

## 18.2 資料繫結

.NET Framework 所提供的大部份控制項皆有資料繫結(DataBinding)的功能。例如 Label、TextBox、CheckBox、ComboBox、DataGridView...等控制項。當控制項做資料繫結的動作後，該控制項即會顯示所查詢的資料記錄。我們將以資料繫結的控制項分成三種類型來介紹：

**Case 01** 使用 Label、LinkLabel、Button、CheckBox、RadioButton、TextBox 控制項做資料繫結，一次只能在該控制項上面顯示一筆記錄中的某個欄位的內容。此種資料繫結的寫法如下：

```
控制項物件名稱.DataBindings.Add ("屬性", 資料來源, "資料成員");
```

- ① 屬性：指定所要繫結的控制項屬性。
- ② 資料來源：指定 DataSet、DataView、DataTable 物件資料來源。
- ③ 資料成員：指定要繫結的資料來源欄位。

譬如：使用 txtName 文字方塊控制項的 Text 屬性與 ds 中「員工」資料表的「姓名」欄位做資料繫結，結果表單上面的 txtName 文字方塊控制項會顯示「姓名」欄位的第一筆資料。其寫法如下：

```
txtName.DataBindings.Add ("Text", ds, "員工.姓名") ;
```

編號	姓名	職稱	電話	薪資
1	大頭龍	技術副總	04-12345671	50000
2	小句子	技術副理	049-1245787	30000
3	顏小星	講師	049-3698521	40000
4	廖小春	工程師	049-1452369	10000
5	黃小慧	技術副理	04-78865412	20000
6	Jack Wu	業務	02-13872336	30000
*	NULL	NULL	NULL	NULL

**Case 02** 使用 ComboBox、ListBox、CheckedListBox 控制項做資料繫結，可以顯示所有記錄的某一個欄位資料。資料繫結語法如下：

```
控制項物件名稱.DataSource = 資料來源；
控制項物件名稱.DisplayMember = 資料成員；
```

- ① 資料來源：指定 DataSet、DataView、DataTable 物件資料來源。
- ② 資料成員：指定要繫結的資料來源欄位。

譬如：使用名稱為 `cboId` 的下拉式清單和「員工」資料表的「編號」欄位做資料繫結，顯示「編號」欄位的所有資料，其寫法和關係圖如下：

```
cboId.DataSource = ds;
cboId.DisplayMember = "員工.員工編號";
```

The diagram illustrates the data binding process. A form titled 'Form1' contains a dropdown menu control named 'cboId'. The dropdown menu is currently displaying a list of numbers: 6, 5, 4, 3, 2, 1. To the right of the dropdown are text boxes for '職稱' (set to '業務') and '薪資' (set to '30000'). Below the form is a data grid with the following data:

編號	姓名	職稱	電話	薪資
1	大頭龍	技術副總	04-12345671	50000
2	小句子	技術副理	049-1245787	30000
3	顏小星	講師	049-3698521	40000
4	廖小喬	工程師	049-1452369	10000
5	黃小憲	技術副理	04-78965412	20000
6	Jackwu	業務	02-14852336	30000
*	NULL	NULL	NULL	NULL

A callout box labeled '資料繫結' (Data Binding) points to the dropdown menu and the '編號' column of the data grid, indicating the connection between the control and the data source.

