commas because the log is comma-delineated.

Code:

```
<%  
Response.AppendToLog("Error in Processing")  
%>
```

Output is appended to the end of the IIS log file:

```
10.78.176.37, - , 03/20/97, 7:55:20, W3SVC, SALES1, 10.78.176.37, Error in Processing
```

# OBJECT: Application

---

Modified in version 3.0

The **Application** object shares, stores, and retrieves information in response to requests from users to the application.

## COLLECTION PROPERTIES

### [Contents](#) Object

A collection of all of the items which have been created and added to the **Application** object during client sessions through script commands, such as the **Server.CreateObject**, rather than by using the HTML <OBJECT> tag.

### [StaticObjects](#) Object

A collection of all of the items which have been created and added to the **Application** object during client sessions by using the HTML <OBJECT> tag, rather than using script commands.

## METHODS

### [Contents.Remove](#) Method Implemented in version 3.0

The **Contents.Remove** method is used to remove a single item from a **Contents** collection.

### [Contents.RemoveAll](#) Method Implemented in version 3.0

The **Contents.RemoveAll** method is used to remove all items from a **Contents** collection.

### [Lock](#) Method

The **Lock** method prevents all other users from making changes in the **Application** object.

### [Unlock](#) Method

The **Unlock** method allows any user to have access to any of the **Application** object properties in order to make changes.

## EVENTS

### [OnEnd](#) Event

The **OnEnd** event occurs when the **Application** quits. This should not occur unless all user sessions are over. The signal of this event will run a handler script in the Global.asa file, if the script exist.

### [OnStart](#) Event

The **OnStart** event occurs before the start of any new session by a user (i.e., before the **Application** object is first referenced). The signal of this event will run a handler script in the Global.asa file, if the script exist.

# COLLECTION PROPERTY: Application.Contents

---

## Application.Contents(Key)

The **Contents** collection property contains a list of the items that have been created and added to the Application object. Objects can be added to the collection by using **Server.CreateObject** or by assigning scalar variables.

You can iterate through a collection using a **For Each item in ... Next** loop.

There is one mandatory argument.

### Key

The **Key** argument is the name of the item to be retrieved.

Code:

```
<%  
Application("name") = "Application Maker"  
Application("publishdate") = "05/15/01"  
Application("author") = "DevGuru"  
Set Application("Obj1") = Server.CreateObject("ADODB.Connection")
```

```
For Each Item in Application.Contents  
  If IsObject(Application.Contents(Item)) Then  
    Response.Write Item & " is an object.<BR>"  
  Else  
    Response.Write Item & "=" & Application.Contents(Item) & "<BR>"  
  End If  
Next  
>%
```

Output:

```
name=Application Maker  
publishdate=05/15/01  
author=DevGuru  
OBJ1 is an object.
```

# COLLECTION PROPERTY: Application.StaticObjects

---

## Application.StaticObjects(Key)

The **StaticObjects** collection property is a collection that contains of all of the items created within the **Application** using the HTML <OBJECT> tag. You can use this collection to find the value of any property for any object.

You can iterate through a collection using a **For Each item in ... Next** loop.

There is one mandatory argument.

### Key

The **Key** argument is the name of the item to be retrieved.

Code:

-----Global.asa-----

```
<OBJECT RUNAT=Server SCOPE=Application ID=MyInfo PROGID="MSWC.MyInfo">
</OBJECT>
```

```
<OBJECT RUNAT=Server SCOPE=Application ID=MyConnection
PROGID="ADODB.Connection">
</OBJECT>
```

```
<OBJECT RUNAT=Server SCOPE=Application ID=MyADRot PROGID="MSWC.ADRotator">
</OBJECT>
```

-----File.asp-----

```
<%
For Each Item In Application.StaticObjects
Response.Write Item & "<BR>"
Next
%>
```

Output:

```
MyInfo
MyConnection
MyADRot
```

## METHOD: Application.Contents.Remove

---

Implemented in version 3.0

### **Application.Contents.Remove** (Name|Integer)

The **Contents.Remove** method is used to delete one specified item in the **Application.Contents** collection.

You may use only one of the two possible choices for the mandatory argument.

#### **Name** or **Integer**

The **Name** argument is the name of the item to be deleted. It must be enclosed in a pair of quotes.

