

Microsoft  
**.net**

## Windows程式設計 ADO.NET資料存取(三)

講師：林賢達  
Peter.lin@imestech.com

Microsoft Professional   Microsoft Solution Developer   Microsoft Application Developer

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## ADO.NET 2.0的新特性

- DataSet和DataTable的增強
- 方便快捷的資料繫結控制項
- 以基底類別為基礎的資料提供者
- 與SQL Server 2005緊密整合

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## DataSet和DataTable的增強

- 提升DataSet的性能與擴展性
  - 改進DataSet的索引引擎，適用於大量的資料處理
  - 支援Binary序列化
- DataTable支援IXMLSerializable介面
  - 提供ReadXML與WriteXML方法來讀寫XML資料
  - 支援Web Service參數傳遞
- DataReader的增強
  - 使用DataReader加載
  - DataReader轉換成DataTable

---

---

---

---

---

---

---

---

Microsoft .net

## DataSet的性能改進

- 重寫DataTable的索引引擎
  - 對大量資料的DataSet，性能有明顯地改善
  - 新增、修改和刪除的速度更快
  - 在具有兩個資料行的DataTable新增500,000筆記錄，使用ADO.NET 1.1耗時約66秒；ADO.NET 2.0僅耗時4秒
  - Fill和Merge的速度更快

---

---

---

---

---

---

---

---

Microsoft .net

## DataSet支援Binary序列化

- 將資料轉換成容易傳遞和儲存的格式
- ADO.NET 1.1，DataSet僅支援XML序列化，不管是使用Binary格式還是SOAP格式
  - 優點：支援Web Service傳遞文字格式的資料
  - 缺點：XML序列化後的檔案較大
- ADO.NET 2.0支援真正的Binary序列化
  - 優點：Binary序列化後的檔案較小
  - 缺點：無法穿越防火牆，即不支援Web Service資料傳遞，支援.NET Remoting

---

---

---

---

---

---

---

---

Microsoft .net

## DataSet支援Binary序列化

- XML序列化與Binary序列化的比較

```
Dim ds As New DataSet()  
Dim da As New SqlDataAdapter("select * from employee",  
Connection物件)  
da.Fill(ds)  
Dim bf As New BinaryFormatter()  
Dim fs As New FileStream("c:\employee.txt", FileMode.Create)  
ds.RemotingFormat = SerializationFormat.Binary  
bf.Serialize(fs, ds)
```

---

---

---

---

---

---

---

---

Microsoft  
.net

### 更為獨立的DataTable

- 針對大部分情況下，一個DataSet只會包含一個DataTable。
- 在ADO.NET 1.1，為了讀寫XML需要將DataTable加入到DataSet，才具有此功能
- 在ADO.NET 2.0，DataTable可以直接讀寫XML，毋須倚賴DataSet
  - ReadXMLSchema和WriteXMLSchema
  - ReadXML和WriteXML

---

---

---

---

---

---

---

---

Microsoft  
.net

### 更為獨立的DataTable

- 提供Load方法，將實作IDataReader介面的物件載入DataTable，也就是可將DataReader物件轉換成DataTable  
MyDataTable.Load(IDataReader物件)
- 提供Merge方法，合併另一個DataTable  
MyDataTable.Merge(DataTable物件)

---

---

---

---

---

---

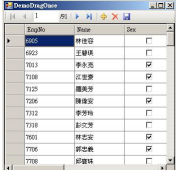

---

---

Microsoft  
.net

### Drag-Once 資料繫結

- VS 2005提供的一種簡單快速的資料繫結方式。只需要將資料庫物件或自訂物件拖曳至Windows表單或控制項上即可完成資料的瀏覽、新增、修改、刪除等功能

---

---

---

---

---

---

---

---

Microsoft .net

## Drag-Once 資料繫結

- Drag-Once 資料繫結的物件類型有
  - Database：資料表、檢視(View)和預存程序
  - Web Service
  - Object：自定物件，適於多層式(N-tier)應用程式
- 開發步驟
  - 建立資料來源
  - 指定顯示模式，DataGridView還是Detail