**Case 03** 使用 DataGridView 控制項做資料繫結可顯示所有記錄，資料繫結語法：

```
控制項物件名稱.DataSource = 資料來源；
控制項物件名稱.DataMember = 資料成員；
```

- ① 資料來源：指定 DataSet、DataView、DataTable 物件資料來源。
- ② 資料成員：指定要繫結的 DataTable。

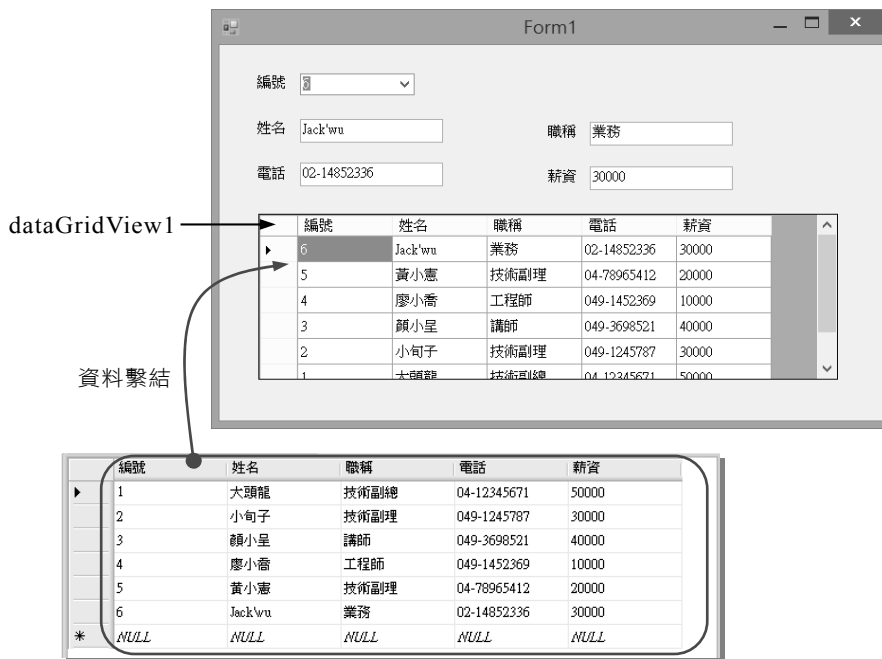
若 DataGridView 控制項的 DataSource 屬性設定成 DataView 或 DataTable 物件，則 DataMember 屬性可以不用設定。

譬如：使用 dataGridView1 控制項和 ds 資料集物件中的「員工」資料表做資料繫結，來顯示「員工」資料表的所有資料，其寫法如下：

```
dataGridView1.DataSource = ds ;
dataGridView1.DataMember = "員工" ;
```

上述兩行敘述可改寫成下面任一行敘述：

```
dataGridView1.DataSource = ds.Tables["員工"];
或
dataGridView1.DataSource = ds.Tables["員工"].DefaultView ;
```

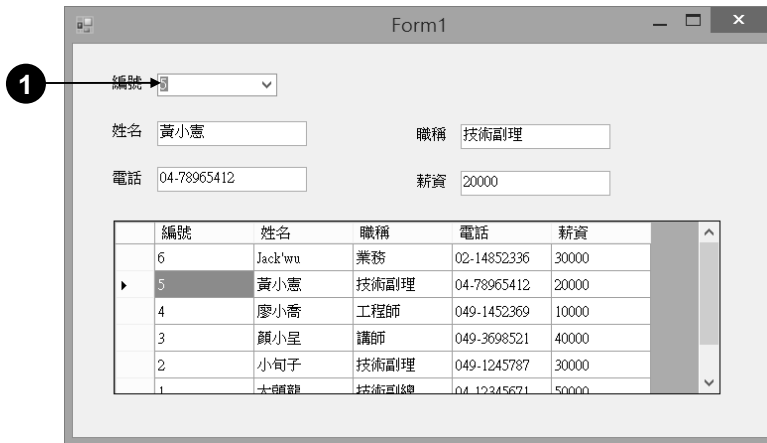


### 範例：DataBindingDemo1.sln

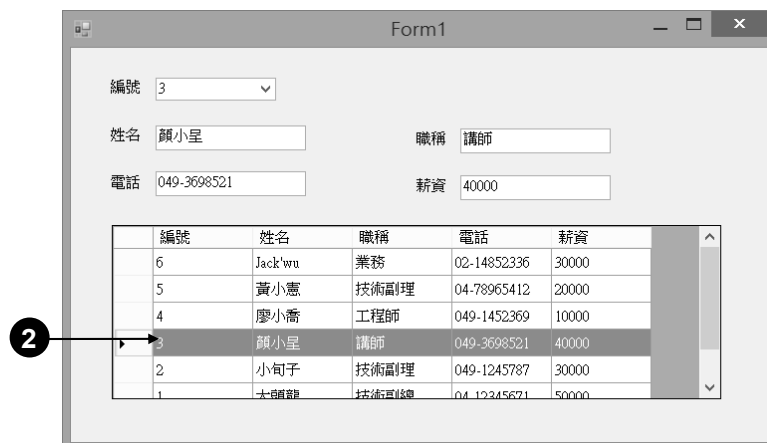
練習將「員工」資料表的欄位記錄與 TextBox、ComboBox、DataGridView 控制項做資料繫結。如下圖，當您選取「編號」下拉式清單控制項的編號時，會將該筆資料對應的欄位內容分別放入姓名、電話、職稱、薪資的文字方塊內。

