

Oggetto ADO (ActiveX Data Objects): metodo di accesso universale

Per gestire i database con **tecnologia ASP** si utilizzano **strumenti ADO** (ActiveX Data Objects): ADO¹ consente di scrivere un'applicazione per la gestione e l'accesso ai dati contenuti in un server di database tramite un provider² OLE DB (interfaccia a basso livello che fornisce un **modello di accesso ai dati universale** che ne consente la gestione indipendentemente dal formato e dal metodo di memorizzazione; permette l'accesso non solo ai database relazionali ma a qualsiasi fonte dati : database locali o remoti, relazionali o non relazionali, sistemi di file, posta elettronica, testo, grafica e oggetti multimediali, aziendali, personalizzati ...). Poiché tuttavia ciascun provider dispone di caratteristiche uniche, le modalità di interazione tra l'applicazione e ADO variano leggermente a seconda del provider stesso.

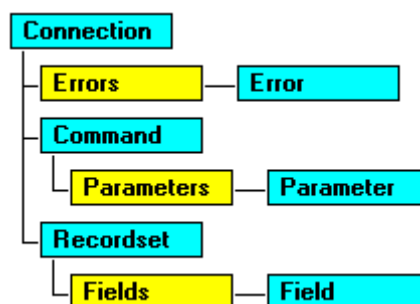
ADO, ActiveX Data Objects

E' un'architettura che fornisce oggetti di alto livello per l'accesso ai dati, e in particolare contiene oggetti per la comunicazione con i database.

I 3 oggetti principali sono:

1. l'oggetto **Connection** consente di stabilire la connessione. Contiene informazioni riguardanti il fornitore dei dati e lo schema del database
2. l'oggetto **Recordset**³ consente di operare con i dati contenuti in una tabella e recuperati attraverso una query. Contiene tutti o parte dei records di una tabella del database indicato nell'oggetto Connection
3. l'oggetto **Field** è una collection (una sorta di array) che rappresenta le colonne di una tabella

L'Object Model degli ADO 2.0



Il modello ad oggetti di ADO 2.0 è molto semplice, infatti gli oggetti sono soltanto sei (più tre Collection).

¹ Ideato come interfaccia dati client/server. Una limitazione di ADO consiste nell'impossibilità di creare fonti di dati ODBC pur se il provider predefinito e' Microsoft OLE DB per ODBC (**Open Database Connectivity** componente Microsoft Windows ed in particolare di Windows Open Services Architecture, che consente l'accesso a tutti i tipi di database relazionali).

² Il termine provider indica in generale un elemento fornitore dal quale si ottengono servizi o dati.

³ Un **Recordset** e' un gruppo di risultati in risposta ad un'interrogazione alla fonte di dati.

- In cima a tutto, abbiamo l'oggetto **Connection** che rappresenta la connessione tra il Client che utilizza gli ADO ed il Server di DataBase (in pratica il motore, che per i file MDB è il Jet). Per quanto sopra l'oggetto Connection gestisce anche le transazioni; nel senso che Begintrans, Committrans e Rollbacktrans avvengono all'interno di un oggetto Connection e non più all'interno di un Workspace. Da ciò discende che una transazione può avvenire solo nei riguardi di un unico DataBase. E se si deve gestire una transazione che riguarda più DataBase bisogna usare il Microsoft Transaction Server (così Microsoft vende un'altra licenza di Windows NT Server), ma questa è un'altra storia...
- Logicamente dipendente dall'oggetto Connection abbiamo l'oggetto **Recordset**, ovvero l'insieme dei Record estratti da una tabella o per mezzo di una Query, un comando SQL ecc. dal nostro DataBase.
- L'oggetto Recordset contiene una Collection **Fields** che è l'insieme degli oggetti **Field**, ovvero i campi che compongono ciascun Record del Recordset.
- Sempre dipendente diretto della Connection troviamo l'oggetto **Command** che è, in pratica, ogni comando specifico che vogliamo eseguire nei riguardi del nostro DataBase; p.es. una Query memorizzata, magari con parametri.
- E infatti l'oggetto Command contiene una Collection **Parameters**, composta da oggetti **Parameter**, ovvero tutti i parametri o argomenti associati al comando che vogliamo eseguire (Query parametrica o, nel caso per esempio di SQL Server o Oracle, anche Stored Procedure, ecc.).
- Infine la Connection contiene una Collection **Errors** composta da oggetti **Error**, ovvero tutti gli errori generati per effetto di un mancato funzionamento dell'OLEDB Provider (impossibilità a stabilire una connessione, comando SQL sintatticamente errato, parametri mancanti o di tipo diverso da quello previsto dalla query parametrica e chi più ne ha più ne metta...).

E' importante capire subito che, dato questo modello ad oggetti fondamentale, non è detto che ciascun oggetto abbia sempre le stesse caratteristiche; questo perché gli ADO possono essere usati in molti modi diversi e per accedere a fonti dati eterogenee. Tanto per fare un esempio l'oggetto Connection potrebbe non essere in grado di gestire le transazioni perché non supportate dall'OLEDB Provider in uso per accedere a quella certa fonte dati, oppure perché si sta usando una Connection lato client ed un Recordset sconnesso.

Connessione ad un database → http://professoressa.altervista.org/Dispense_V/Intro_ADO.pdf

Linkografia:

http://www.mrwebmaster.it/asp/introduzione-ado_9948.html

http://www.w3schools.com/asp/ado_intro.asp

<http://www.html.it/pag/31805/database-access-with-ado/>

<https://www.soft-land.org/documenti/adomod>

ADO vs ADO.NET

[https://msdn.microsoft.com/it-it/library/904fck4k\(v=vs.90\).aspx](https://msdn.microsoft.com/it-it/library/904fck4k(v=vs.90).aspx)

.NET vs ADO.NET [slides](#) *ACCESSO ai DATI di Maurizio Fermaglia* ([scaricabili](#))

