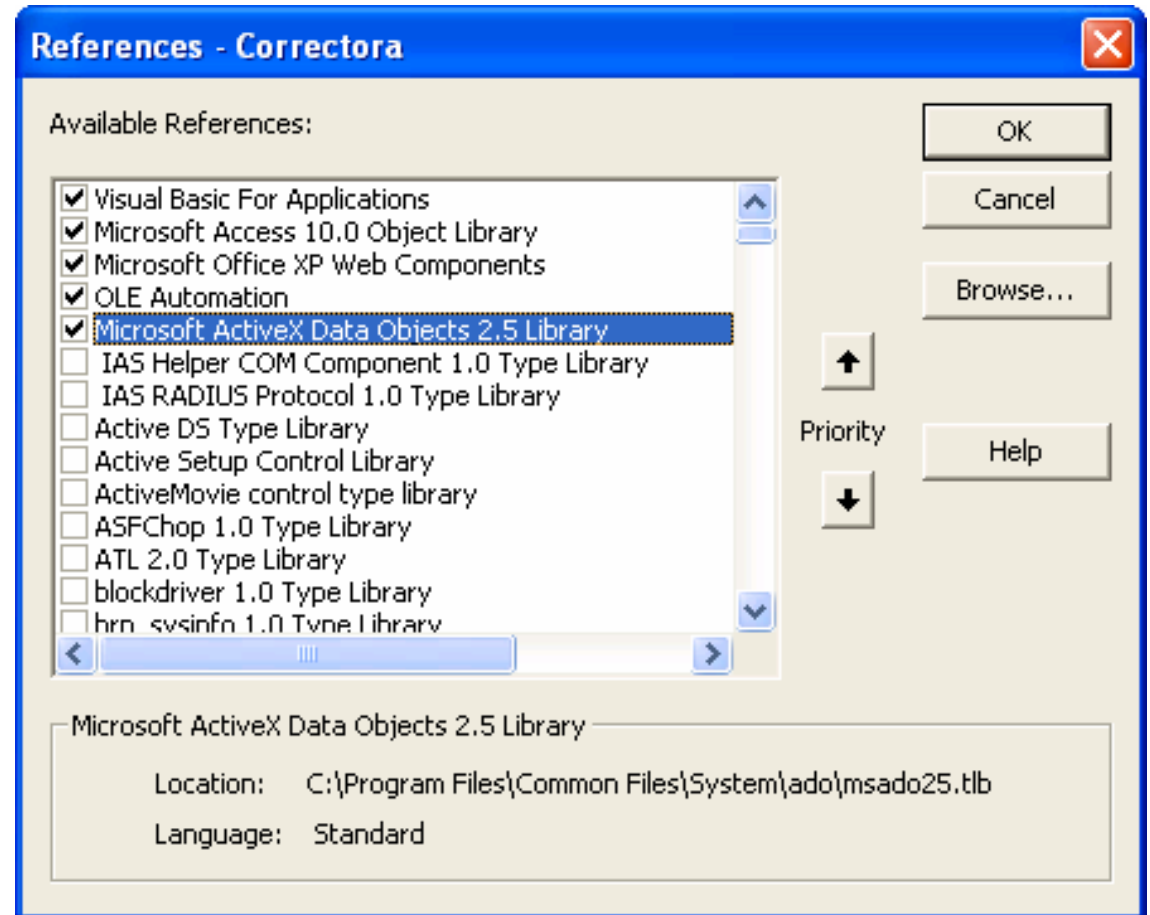


ActiveX Data Objects I

● Aspectos gerais

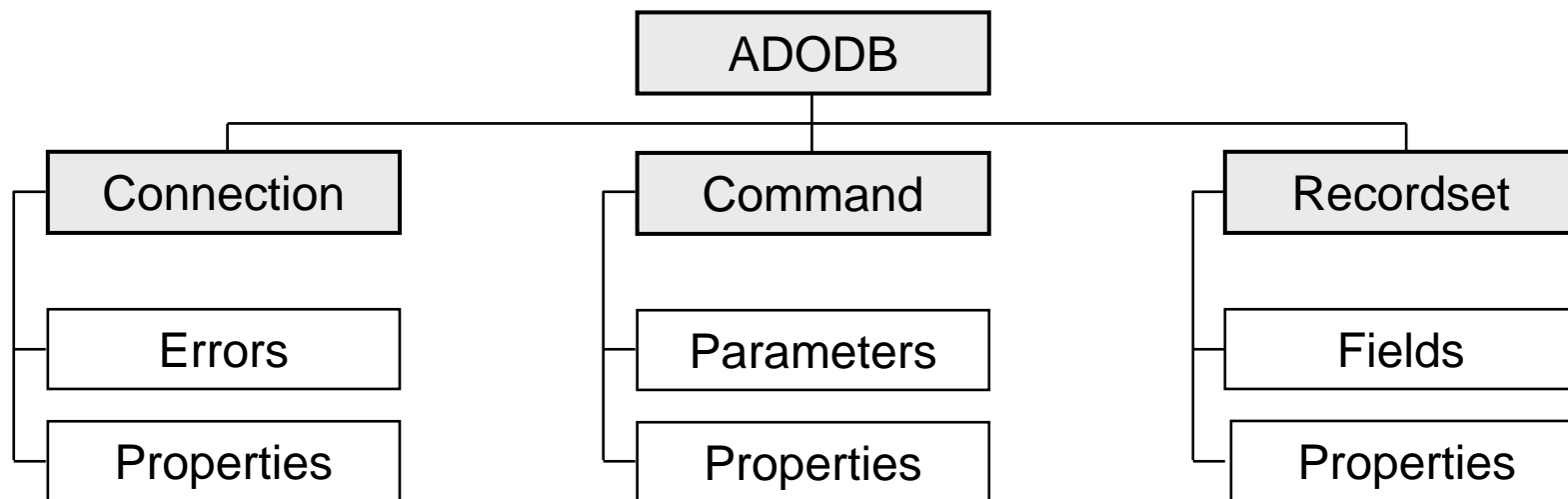
- A biblioteca ADO oferece uma interface genérica para acesso a dados
- Permite comunicar com um amplo conjunto de diferentes formatos de dados
- Define uma única sintaxe para manipular o mesmo tipo de objectos (é independente do formato dos dados)
- Constitui por si só uma hierarquia de objectos independente da hierarquia de objectos da aplicação principal



ActiveX Data Objects II

● Hierarquia de objectos

- **ADODB**: objecto de topo da hierarquia ADO
- **Connection**: para definir e estabelecer ligações com diferentes bases de dados
- **Command**: para executar instruções sobre uma dada ligação
- **Recordset**: para manipular conjuntos de registos
- **Errors**: conjunto de erros ocorridos durante a última operação numa ligação
- **Parameters**: conjunto de parâmetros de um dado comando com parâmetros
- **Fields**: conjunto de atributos (campos) de um dado registo
- **Properties**: conjunto de propriedades específicas de um dado objecto



Objectos Connection I

● Métodos

- `connection.Open ConnectionString`: estabelece uma nova ligação
- `connection.Execute(CommandText,,Options)`: para executar comandos
