

Utilizzo di documenti Office con ASP

A partire da Office 97 Excel e Word possono interpretare l'HTML, così si può fare in modo di costruire una pagina ASP che riporti in output file Word o Excel. Può essere utile, ad esempio, presentare dei dati da stampare in una finestra di Microsoft Word oppure mostrare dei dati e il risultato di alcune formule realizzate attraverso Excel.

Per fare questo si sfrutta l'apposita proprietà [ContentType](#) dell'Oggetto [Response](#).

Per mezzo di questa proprietà si può definire quale tipo di dati (es. Word o Excel) saranno contenuti all'interno della pagina.

Innanzitutto bisogna **bufferizzare** il contenuto della pagina (cioè spedire al browser dell'utente la pagina solo dopo che è terminata l'elaborazione da parte del server) impostando a true la proprietà [Buffer](#) dell'oggetto Response.

Vediamo come:

'Attivazione del buffer

Response.Buffer = true

A questo punto bisogna dichiarare *il tipo di documento*.

Per **Word** bisogna scrivere:

' Dichiarazione del tipo di documento scelto (word)

Response.ContentType = "application/msword"

Mentre per **Excel** bisogna scrivere:

' Dichiarazione del tipo di documento scelto (excel)

Response.ContentType = "application/vnd.ms-excel"

Va inserita ora una riga che impiega il metodo [AddHeader](#) dell'Oggetto Response per **inserire il nome del file**.

Ad esempio, per **Word** si può scrivere:

Response.AddHeader "content-disposition", "inline; filename=esempio.doc" ' oppure esempio.rtf

Ad esempio, per **Excel** bisogna scrivere:

Response.AddHeader "content-disposition", "inline; filename=esempio.xls"

A questo punto vanno inserite le scritture su schermo di ciò che si vuole visualizzare.

Un esempio di utilizzo di un **file Excel con ASP**:

```
<%
  Response.Expires=0      ' la pagina scadrà dopo 0 minuti
                          ' serve a non memorizzare la pagina all'interno della cache del Browser
  Response.buffer = true  ' bufferizzare il contenuto della pagina
                          ' (cioè spedire al browser dell'utente la pagina solo dopo che è
                          ' terminata l'elaborazione da parte del server)

  Response.ContentType = "application/vnd.ms-excel"
                          ' Dichiarazione del tipo di documento scelto (excel)

  Response.AddHeader "content-disposition", "inline; filename=esempio.xls"

  response.write "<table width=240>"
  response.write "<tr>"
  for i = 1 to 5
    response.write "<td width=40>"
    response.write i + i
    response.write "</td>"
  next
  response.write "<td width=40><b>=Somma(A1:E1)</b></td>"
  response.write "</tr>"
  response.write "</table>"

  Response.flush ' metodo Flush per inviare all'utente il contenuto del buffer
  Response.end   ' metodo End per interrompere l'esecuzione della pagina.
%>
```

Un esempio di utilizzo di un **file Word con ASP**:

VBScript

```
<%
  Response.Buffer = true          ' Attivazione del buffer

  Response.ContentType = "application/msword"  ' Dichiarazione del tipo di documento scelto
                                              ' Scelta del file su cui scrivere
  Response.AddHeader "content-disposition", "inline; filename=stampa.doc"

  ' Scrittura di un testo: genero il file stampa.rtf oppure stampa.doc non fisicamente sul Server
  ' ma conservato in una locazione di memoria virtuale e muore appena l'utente lo chiude,
  ' ad esempio dopo aver eseguito la stampa.
  ' Questo particolare processo evita problemi di concorrenzialità negli accessi.

  Response.Write("Hello Word :-)")

  Response.Flush;    ' Invio dell'output bufferizzato
  Response.End;      ' Chiusura del documento
%>
```

E con uso di **JScript**:

```
<%@LANGUAGE = JScript%>
<%
    Response.Buffer = true;                                // Attivazione del buffer

                                                         // Dichiarazione del tipo di documento scelto
    Response.ContentType = "application/msword";

                                                         // Scelta del file su cui scrivere
    Response.AddHeader("content-disposition", "inline; filename=stampa.doc");

// Scrittura di un testo

// Genero il file stampa.rtf oppure stampa.doc non fisicamente sul Server
// ma conservato in una locazione di memoria virtuale e muore appena l'utente lo chiude,
// ad esempio dopo aver eseguito la stampa.
// Questo particolare processo evita problemi di concorrenzialità negli accessi.

    Response.Write("Hello Word :-");

    Response.Flush;    // Invio dell'output bufferizzato
    Response.End;      // Chiusura del documento
%>
```