同樣地，當您選取 DataGridView 控制項中任何一筆記錄，則在編號下拉式清單，姓名、電話、職稱、薪資的文字方塊內會顯示 DataGridView 控制項所選取的那一筆記錄。

### 執行結果



▲編號下拉式清單選編號 5 記錄的畫面



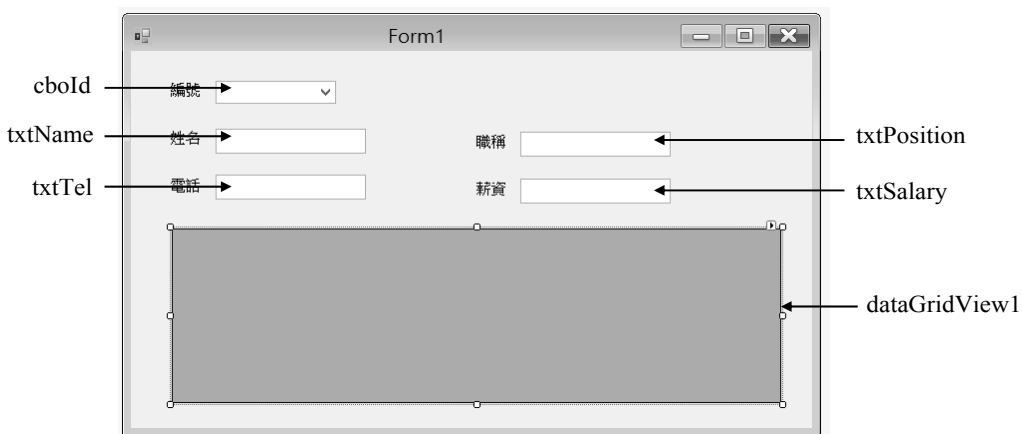
▲dataGridView1 選編號 3 記錄的畫面

**操作步驟**

**Step 01** 複製資料庫

請將書附光碟「資料庫」資料夾下的 ch18DB.mdf 和 ch18DB\_log.ldf 複製到目前製作專案的 bin\Debug 資料夾下，使 ch18DB.mdf 資料庫與範例執行檔在相同路徑下。

**Step 02** 建立表單輸出入介面



**Step 03** 撰寫程式碼

**程式碼** FileName:Form1.cs

```

01 using System.Data;
02 using System.Data.SqlClient;
03
04 namespace DataBindingDemo1
05 {
06     public partial class Form1 : Form
07     {
08         public Form1()
09         {
10             InitializeComponent();
11         }
12
13         private void Form1_Load(object sender, EventArgs e)
14         {

```

```

15
16     using (SqlConnection cn = new SqlConnection())
17     {
18         cn.ConnectionString = @"Data Source=(LocalDB)\v11.0;" +
19             "AttachDbFilename=|DataDirectory|ch18DB.mdf;" +
20             "Integrated Security=True";
21         SqlDataAdapter daEmployee = new SqlDataAdapter
22             ("SELECT * FROM 員工 ORDER BY 編號 DESC", cn);
23         DataSet ds = new DataSet();
24         daEmployee.Fill(ds, "員工");
25         // ComboBox 控制項資料繫結
26         cboId.DataSource = ds;
27         cboId.DisplayMember = "員工.編號";
28         // TextBox 控制項資料繫結
29         txtName.DataBindings.Add("Text", ds, "員工.姓名");
30         txtTel.DataBindings.Add("Text", ds, "員工.電話");
31         txtPosition.DataBindings.Add("Text", ds, "員工.職稱");
32         txtSalary.DataBindings.Add("Text", ds, "員工.薪資");
33         // DataGridView 控制項資料繫結
34         dataGridView1.DataSource = ds;
35         dataGridView1.DataMember = "員工";
36     }
37 }
38 }

```

### 說明

1. 第 25-26 行：將 cboId 下拉式清單繫結至 ds 物件中「員工」DataTable 的「員工.編號」欄位。
2. 第 28-31 行：將 txtName、txtTel、txtPosition、txtSalary 文字方塊的 Text 屬性繫結至「員工」DataTable 的對應欄位。
3. 第 33-34 行：將 dataGridView1 控制項繫結到 ds 的「員工」DataTable 物件。



