

Microsoft  
.net

## Windows程式設計 ADO.NET資料存取(四)

講師：林賢達  
Peter.lin@imestech.com

Microsoft Professional   Microsoft Solution Developer   Microsoft Application Developer

---

---

---

---

---

---

---

---

Microsoft  
.net

### 課程大綱

- 性能分析與設計原則
- 使用Connection的最佳實務
- 使用Command的最佳實務
- 使用DataSet的最佳實務
- 使用DataReader的最佳實務
- 使用DataAdapter的最佳實務

---

---

---

---

---

---

---

---

Microsoft  
.net

### 性能分析

- 效率差的SQL語法，參考  
<http://www.ithome.com.tw/plog/index.php?op=ViewArticle&articleId=5421&blogId=620>
- 一次讀取大量的資料
- 同時開啟太多的連接
- 沒有明確地釋放資源
- 錯誤使用交易處理
- 過於"正規化"的資料表設計
- 沒有建立索引或不當的索引設計

---

---

---

---

---

---

---

---

### 設計原則

- 越晚開啟連接，越早關閉連接；若須連續使用時，應避免開開關關
- 對於實作IDisposable介面的類別，例如 Connection、DataReader等。當不再使用該物件時，應使用Dispose方法來釋放配置的 Unmanaged資源
- 儘可能使用預存程序，第一次編譯後便儲存在記憶體中，以便重覆使用

---

---

---

---

---

---

---

---

### 設計原則

- 做好例外處理，避免洩漏機密訊息
- 正確地使用交易處理，避免執行過多的SQL語法
- 在設計資料存取層時，需同時兼顧可維護性與性能

---

---

---

---

---

---

---

---

### 使用Connection的最佳實務

- 使用連接集區(Connection Pool)
- 使用DataAdapter最佳化連接
- 即時關閉Connection
- 處理連接例外

---

---

---

---

---

---

---

---

Microsoft  
.net

## 使用連接集區

- 連接集區(Connection Pool)
  - 建立Connection需要花費時間和資源
  - 使用連接集區保留使用完畢的Connection；下次從連接集區取出即可直接使用

```

graph TD
    Open[Open] --> Connection{Connection?}
    Connection -- No --> Create[建立Connection]
    Connection -- Yes --> Command[下達command]
    Create --> Result[傳回執行結果]
    Command --> Result
    Result -.->|Close| Connection
    Note["(相同的連接字串)"]
  
```

---

---

---

---

---

---

---

---

Microsoft  
.net

## 使用連接集區

- 在連接字串設定相關的參數
  - pooling = True  
(SqlConnection和OleDbConnection預設為True)
  - connection lifetime = 5  
(以秒計算，預設為0，使用最長的逾時)
  - min pool size = 5 (預設為0)
  - max pool size = 20 (預設為100)

---

---

---

---

---

---

---

---

Microsoft  
.net

## 使用連接集區

- 使用完整的連接字串來識別連接集區
  - Integrated Security=SSPI;Database=testdb
  - Integrated Security=SSPI;Database=**T**estdb
  - Integrated Security=SSPI;Database=testdb
 上面的連線字串會分別建立不同的連接集區

---

---

---

---

---

---

---

---

### 即時關閉Connection

- 用完Connection後，務必呼叫Close方法關閉之，釋放伺服器資源
- 儘可能使用Using區塊。一旦離開Using區塊後，會自動呼叫Dispose方法摧毀該物件，釋放資源

---

---

---

---

---

---

---

---

### 處理連接例外

- DataException類別：使用ADO.NET元件發生錯誤時所拋出的例外
- DBConcurrencyException類別：執行新增、修改和刪除資料時，若受影響的紀錄筆數為零，由DataAdapter所拋出的例外
- SqlException類別：當SQL Server傳回警告或錯誤時拋出的例外

---

---

---

---

---

---

---

---

### 處理連接例外

- SqlException類別(續)
  - 至少包含一個SqlError物件
  - SqlError物件的Class屬性用於指出此錯誤的嚴重等級
    - 01~10：提示訊息
    - 11~16：發生使用者錯誤
    - 17~25：發生軟體或硬體錯誤，其中17~19 SqlConnection仍保持開啟狀態，而20~25則伺服器會自動關閉連接

---

---

---

---

---

---

---

---

## 使用Command的最佳實務

- 使用適當的Execute方法
- 盡可能使用預存程序(Stored Procedure)
- 正確使用交易處理

---

---

---

---

---

---

---

---

## 使用適當的Execute方法

- 執行唯讀且順向讀取的查詢，請使用ExecuteReader方法
- 執行Count(\*)、Sum(Qty)或Avg(Qty)等傳回單值的查詢，請使用ExecuteScalar方法
- 執行Insert、Update和Delete等資料異動的命令或使用輸出參數傳回結果，請使用ExecuteNonQuery方法

---

---

---

---

---

---

---

---

## 使用DataSet的最佳實務

- 使用時機
  - 需要離線存取資料
  - 使用DataAdapter批次修改的大量資料
  - 處理來自多個不同資料來源的資料
  - 在多層式(N-tier)應用程式的各層之間傳遞或快取資料
  - 對重覆使用而且不會經常變動的資料，可藉由DataSet快取(Cache)資料來改善性能
  - 使用DataGridView控制項展現資料表的內容

---

---

---

---

---

---

---

---

Microsoft .net

## 使用DataReader的最佳實務

- 使用時機
  - 只要唯讀且順向快速地存取資料
  - 使用ArrayList快取資料
  - 對於記憶體受限的系統，需要處理大量的資料
- 使用CommandBehavior
  - 每次呼叫Read方法時，DataReader會將下一筆記錄載入記憶體中，便可隨機讀取任意欄位值。若不需要隨機讀取，而且為了提高性能可在呼叫ExecuteReader方法時，應設定CommandBehavior為SequentialAccess只載入欲讀取的資料

---

---

---

---

---

---

---

---

Microsoft .net

## 使用DataReader的最佳實務

- 使用CommandBehavior(續)
  - 讀取大型二進位值時，例如圖形或大量文字，應指定CommandBehavior為SequentialAccess，然後使用GetBytes或GetChars方法從某個開始位置讀取多少位元組的資料
  - 若需傳回DataReader時，在呼叫ExecuteReader方法時，應將CommandBehavior設定為CloseConnection，以便關閉DataReader時能一併關閉資料庫連接

---

---

---

---

---

---

---

---

Microsoft .net

## 使用DataAdapter的最佳實務

- 在事先未開啟連線的情況，呼叫Fill和Update方法會自動為相關Command物件開啟資料庫連接
- DataAdapter會在使用完畢後，自動關閉自行開啟的連接；但若是已事先開啟的連接，則用完後不會自動關閉
- 若只執行一次，使用自動開啟連接方式，其效率較佳；若需連續執行多次，則使用事先開啟連接方式，其效率較佳

---

---

---

---

---

---

---

---

## CommandBuilder物件的使用

- 如果DataTable只對應資料庫的一個Table，而且此Table定義了主索引，則可以利用CommandBuilder物件自動產生DataAdapter的InsertCommand、UpdateCommand和DeleteCommand
- 為了產生Insert、Update和Delete的SQL語法，CommandBuilder會自動使用SelectCommand屬性來檢索所需的資料結構(Schema)
- 如果SelectCommand發生改變時，應使用CommandBuilder的RefreshSchema方法重新產生其他的Command

---

---

---

---

---

---

---

---

## 下課鐘響，回家真好...



---

---

---

---

---

---

---

---