Inserire immagini

Uso di tag img estraendo da DB il link a immagini (stamp_imp.asp)

```
<%
    Option Explicit

    Dim Cn
    Dim sSQL
    Dim RS
    Dim num
    Dim data

    Set Cn = Server.CreateObject("ADODB.Connection")
    Cn.Open("Stringa di connessione")

    num = Request.QueryString("idImg")          ' impostata in indexI.htm

    sSQL ="select * from Tabella3 WHERE id =" &num
        ' campi id, link e tipo con 2 immagini (per prova WHERE id =1")

    Set RS = Cn.execute(sSQL) ' esecuzione query

    Response.Expires=-1 ' la pagina scadrà subito
        ' serve a non memorizzare la pagina all'interno della cache del Browser
    Response.Write("Informazioni sull'immagine richiesta: <br />")
```

```

Response.Write(" <br />")

Response.Write("link: " & RS("link") & "<br />")
Response.Write("tipo: " & RS("tipo") & "<br />")
Response.Write(" <br />")

data = RS.Fields(1).Value

Response.Write("Ora visualizzo l'immagine richiesta: <br />")
Response.Write(" <br />")

' response.write("<img src= " & data & " /> immagine di tipo " & RS(2) & " <br />")

' inserimenti alternativi dell'immagine funzionanti

response.write("<img src= " & RS("Link") & " /> immagine di tipo " & RS("tipo") & "<br />")
rem response.write("<img src= " & RS(1) & " /> immagine di tipo " & RS(2) & " <br />")
rem response.write("<img src= " & RS.Fields(1).Value & " />immagine di tipo "
                    & RS(2) & " <br/>")

Cn.Close

' Un po di pulizia...

Set RS = Nothing
Set Cn = Nothing
%>

```

indexI.htm per impostare l'identificativo nell'estrarre da **DB** il **link** all'**immagine**

```

<html>
<head>
</head>
<body>Ora inserisci l'immagine (con nome in forma di numero intero) come link:
<ul>
<p><li><a href="stamp_img.asp?idImg=1">prima</a> immagine</li>
<li><a href="stamp_img.asp?idImg=2">seconda</a> immagine</li></p>
</ul>
</body>
</html>

```

Link all' immagine (img_VB.asp) con passaggio di identificativo in QueryString nello stesso file

VBScript

```
<%  
    idImg = Request.QueryString("idImg")  
%>  
<html>
```

Ora vedi la posizione dell'immagine:

```
<div align="center"></div>
```

Ora inserisci l'immagine (con nome in forma di numero intero) come link:

```
<a href="img_VB.asp?idImg=1">immagine </a>
```

```
</html>
```

Link all' immagine (img_JS.asp) con passaggio di identificativo in QueryString nello stesso file con uso di **JScript**:

```
<%@LANGUAGE = JScript%>  
<%  
    var idImg = parseInt(Request.QueryString("idImg"));  
%>  
<html>
```

Ora vedi la posizione dell'immagine:

```
<div align="center"></div>
```

Ora inserisci l'immagine (con nome in forma di numero intero) come link:

```
<a href="img_JS.asp?idImg=2">immagine </a>
```

```
</html>
```

Stampare al volo file Ms Word 2000 (indexVB.asp) come link in tabella report da DB

```
<html>
<head><title>Stampare al volo file Ms Word 2000</title></head>
<body>
<%
Dim oConn
Dim oRS
Dim sSQL

Set oConn = Server.CreateObject("ADODB.Connection")

Set oRS = Server.CreateObject ("ADODB.RecordSet")

oConn.Open("Stringa di connessione")

'Se si vuole visualizzare su finestra del browser per controllare il risultato

sSQL = "SELECT * FROM Tabella1"

oRs.open sSQL, oConn

Response.Write("<table border='1'>")

Response.Write("<tr>")

while not oRs.eof

    Response.Write("<td><a href='stampa0.asp?id=" & oRS("id")
                    & "' target='_blank'>Stampa</a></td>")
    Response.Write("<td>" & oRS("nome") & "</td>")
    Response.Write("<td>" & oRS("classe") & "</td>")
    Response.Write("</tr>")

    oRS.movenext
wend

Response.Write("</table>")

oConn.close
Set oRS=nothing
Set oConn=nothing
%>

</body>
</html>
```

Stampare al volo file Ms Word 2000 (stampa0.asp) dell'occorrenza nel **DB** il cui identificativo è impostato nel file *indexVB.asp*

```
<%
Option Explicit

Dim Cn
Dim sSQL
Dim RS
Dim num

Set Cn = Server.CreateObject("ADODB.Connection")
Cn.Open("Stringa di connessione")

num = Request.QueryString("id") ' id passato da link attivato nella pagina aperta da indexVB.asp

sSQL ="select * from Tabella1 WHERE id =" & num
Set RS = Cn.execute(sSQL) ' esecuzione della query

' Genero il file stampa.rtf oppure stampa.doc non fisicamente sul Server
' ma conservato in una locazione di memoria virtuale e muore appena l'utente lo chiude,
' ad esempio dopo aver eseguito la stampa.
' Questo particolare processo evita problemi di concorrenzialità negli accessi.

Response.Buffer = true ' Attivazione del buffer

Response.ContentType = "application/msword" ' Dichiarazione del tipo di documento scelto

Response.AddHeader "content-disposition", "inline; filename=stampa.rtf"
' Scelta del file su cui scrivere
' sintassi Response.AddHeader nome, valore

' accelera
Response.CacheControl = "no-cache"
Response.Expires=-1 ' la pagina scadrà senza
' memorizzazione all'interno della Cache del Browser

'Creo graficamente il file Ms Word (Office '97 o successivi)

Response.Write("ID: " & RS("id")& " ")
Response.Write("Nome: " & RS("nome") & " ")
Response.Write("Classe: " & RS("classe") & " ")

Response.Flush ' Invio dell'output bufferizzato
Response.End ' Chiusura del documento

Cn.Close
' Un po di pulizia...

Set RS = Nothing
Set Cn = Nothing
%>
```

Nb: per [Stringhe di connessione quale usare ?](#)