 - `adCmdTable`: indica que `CommandText` refere o nome de uma tabela ou consulta
 - `adCmdText`: indica que `CommandText` refere uma expressão SQL
- `connection.Close`: fecha a ligação

● Exemplos

```
`estabelecer ligação explícita a uma base de dados do Access
Dim con As New Connection      `a declaração New cria desde logo
                                `uma nova instância do objecto
con.Open "Provider=Microsoft.Jet.OLEDB.4.0;
        Data Source=teste.mdb"
con.Close
`ligação corrente ao Access
Dim con As Connection          `se vou usar Set posso omitir New
Set con = CurrentProject.Connection
con.Close
```

Objectos Connection II

● Exemplos

```
`criar recordset a partir duma ligação explícita (3 exemplos)
Dim con As Connection : Dim rst As Recordset
Set con = ...
Set rst = con.Execute("Clientes", , adCmdTable)
Set rst = con.Execute("[Query sem Parâmetros]", , adCmdTable)
Set rst = con.Execute("SELECT * FROM Clientes", , adCmdText)
rst.Close
con.Close

`criar recordset a partir da ligação corrente ao Access
Dim rst As New Recordset
rst.Open "Clientes", CurrentProject.Connection
rst.Close

`criar recordset a partir duma ligação implícita
Dim rst As New Recordset
rst.Open "Clientes", "Provider=Microsoft.Jet.OLEDB.4.0;
                    Data Source=teste.mdb"

rst.Close
```