## 18.3 如何將 DataTable 進行關聯

第 16 章我們使用資料工具建立簡單的關聯式資料庫應用程式，而本節我們將介紹如何使用 DataSet 物件的 Relations 集合，將記憶體 DataSet 內的多個 DataTable 進行關聯，以建立關聯式的資料庫應用程式，步驟如下：

**Step 01** 建立 DataSet 物件，再使用 DataAdapter 物件執行 SQL 語法，並使用 DataAdapter 物件的 Fill 方法將 DataTable 物件(查詢資料的結果)放到 DataSet 物件中。因為要製作關聯式資料庫，所以 DataSet 物件必須要有兩個以上的 DataTable 物件。寫法如下：

```
DataSet ds = new DataSet();
SqlDataAdapter da1 = new SqlDataAdapter("SQL 語法 1", cn);
da1.Fill(ds, "dt1"); // ds 含有 DataTable 物件 dt1
SqlDataAdapter da2 = new SqlDataAdapter("SQL 語法 2", cn);
da2.Fill(ds, "dt2"); // ds 含有 DataTable 物件 dt2
```

**Step 02** 使用 DataSet 物件的 Relations 集合物件的 Add 方法加入一個關聯物件，並且設定兩個 DataTable 物件中哪兩個欄位要進行關聯。寫法如下：

```
ds.Relations.Add("關聯名稱",
    ds.Tables["dt1"].Columns["dt1 要關聯的 Primary Key 欄位名稱"],
    ds.Tables["dt2"].Columns["dt2 要關聯的 Foreign Key 欄位名稱"]);
```

**Step 03** 接著將主資料表 dt1 繫結至 dataGridView1，而關聯物件「dt1.關聯名稱」繫結至 dataGridView2，此時你只要選取 dataGridView1 中的某一筆記錄，則 dataGridView2 即會顯示對應的記錄。寫法如下：

```
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "dt1";
dataGridView2.DataSource = ds;
dataGridView2.DataMember = "dt1.關聯名稱";
```



## 範例：RelationsDemo1.sln

製作可以將記憶體內的兩個 DataTable 關聯的資料庫應用程式。將 Northwind.mdf 資料庫「產品類別」及「產品資料」兩個資料表放入兩個 DataTable 後，接著再將這兩個 DataTable 物件的「類別編號」欄位進行關聯，此時選取上方 DataGridView 中產品類別的某一筆記錄時，則下方的 DataGridView 即會顯示對應的產品資料所有記錄。

### 執行結果



類別編號	類別名稱	說明	圖片
1	飲料	軟性飲料,咖啡,...	
2	調味品	甜酸醬,配料,塗...	
3	點心	甜點心,糖果,甜...	

產品編號	產品	供應商編號	類別編號	單位數量
3	蕃茄醬	1	2	每箱12瓶
4	鹽巴	2	2	每箱12瓶
5	麻油	2	2	每箱12瓶
6	醬油	3	2	每箱12瓶
8	胡椒粉	3	2	每箱30盒
15	味素	6	2	每箱30盒
44	蠔油	20	2	每箱24瓶

▲上方產品類別 DataGridView 選取類別編號 2 記錄所顯示的畫面



類別編號	類別名稱	說明	圖片
4	日用品	起司	
5	穀類/麥片	麵包,餅干,麵糰,...	
6	肉/家禽	肉品	
7	特製品	水果乾及豆腐	

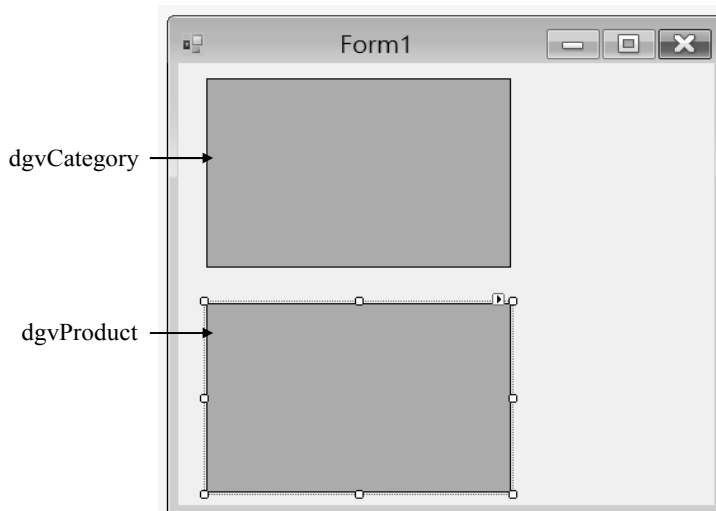
產品編號	產品	供應商編號	類別編號	單位數量
22	再來米	9	5	每袋3公斤
23	燕麥	9	5	每袋3公斤
42	糙米	20	5	每袋3公斤
52	三合一麥片	24	5	每箱24包
56	白米	26	5	每袋3公斤
57	小米	28	5	每袋3公斤
64	黃豆	12	5	每袋3公斤