---

---

---

---

---

---

---

---

Microsoft .net

## Drag-Once 資料繫結

- 將各種不同的資料來源抽象為BindingSource

---

---

---

---

---

---

---

---

Microsoft .net

## 資料繫結相關元件

- BindingSource 元件
  - 扮演控制項與資料來源之間間接取值(Indirection)層
  - 與資料來源進行互動，提供巡覽、新增、修改和刪除等操作，以及排序和過濾資料

---

---

---

---

---

---

---

---

## 資料繫結相關元件

### ■ BindingNavigator控制項

- 提供瀏覽、新增、修改和刪除資料等操作
- 對應至某個BindingSource元件



---

---

---

---

---

---

---

---

## DataSet Designer

- 設計DataSet的資料結構，支援視覺化的方式建立和修改DataTable與DataRelation
- 提供新的TableAdapter類別
  - designer-generated components，不存在於.NET 類別庫
  - 每一個DataTable會對應一個TableAdapter，執行查詢和預存程序，填充DataTable和更新資料來源

---

---

---

---

---

---

---

---

## TableAdapter類別

### ■ 重要成員

- Fill(): 填充資料到DataTable (非DataSet)
- Update(): 將DataSet的異動資料寫回資料庫
- GetData(): 取出填充資料的DataTable
- Insert(): 新增一筆記錄
- Delete(): 刪除一筆記錄
- ClearBeforeFill: 填充前是否清空DataTable，可支援多次填充

---

---

---

---

---

---

---

---

Microsoft .net

## 以基底類別為基礎的資料提供者

- Data Provider模型
  - ADO.NET 1.1 Data Provider是以介面為基礎
  - ADO.NET 2.0 Data Provider是以基底類別為基礎

---

---

---

---

---

---

---

---

---

---

Microsoft .net

## 以基底類別為基礎的資料提供者

SQL Data Provider	基底類別	介面
System.Data.SqlClient	System.Data.Common	System.Data
SqlConnection	DbConnection	IDbConnection
SqlCommand	DbCommand	IDbCommand
SqlParameter	DbParameter	IDbDataParameter
SqlDataReader	DbDataReader	IDbDataReader
SqlDataAdapter	DbDataAdapter	IDbDataAdapter
SqlConnectionStringBuilder	DbConnectionStringBuilder	
... and so on.		

---

---

---

---

---

---

---

---

---

---

Microsoft .net

## 以基底類別為基礎的資料提供者

- 優點
  - 使用相同的程式存取不同的資料庫，從商務邏輯層抽離出來，做為獨立的資料存取層
  - 使用Db開頭的基底類別存取資料庫，不需要針對特定的資料庫來開發應用程式

---

---

---

---

---

---

---

---

---

---

Microsoft  
.net

## 以基底類別為基礎的資料提供者

- 建立通用的Connection物件
 

```
Public Function GetConnectionBase() As DbConnection
  Dim cnn As DbConnection = Nothing
  Dim Provider As String = My.Settings.DBProvider
  Select Case Provider
    Case "sqlserver"
      cnn = New SqlConnection()
    Case "oracle"
      cnn = New OracleConnection()
    Case Else
      cnn = New OleDbconnection()
  End Select
  Return cnn
End Function
```

---

---

---

---

---

---

---

---

---

---

Microsoft  
.net

## Data Provider Factory 模式

- 新增System.Data.Common命名空間
- DBProviderFactory類別
  - 在執行時期識別資料來源，自動建立對應的資料提供者
  - 支援OleDbFactory、OdbcFactory、SQLClientFactory和OracleClientFactory