Objectos Command

● *Propriedades e métodos*

- `command.ActiveConnection`: ligação associada ao comando
- `command.CommandText`: texto do comando
- `command.CommandType`: tipo de comando (`adCmdTable` / `adCmdText`)
- `command.Parameters`: parâmetros do comando
- `command.Execute`: para executar o comando

● *Exemplos*

`Dim com As New Command`

``comando sem parâmetros`

`com.ActiveConnection = CurrentProject.Connection`

`com.CommandText = "SELECT * FROM Clientes"`

`com.CommandType = adCmdText : Set rst = com.Execute`

``comando com parâmetros`

`com.ActiveConnection = CurrentProject.Connection`

`com.CommandText = "[Query com Parâmetros]"`

`com.CommandType = adCmdTable`

`com.Parameters.Refresh` ``lê os parâmetros`

`com.Parameters("[Limite?]") = 100 : Set rst = com.Execute`

Objectos Recordset I

● **Propriedades de ligação**

- **recordset.Source**: fonte de dados do *recordset*
- **recordset.ActiveConnection**: ligação associada ao *recordset*
- **recordset.CursorType**: tipo de cursor do *recordset*
 - **adOpenForwardOnly**: navegação apenas para a frente
 - **adOpenStatic**: navegação livre
 - **adOpenDynamic**: navegação livre; visualização em tempo real das adições e edições (alterações e eliminações) feitas por outros utilizadores
 - **adOpenKeyset**: navegação livre; visualização em tempo real apenas das edições feitas por outros utilizadores; *batch updates*
- **recordset.LockType**: tipo de bloqueio para edição dos registos do *recordset*
 - **adLockReadOnly**: não permite adição nem edição
 - **adLockPessimistic**: bloqueia o registo corrente no início do processo de edição
 - **adLockOptimistic**: bloqueia o registo corrente apenas quando sincroniza os dados com o servidor (método *Update*)
 - **adLockBatchOptimistic**: bloqueia os vários registos alterados apenas quando sincroniza os dados com o servidor (método *UpdateBatch*)

Objectos Recordset II

● Abrir e Fechar

- recordset.**Open** [**Source**, **ActiveConnection**, **CursorType**, **LockType**, **Options**]: abre um novo *recordset*
- recordset.**Close**: fecha o *recordset*

● Movimentação

- recordset.**Move** **NumRecords**, **Start**: move a posição do registo corrente
- recordset.**MoveFirst**: o mesmo que **.Move 0,adBookmarkFirst**
- recordset.**MoveLast**: o mesmo que **.Move 0,adBookmarkLast**
- recordset.**MoveNext**: o mesmo que **.Move 1,adBookmarkCurrent**
- recordset.**MovePrevious**: o mesmo que **.Move -1,adBookmarkCurrent**
- recordset.**BOF**: indica se a posição do registo corrente é antes do primeiro
- recordset.**EOF**: indica se a posição do registo corrente é depois do último
- recordset.**Bookmark**: marcador que identifica unicamente o registo corrente
- recordset.**RecordCount**: número total de registos

Objectos Recordset III

● Exemplos

```
Dim rst As New Recordset
```

```
Dim mark As Variant
```

```
Dim cont As Integer
```

```
rst.Open "Clientes", CurrentProject.Connection, adOpenStatic,  
        adLockOptimistic, adCmdTable
```

```
...
```

```
mark = rst.Bookmark    'guarda o marcador do registo corrente
```

```
cont = 0
```

```
rst.MoveFirst
```

```
Do Until rst.EOF
```

```
    If rst.Fields("Cliente") Like "A*" Then cont = cont + 1
```

```
    rst.MoveNext
```

```
Loop
```

```
MsgBox "Existem " & cont & " nomes começados por A em "  
        & rst.RecordCount & " clientes!"
```

```
rst.Bookmark = mark    'repõe o registo guardado como corrente
```

```
rst.Close
```


Objectos Recordset IV

● Edição e adição

- recordset.**AddNew**: para adicionar novos registos
- recordset.**Update**: guarda as alterações feitas ao registo corrente
- recordset.**CancelUpdate**: anula as alterações feitas ao registo corrente

● Exemplos

```
rst.MoveLast
Do Until rst.EOF
    rst.Fields("Taxa de Comissão") = 0.01
    rst.Update
    rst.MovePrevious
Loop
rst.AddNew    `cria um novo registo que passa a ser o corrente
rst.Fields("Cliente") = "António"
resp = MsgBox("Quer adicionar?", vbYesNo)
If resp = vbYes Then
    rst.Update
Else
    rst.CancelUpdate    `o registo corrente volta a ser aquele
                        `que era antes da chamada a AddNew
End If
```