▲上方產品類別 DataGridView 選取類別編號 5 記錄所顯示的畫面



**操作步驟****Step 01** 複製資料庫

請將書附光碟「資料庫」資料夾下的 Northwind.mdf 和 Northwind\_log.ldf 複製到目前製作專案的 bin\Debug 資料夾下，使 Northwind.mdf 資料庫與範例執行檔在相同路徑下。

**Step 02** 建立表單輸出入介面**Step 03** 撰寫程式碼**程式碼** FileName:Form1.cs

```
01 using System.Data;
02 using System.Data.SqlClient;
03
04 namespace RelationsDemol
05 {
06     public partial class Form1 : Form
07     {
08         public Form1()
09         {
10             InitializeComponent();
11         }
12
13         private void Form1_Load(object sender, EventArgs e)
```

```

14      {
15          using (SqlConnection cn = new SqlConnection())
16          {
17              cn.ConnectionString = @"Data Source=(LocalDB)\v11.0;" +
18                  "AttachDbFilename=|DataDirectory|Northwind.mdf;" +
19                  "Integrated Security=True";
20              DataSet ds = new DataSet();
21              SqlDataAdapter daCategory = new SqlDataAdapter
22                  ("SELECT * FROM 產品類別", cn);
23              daCategory.Fill(ds, "產品類別");
24              SqlDataAdapter daProduct = new SqlDataAdapter
25                  ("SELECT * FROM 產品資料", cn);
26              daProduct.Fill(ds, "產品資料");
27              ds.Relations.Add("FK_產品資料_產品類別",
28                  ds.Tables["產品類別"].Columns["類別編號"],
29                  ds.Tables["產品資料"].Columns["類別編號"]);
30              dgvCategory.DataSource = ds;
31              dgvCategory.DataMember = "產品類別";
32              dgvCategory.Dock=DockStyle.Top;//dgvCategory 停駐在表單上方
33              dgvProduct.DataSource = ds;
34              dgvProduct.DataMember = "產品類別.FK_產品資料_產品類別";
35              dgvProduct.Dock=DockStyle.Fill;//dgvProduct 填滿整個表單
36          }
37      }
38  }
39  }

```

## 說明

1. 第 21-22 行：建立 SqlDataAdapter 物件 daCategory，該物件用來查詢「產品類別」資料表所有記錄，並將查詢結果填入 ds 物件。
2. 第 23-24 行：建立 SqlDataAdapter 物件 daProduct，該物件用來查詢「產品資料」資料表所有記錄，並將查詢結果填入 ds 物件。
3. 第 25 行：將 ds 物件內的產品類別的「類別編號」(主鍵)關聯到產品資料的「類別編號」(外來鍵)，關聯物件名稱設為「FK\_產品資料\_產品類別」。
4. 第 30 行：將 dgvProduct 控制項的 DataMember 屬性設為「FK\_產品資料\_產品類別」關聯物件，因此該控制項會顯示對應的產品資料。

## 18.4 使用 BindingManagerBase 巡覽資料表記錄

透過 BindingManagerBase 物件的 BindingContext 集合物件可設定想要巡覽的資料記錄，例如下圖表單有 **第一筆**、**上一筆**、**下一筆**、**最末筆** 鈕，透過這四個按鈕可以動態巡覽每一筆記錄。

如下是使用 BindingManagerBase 物件的步驟：

**Step 01** 宣告 BindingManagerBase 物件。寫法如下：

```
BindingManagerBase bm ;
```

**Step 02** 建立 Binding 物件並指定所要資料繫結的控制項屬性、資料來源(DataSet 物件)及資料成員(DataTable 名稱與欄位名稱)。寫法如下：

```
Binding myBinding1 = new Binding("屬性", 資料來源, "資料成員");
Binding myBinding2 = new Binding("屬性", 資料來源, "資料成員");
.....
Binding myBindingN = new Binding("屬性", 資料來源, "資料成員");
```