---

---

---

---

---

---

---

---

---

---

Microsoft  
.net

## Data Provider Factory 模式

- Provider Factories 模式

```

graph TD
    subgraph Factories
        direction LR
        S[SqlClientFactory]
        O[OracleClientFactory]
        E[OleDbFactory]
    end
    S -- Inherits --> D[DbProviderFactory]
    O -- Inherits --> D
    E -- Inherits --> D
    G[DbProviderFactories.GetFactory(提供者類型名稱)] -- Initialize --> D
  
```

---

---

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## DbProviderFactories類別

- 重要方法
  - GetFactoryClasses(), 傳回一個DataTable物件，取出所有已註冊的Data Provider，參考machine.config
  - GetFactory(), 根據提供者名稱(InvariantName)或DataRow物件，傳回一個DbProviderFactory物件

Name	Description	InvariantName	AssemblyQualified Name
DbOleDbProvider	Net Framework ...	System.Data.OleDb	System.Data.OleDb.Dbb...
DbOdbcData Provider	Net Framework ...	System.Data.Odbc	System.Data.Odbc.Odb...
DbOdbcClient Data Provider	Net Framework ...	System.Data.OdbcClient	System.Data.OdbcClie...
SqlClient Data Provider	Net Framework ...	System.Data.SqlClient	System.Data.SqlClient.S...

---

---

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## DbProviderFactory類別

- 重要方法
  - 以Create開頭的一系列用於建立對應Data Provider的相關物件的方法。使用CreateConnection建立Connection物件...

```

classDiagram
    class DbProviderFactory {
        + CreateCommand()
        + CreateConnection()
        + CreateDataAdapter()
        + CreateDataSourceEnumerator()
        + CreatePermission()
        + CreatePermission()
    }
    class SqlClientFactory
    class OleDbFactory
    class OleDbClientFactory
    DbProviderFactory <|-- SqlClientFactory
    DbProviderFactory <|-- OleDbFactory
    DbProviderFactory <|-- OleDbClientFactory
  
```

---

---

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## 與SQL Server 2005緊密整合

- 非同步叫用(Asynchronous Invoke)
- 多個活動結果集(MARS, Multiple Active Result Set), 支援開啟多個DataReader
- 大量資料的複製(Bulk Copy Operations)

---

---

---

---

---

---

---

---

---

---



### Asynchronous Invoke

- 對於大量資料的批次新增和更新時，使用非同步處理，可以不用等待所有資料異動都處理完畢，即可操作使用者介面或進行下一個處理，可有效改善使用者體驗
- 目前僅有SQL Data Provider直接支援非同步方式存取資料
- 其他的Data Provider仍可使用非同步API來實現非同步處理

---

---

---

---

---

---

---

---

### Asynchronous Invoke

- 使用SqlCommand物件來實現非同步處理

同步方法	非同步方法	
ExecuteNoQuery	BeginExecuteNoQuery	EndExecuteNoQuery
ExecuteReader	BeginExecuteReader	EndExecuteReader
ExecuteXmlReader	BeginExecuteXmlReader	EndExecuteXmlReader
ExecuteScalar	-	-

---

---

---

---

---

---

---

---

### 非同步叫用

- 在連接字串加入"async=true"，指定connection物件支援非同步處理
- 在連接字串中使用"async=true"會耗損較多的伺服器資源，微軟官方建議使用兩個連接字串分別執行同步和非同步處理

---

---

---

---

---

---

---

---

Microsoft .net

## Asynchronous Invoke

- 使用Begin方法傳入參數，傳回IAsyncResult物件，可用於追蹤非同步處理的執行狀態
- 使用End方法，傳回輸出參數或執行結果

```
Dim ar As IAsyncResult = cmd.BeginExecuteReader()  
'執行其他處理  
Dim dr As SqlDataReader = cmd.EndExecuteReader(ar)  
'取出執行結果
```

---

---

---

---

---

---

---

---

Microsoft .net

## Asynchronous Invoke

- IAsyncResult介面
  - AsyncState：使用者自訂的物件，Object類型
  - AsyncWaitHandle：設定非同步處理的等候方式  
只要其中一個非同步處理完成，還是等待所有的非同步處理都完成
  - CompletedSynchronously：所有非同步處理是否同時完成
  - IsCompleted：非同步處理是否完成，可根據此屬性執行下一個處理