Objectos Recordset V

● *Eliminação*

- recordset.Delete: elimina o registo corrente

● *Exemplo*

```
Dim rst As New Recordset
```

```
rst.Open "SELECT * FROM [Clientes] WHERE  
        [Taxa de Comissão]=0", CurrentProject.Connection,  
        adOpenStatic, adLockOptimistic, adCmdText
```

```
rst.MoveFirst
```

```
Do Until rst.EOF
```

```
    rst.Delete    'a posição do registo corrente não se altera
```

```
    rst.MoveNext  'para recolocar válido o registo corrente
```

```
Loop
```

```
rst.Close
```

Objectos Recordset VI

● *Batch Updates*

- recordset.UpdateBatch: guarda as alterações feitas em todos os registos
- recordset.CancelBatch: anula as alterações feitas em todos os registos

● *Exemplo*

```
`para batch updates no Access o cursor precisa de estar do
`lado do cliente (por defeito CursorLocation é adUseServer)
rst.CursorLocation = adUseClient
rst.Open "Clientes", CurrentProject.Connection, adOpenKeyset,
        adLockBatchOptimistic, adCmdTable
rst.AddNew "Cliente", "António"
rst.AddNew "Cliente", "Manuel"
resp = MsgBox("Quer adicionar?", vbYesNo)
If resp = vbYes Then
    rst.UpdateBatch
Else
    rst.CancelBatch    `o registo corrente fica indeterminado
    rst.MoveFirst      `para recolocar válido o registo corrente
End If
rst.Close
```

Objectos Recordset VII

● **Transacções**

- São úteis para quando se pretende garantir o sucesso de uma série de operações que no seu conjunto funcionam como um todo
- Por exemplo, para transferir dinheiro entre duas contas é necessário subtrair uma dada quantidade de um lado e adicionar num outro. Se algumas destas operações falha a outra deixa de fazer sentido e deverá ser anulada
- `connection.BeginTrans`: inicia uma nova transacção
- `connection.CommitTrans`: confirma as alterações feitas durante a transacção e termina a transacção
- `connection.RollbackTrans`: repõe as alterações feitas durante a transacção e termina a transacção
- `connection.Attributes`: atributos para criação automática de transacções
 - `adXactAbortRetaining`: cria uma nova transacção após `RollbackTrans`
 - `adXactCommitRetaining`: cria uma nova transacção após `CommitTrans`

Objectos Recordset VIII

● Exemplo

```
Dim con As Connection
Dim rst As New Recordset
Dim rst_check As New Recordset

Set con = CurrentProject.Connection
con.Attributes = adXactAbortRetaining + adXactCommitRetaining
con.BeginTrans
rst.Open "Clientes", con, adOpenStatic,
        adLockOptimistic, adCmdTable
rst.AddNew "Cliente", "António"
con.CommitTrans
rst.AddNew "Cliente", "Manuel"
rst_check.Open "Clientes", con, adOpenDynamic,
              adLockOptimistic, adCmdTable
rst_check.MoveLast
MsgBox rst_check.Fields("Cliente")           `apresenta "Manuel"
con.RollbackTrans
MsgBox rst_check.Fields("Cliente")           `dá um erro
```

Objectos Recordset IX

● *Filtrar, ordenar e pesquisar registos*

- `recordset.Filter`: critério para filtragem dos registos do *recordset*
- `recordset.Sort`: critério para ordenação dos registos do *recordset*
- `recordset.Find Criteria, SkipRows, SearchDirection`: procura o primeiro registo que verifica o critério e torna-o o registo corrente

● *Exemplos*

```
rst.Filter = "[Taxa de Comissão]=0"
```

```
rst.Filter = "" 'elimina qualquer filtro
```

```
rst.CursorLocation = adUseClient 'necessário para ordenação
```

```
rst.Sort = "[Taxa de Comissão] DESC, Cliente"
```

```
rst.Sort = "" 'elimina qualquer ordenação
```

```
rst.Find "[Taxa de Comissão]=0", 0, adSearchForward
```

```
Do Until rst.EOF 'adSearchBackward como alternativa
```

```
MsgBox rst("Cliente")
```

```
rst.Find "[Taxa de Comissão]=0", 1, adSearchForward
```

```
Loop
```