The **Integer** argument is the position number of the item in the collection to be deleted. The numbering sequence for a collection starts at one, not zero.

Code:

```
<%  
Application("name") = "Application Maker"  
Application("publishdate") = "05/15/01"  
Application("author") = "DevGuru"  
Set Application("Obj1") = Server.CreateObject("ADODB.Connection")
```

```
Application.Contents.Remove(1)  
Application.Contents.Remove("publishdate")
```

```
For Each Item in Application.Contents  
  If IsObject(Application.Contents(Item)) Then  
    Response.Write Item & " is an object.<BR>"  
  Else  
    Response.Write Item & "=" & Application.Contents(Item) & "<BR>"  
  End If  
Next  
>%
```

Output:

```
author=DevGuru  
Obj1 is an object.
```

## METHOD: Application.Contents.RemoveAll

---

Implemented in version 3.0

### Application.Contents.RemoveAll

The **Contents.RemoveAll** method deletes all items that are in the **Application.Contents** collection.

Remember to include the pair of empty ( ).

Code:

```
<%
```

```
Application.Contents.RemoveAll( )
```

```
%>
```

## METHOD: Application.Lock

---

### Application.Lock

The **Lock** method prevents all other users from changing any of the variables in the **Contents** collection of the **Application** object.

**Application** objects are designed to be shared among an unlimited number of users. Therefore, you need the ability to allow only one user at a time to make changes and you do this by locking everybody else out.

You can use the **Unlock** method to explicitly remove the **Lock** placed upon the **Application** object. Remember, to completely unlock an object, you must call **Unlock** the same number of times you have called **Lock**. Fortunately, the server will automatically unlock all locks placed on the object when the script times out or the .asp file ends.

Code:

```
<%
```

```
Application.Lock
```

```
%>
```

## METHOD: Application.Unlock

---

### Application.Unlock

The **Unlock** method is used to explicitly unlock the variables in the **Contents** collection of the **Application** object.

In contrast, the **Lock** method prevents all other users from changing any of the variables in the **Contents** collection of the **Application** object. This ability to lock is required since **Application** objects are designed to be shared among an unlimited number of users. Therefore, you need the ability to allow only one user at a time to make changes and you do this by locking everybody else out.

Remember, to completely unlock an object, you must call **Unlock** the same number of times you have called **Lock**. Fortunately, the server will automatically unlock all locks placed on the object when the script times out or the .asp file ends.

Code:

```
<%  
Application.Unlock  
%>
```

## EVENT: Application\_OnEnd

---

The **Application\_OnEnd** event occurs when the **Application** ends. This should only happen when the web server is stopped by the operating system in a normal manner. The **Application\_OnEnd** event is simply a subroutine with a reserved name that is placed within the Global.asa file. It can contain any script that you wish to run after all user sessions are finished. For example, you may wish to compute and store statistics on user sessions for future reference.

Note: The only built-in ASP objects available from within the OnEnd event handler are Server and Application.

Code:

```
-----Global.asa-----
```

```
<script Language="VBScript" RUNAT=Server>
```

```
Sub Application_OnEnd()
```

```
    Calculate_Stats()
```

```
End Sub
```

```
Sub Application_OnStart()
```

```
    Application("NumSession") = 0
```

```
    Application("NumVisited") = 0
```

```
End Sub
```

```
Sub Session_OnEnd()
```

```
    Application("NumSession") = Application("NumSession") - 1
```

```
End Sub
```

```
Sub Session_OnStart()
```

```
    Application("NumSession") = Application("NumSession") + 1
```

```
    Application("NumVisited") = Application("NumVisited") + 1
```

```
End Sub
```

```
</script>
```

```
-----File1.asp-----
```

```
Response.Write "You are " & Application("NumSession") & " of " & Application("NumVisited") &  
" users."
```



## **Quick Reference Library**

### **WML Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **VBScript Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **JetSQL Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **JavaScript Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **HTML Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **CSS Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **ADO Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **XSLT Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **XMLDOM Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **XHTML Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **WSH Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.



## **Quick Reference Library**

### **WMLScript Quick Reference**

This content is not available in this sample PDF.

Please click [here](http://DevGuru.com) to visit DevGuru.com to view the entire Quick Reference online for free or to purchase the PDF version..

Copyright 1999-2004 by Infinite Software Solutions, Inc. All rights reserved.