**Step 03** 使用控制項 DataBindings 集合屬性的 Add 方法加入 Binding 資料繫結物件。寫法如下：

```
控制項名稱 1.DataBindings.Add(myBinding1);
控制項名稱 2.DataBindings.Add(myBinding2);
.....
控制項名稱 N.DataBindings.Add(myBindingN);
```

- Step 04** 使用控制項的 `BindingContext` 方法建立 `BindingManagerBase` 物件，`BindingContext` 方法必須傳入資料來源 (`DataSet` 物件) 及資料成員 (`DataTable` 名稱)。寫法如下：

```
bm = this.BindingContext(資料來源, "資料成員");
```

- Step 05** 完成上面設定之後，接著您可透過下表 `BindingManagerBase` 物件所提供的成員來操作資料表記錄：

成員	說明
Position 屬性	取得目前記錄的位置 0 ~ Count - 1。
Count 屬性	取得資料記錄總筆數。
AddNew()方法	加入一筆空記錄
RemoveAt(index)方法	刪除第 index-1 筆資料記錄。
EndCurrentEdit()方法	結束目前編輯，將控制項上的資料寫回 DataSet 內。

### 範例：BindingManagerBase1.sln

使用上面介紹的步驟，使用 `BindingManagerBase` 物件與各控制項做資料繫結，讓使用者可以動態巡覽 `ch18DB.mdf` 中「會員」資料表上一筆、下一筆的記錄。

#### 執行結果

1. 表單有 `第一筆`、`上一筆`、`下一筆`、`最末筆` 鈕，透過這四個按鈕可動態巡覽每一筆會員記錄。
2. 如左下圖，當會員記錄在第一筆時，則 `第一筆` 及 `上一筆` 鈕無法使用；如右下圖，當會員記錄在最後一筆時，則 `下一筆`、`最末筆` 鈕無法使用。

資料表

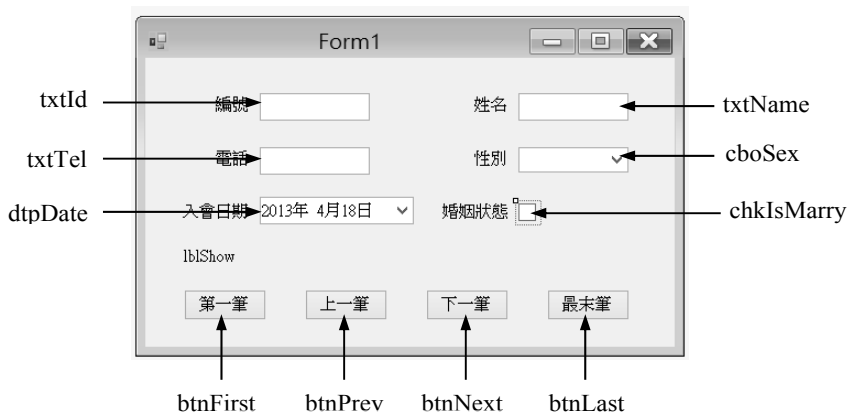
資料表名稱	會員				
主鍵值欄位	編號				
欄位名稱	資料型態	長度	允許 null	預設值	備註
編號	nvarchar	10	否		Primary Key
姓名	nvarchar	10	否		
電話	nvarchar	10	否		
性別	nvarchar	4	否		
入會日期	date		否		日期型別資料
婚姻狀態	bool		否		需輸入 true/false

操作步驟

Step 01 複製資料庫

請將書附光碟「資料庫」資料夾下的 ch18DB.mdf 和 ch18DB\_log.ldf 複製到目前製作專案的 bin\Debug 資料夾下，使 ch18DB.mdf 資料庫與範例執行檔在相同路徑下。