---

---

---

---

---

---

---

---

Microsoft .net

## Asynchronous Invoke

- 非同步處理的等待模式：輪詢、等待和回呼
  - 等候(Wait)模式又分成WaitAll和WaitAny
  - 輪詢(Polling)模式

```
Dim ar As IAsyncResult = cmd.BeginExecuteReader()  
Do While Not ar.IsCompleted  
'執行其他處理  
Loop  
Dim dr As SqlDataReader = cmd.EndExecuteReader(ar)
```

---

---

---

---

---

---

---

---

Microsoft .net

## Asynchronous Invoke

- 非同步處理的等待模式(續)
  - WaitAll模式

```
Dim ar1 As IAsyncResult = cmd1.BeginExecuteReader()
Dim ar2 As IAsyncResult = cmd2.BeginExecuteReader()
Dim HandleArray(1) As WaitHandle
HandleArray(0) = ar1
HandleArray(1) = ar2
WaitHandle.WaitAll(HandleArray, 5000, True)
Dim dr1 As SqlDataReader = cmd1.EndExecuteReader(ar1)
Dim dr2 As SqlDataReader = cmd2.EndExecuteReader(ar2)
```

---

---

---

---

---

---

---

---

Microsoft .net

## Asynchronous Invoke

- 非同步處理的等待模式(續)
  - WaitAny模式

```
Dim ar1 As IAsyncResult = cmd1.BeginExecuteReader()
Dim ar2 As IAsyncResult = cmd2.BeginExecuteReader()
Dim HandleArray(1) As WaitHandle
HandleArray(0) = ar1
HandleArray(1) = ar2
```

---

---

---

---

---

---

---

---

Microsoft .net

## Asynchronous Invoke

- 非同步處理的等待模式(續)
  - WaitAny模式(續)

```
For i As Integer = 0 To HandleArray.Length - 1
    Dim Index As Integer
    Index = WaitHandle.WaitAny(HandleArray, 5000, True)
    Select Case Index
        Case 0
            Dim dr1 As SqlDataReader = cmd1.EndExecuteReader(ar1)
        Case 1
            Dim dr2 As SqlDataReader = cmd2.EndExecuteReader(ar2)
    End Select
Next
```

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## Asynchronous Invoke

- 非同步處理的等待模式(續)
  - Callback模式
 

```
cmd.BeginExecuteReader(New AsyncCallbck(AddressOf
            CallbackHandler), cmd)

Private Sub CallbackHandler(ar As IAsyncResult)
    Dim dr As SqlDataReader
    dr = CType(ar.AsyncState, SqlCommand).EndExecuteReader(ar)
End Sub
```

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## Multiple Active Result Sets (MARS)

- 目前僅支援SQL Server 2005
- ADO.NET 1.1的SqlConnection僅能同時傳回一個SqlDataReader，即一次只能執行一個查詢，執行下一個查詢前，須關閉前一個SqlDataReader
- ADO.NET 2.0的SqlConnection允許同時傳回多個SqlDataReader，即同時執行多個查詢
- 在連接字串加入"MultiActiveResultsets=true"，指定connection物件支援MARS

---

---

---

---

---

---

---

---

Microsoft  
**.net**

## Bulk Copy Operations

- 目前僅支援SQL Data Provider
- 使用SqlBulkCopy類別快速地寫入大量資料，包含寫入DataRowCollection、DataTable和SqlDataReader等物件
 

```
Dim bcp As New SqlBulkCopy(連接字串)
bcp.DestinationTableName="Table名稱"
bcp.WriteToServer(DataTable物件)
```

---

---

---

---

---

---

---

---

Microsoft  
.net

下課鐘響，回家真好...



---

---

---

---

---

---

---