

Q&A@infomedia.it

Questo spazio è dedicato alla risoluzione dei vostri problemi di programmazione. Proponete domande tecniche ed otterrete risposte pratiche ed esaurienti ai vostri quesiti



Upload di file con ASP

Sto facendo l'upload di alcuni file grafici da client utilizzando un database Access: vorrei salvare l'immagine in una directory del mio server web (e fin qui non ci sono problemi) e salvare il path e relativa descrizione nel database per poi richiamarli tramite ASP nel momento della visualizzazione.

Riesco a salvare correttamente l'immagine nella directory del mio web server, ma non riesco a salvare path e descrizione dell'immagine nel database. Cosa posso fare?

Grazie per qualsiasi aiuto!

Antonio



Andrea Chiarelli

Ciao Antonio, nel tuo messaggio non indichi il componente che utilizzi sul server per gestire l'upload né indichi le difficoltà che incontri per il salvataggio dei dati nel database.

In ogni caso, immagino che la descrizione dell'immagine e il nome del file venga data dall'utente in uno dei campi della form di upload mentre l'informazione sul path di memorizzazione del file d'immagine sul server lo decidi tu.

Supponendo che tu sia in questa situazione e che utilizzi Posting Acceptor come componente per l'upload, dovresti creare un recordset a partire dal tuo database e memorizzare le informazioni come nel seguente esempio:

```
<%
'Recuperazione delle informazioni sul file
di immagine
Dim NomeFile: NomeFile = Request.Form
("FileName")
Dim Estensione: Estensione = Request.Form
("FileExtension")
Dim Descrizione: Descrizione = Request.Form
("Descrizione")

'Creazione del recordset e salvataggio
delle informazioni
Dim rs: Set rs = Server.CreateObject
("ADODB.Recordset")
```

```
rs.ActiveConnection = "Provider=
Microsoft.Jet.OLEDB.4.0;DataSource=
c:\db\mydb.mdb"
rs.LockType = 3 'adLockOptimistic
rs.Open "SELECT * FROM Immagini"

rs.AddNew

rs("codice") = 1
rs("path") = "immagini/" & Left(NomeFile,
Instr(NomeFile, ",") - 1) & Estensione
rs("descrizione") = Descrizione

rs.Update
rs.Close: Set rs = Nothing
%>
```

Posting Acceptor aggiunge automaticamente alla collection Form dell'oggetto Request i seguenti elementi: FileName, FilePath, FileSize, FileExtension.

In particolare FileName conterrà una stringa del tipo: "immagine, c:\percorso_del_client\immagine.gif", da cui la necessità di estrarre in modo un po' articolato il nome del file d'immagine.

Naturalmente se usi un componente diverso per l'upload dovrai consultare la relativa documentazione per capire come accedere alle informazioni che ti servono, ma in linea di massima credo che dovresti soltanto modificare le prime tre istruzioni di assegnamento.

A questo punto per la visualizzazione dell'immagine dovresti scrivere del codice analogo al seguente:

```
<%
Dim rs: Set rs = Server.CreateObject
("ADODB.Recordset")
rs.ActiveConnection = "Provider=
Microsoft.Jet.OLEDB.4.0;DataSource=
c:\db\mydb.mdb"
rs.Open "SELECT * FROM Immagini WHERE
codice = 1"

Response.Write "<IMG SRC=' " & rs("path")
& "'><BR>"
Response.Write rs("descrizione")

rs.Close: Set rs = Nothing
%>
```

Q

La classe System.Convert

Spettabile redazione,
vorrei sapere come sia possibile convertire una stringa in un intero e viceversa in C#.
Grazie.

Emanuele

A

Alberto Falossi

La classe System.Convert espone molti metodi statici per tutte le conversioni tra tipi di dati base. Per convertire una stringa in intero devi usare:

```
// stringa iniziale
string theString = "123";

// intero a 16 bit
short s =
System.Convert.ToInt16(theString);

// intero a 32 bit
int i = System.Convert.ToInt32(theString);

// intero a 64 bit
long l = System.Convert.ToInt64(theString);
```

Per l'operazione inversa puoi usare

```
theString = System.Convert.ToString(i);
```

oppure il metodo ToString presente in tutti gli oggetti .NET:

```
theString = i.ToString();
```

Saluti

Q

Compatibilità binaria in COM

Salve,
avrei bisogno di un consiglio per realizzare un pacchetto di installazione. Si tratta di una applicazione VB6 il cui pacchetto è già in distribuzione ed è fatto in modo da installare tutto (eseguibili e librerie) in una unica directory senza registrare alcun componente. Questo ha causato problemi sulle macchine dove erano presenti nella directory di sistema le stesse DLL e OCX della mia applicazione, ma di VB5. Vorrei cambiare il pacchetto in modo che scarichi DLL ed OCX nella directory di sistema e così facendo le applicazioni installate fatte con VB6 dovrebbero continuare a funzionare. È una buona soluzione? Gli OCX e le DLL devono essere necessariamente registrati?

Ciao e grazie

Luca Dolciotti

A

Marco Bellinaso

Solitamente l'installazione di componenti COM (ActiveX, DLL) o di librerie condivise viene effettuata nella directory \System di Windows. In realtà per i componenti COM questo non è un obbligo, in quanto l'applicazione per caricarli non sa a priori il percorso fisico del file, ma lo estrae dal registro di Windows (questo è il motivo per cui i componenti devono necessariamente essere registrati), quindi potrebbero venire copiati in qualsiasi posto. L'installazione sotto \System è però consigliata perché evita di avere copie multiple dello stesso file sparse per il disco fisso, che è effettivamente uno dei vantaggi dei componenti COM.

A proposito della compatibilità dei nuovi file con i programmi VB5, per la libreria runtime di VB6 (MSVBVM60.DLL) il problema non si pone, in quanto il runtime di VB5 ha comunque un nome diverso. Per i controlli ActiveX o altre DLL, il programma di setup creato con il Package and Deployment Wizard controlla la versione e la data dell'eventuale file già presente su disco, e installa il file del pacchetto solo nel caso questo sia più recente. I programmi VB5 continueranno a funzionare, in quanto i nuovi componenti installati avranno compatibilità binaria o di progetto rispetto al file sovrascritto, quindi potranno essere caricati senza alcuna differenza dal programma client.

Saluti.

Q

CLI e CLR: qual è la differenza?

Studiando la documentazione del .NET Framework mi imbatto spesso nelle sigle CLI e CLR: sono sinonimi del runtime .NET? Se sì, perché usare due termini differenti?

Gianfranco

A

Alberto Falossi

La Common Language Infrastructure (CLI) è un sottoinsieme delle funzionalità del Common Language Runtime (CLR). La CLI comprende l'Execution Engine e una parte della libreria della classi (tipi base, Reflection, XML): come si può intuire definisce i componenti essenziali per l'esecuzione di applicazioni .NET. Il CLR è l'implementazione della CLI sotto Windows con l'aggiunta di alcune componenti come ADO.NET, ASP.NET e Windows Forms.

Recentemente Microsoft ha commissionato a Corel il porting della CLI sotto Linux e presto (probabilmente nel primo trimestre 2002) sarà disponibile l'implementazione di .NET per un sistema operativo diverso da Windows.