Step 02 建立表單輸出入介面



## Step 03 撰寫程式碼

## 程式碼 FileName:Form1.cs

```
01 using System.Data;
02 using System.Data.SqlClient;
03
04 namespace BindingManagerBase1
05 {
06     public partial class Form1 : Form
07     {
08         public Form1()
09         {
10             InitializeComponent();
11         }
12
13         // 宣告 BindingManagerBase 物件 bm
14         // 使用此物件來巡覽產品資料表的記錄
15         BindingManagerBase bm;
16
17         // 定義 CheckBm 方法，該方法用來顯示目前記錄的位置
18         // 使 第一筆、上一筆、下一筆、最末筆 鈕是否可被使用
19         private void Checkbm()
20         {
21             if (bm.Position == 0)
22             {
23                 btnFirst.Enabled = false;
24                 btnPrev.Enabled = false;
25                 btnNext.Enabled = true;
26                 btnLast.Enabled = true;
27             }
28             else if (bm.Position == bm.Count - 1)
29             {
30                 btnFirst.Enabled = true;
31                 btnPrev.Enabled = true;
32                 btnNext.Enabled = false;
33                 btnLast.Enabled = false;
34             }
35             else
36             {
```

## 18.6 習題

### 一. 選擇題

- 下列何者不是使用預存程序的優點？
  - (1) 可一次執行一系列的 SQL 敘述
  - (2) 簡化複雜的 SQL 敘述
  - (3) 執行速度快
  - (4) 採直譯方式
- 呼叫預存程序時，參數的預設傳遞方向為？
  - (1) 輸入
  - (2) 輸入與輸出
  - (3) 輸出
  - (4) 傳回值
- Command 物件欲使用預存程序，則 CommandType 屬性應設為？
  - (1) CommandType.Text
  - (2) CommandType.StoredProcedure
  - (3) CommandType.Procedure
  - (4) CommandType.SqlProcedure
- 欲傳回預存程序的傳回值，則 SqlParameterCollection 集合物件的 Direction 屬性應設為？
  - (1) ParameterDirection.Input
  - (2) ParameterDirection.InputOutput
  - (3) ParameterDirection.Output
  - (4) ParameterDirection.ReturnValue
- 欲對 DataTable 物件內的記錄資料進行排序或篩選，可透過 DataTable 建立什麼物件？
  - (1) DataReader
  - (2) DataView
  - (3) DataColumn
  - (4) DataRow
- DataView 物件的什麼方法，可用來排序記錄資料？
  - (1) SortColumn
  - (2) Sort
  - (3) OrderBy
  - (4) RowFilter
- DataView 物件的什麼方法，可用來篩選記錄資料？
  - (1) SortColumn
  - (2) Sort
  - (3) OrderBy
  - (4) RowFilter
- 已建立 DataSet 物件 ds，ds 內含會員 DataTable，現在有一個姓名下拉式清單 cboName 欲顯示會員 DataTable「姓名」欄位所有資料，則程式應如何撰寫？
  - (1) cboName.DataSource=ds; cboName.DataMember="會員";
  - (2) cboName.DataSource=ds; cboName.DisplayMember="會員";
  - (3) cboName.DataSource=ds; cboName.DataMember="會員.姓名";
  - (4) cboName.DataSource=ds; cboName.DisplayMember="會員.姓名";

9. 下列哪個物件可用來巡覽資料表的記錄？  
 (1) Binding (2) BindingContext (3) DataView (4) BindingManagerBase
10. 有一個 DataView 物件 dv，若 dv 要篩選「地址」欄位住在「忠明路」的會員，則程式應如何撰寫？  
 (1) dv.RowFilter="地址='忠明路'"; (2) dv.RowFilter="地址 Like '%忠明路'";  
 (3) dv.RowFilter="地址 Like '%忠明路%'"; (4) dv.RowFilter="地址 Like '忠明路%'";

## 二. 程式設計

1. 延續 BindingManagerBase1.sln 範例，新增可巡覽第一筆、上一筆、下一筆、最末筆的功能表項目，結果如下圖：



2. 試設計 InsertBook、UpdateBook、DeleteBook、GetBook 四個預存程序，透過上述這四個預存程序來新增、修改、刪除、預覽「書籍」資料表。
3. 延續上例，請使用 InsertBook、UpdateBook、DeleteBook、GetBook 四個預存程序來建立可新增、修改、刪除、預覽的書籍管理系統。
4. 試製作兩個關聯資料表的資料庫應用程式。將書附光碟中 [資料庫/Northwind.mdf] 的「訂貨主檔」及「訂貨明細」資料表的「訂單編號」欄位關聯，當選取上方的 DataGridView 中訂貨主檔的某一筆記錄，下方的 DataGridView 即會顯示訂貨主檔對應的所有訂貨明細記錄。