序

資訊應用系統可大可小，大的應用系統包山包海，複雜度與困難度也高；而規模小的應用系統，仍然是麻雀雖小、五臟俱全，該有的功能都要有。常見的功能和要求，例如：系統穩定性、報表、界面美觀、權限控制、多國語、資安、系統文件、簽核流程……等，這樣的完整性不管對於個人或是團隊都會是不小的挑戰；除此之外，在開發系統的過程中，有時候會遇到一些困難，像是：標準化、人員溝通、人員異動、技術門檻、文件不足、規格變動……等，這些問題可能造成的結果是：進度落後、成本增加、系統不穩定、維護困難……等。

ASP.NET Core 是微軟最新的 Web 開發平台，正式發佈以來有很好的功能和穩定性，基於舊的平台可能逐漸退出市場，以及軟體程式重複使用的特性，我們根據以往的開發經驗，選擇 ASP.NET Core 加上 jQuery、Bootstrap，實作了 20 個功能常見的軟體積木以及相關的內容，希望解決上面提到的問題，同時達到降低技術門檻與開發成本，提升系統穩定性的目的。

在本書 20 個章節中，第 1 章的資料庫文件系統是一個輔助工具；第 4 章的人事管理系統用來當做範例解說；其餘每一章主要介紹一個獨立的功能，即是我們所謂的軟體積木，透過組合這些積木可以建立一個軟體系統。「站在巨人的肩膀上」是牛頓的名言，本書希望能藉由軟體積木，加速讀者建置資訊應用系統。

碁峰 www.gotop.com.tw

如果你是程式新手，可以先了解這些軟體積木的應用，按照上面的步驟可以快速實作所需要的功能；如果你已經有豐富的開發經驗，而且也有興趣，則可以看看裡面的核心程式，對於開發大型專案會有一些幫助；關於本書的任何想法可以到臉書社團「ASP.NET Core 軟體積木」來交流，書中所有的程式碼都可以從 GitHub 下載，我們也會持續更新並且上傳。

最後，謝謝碁峰資訊的協助，讓這本書可以順利出版，更感謝你的支持購買，讓我們有機會分享這些技術，書中有不清楚的地方，歡迎寫信到底下的信箱，我們會盡可能一一回覆。

陳明山、江通儒

bruce68tw@gmail.com

線上下載

本書程式碼請至 https://github.com/bruce68tw 下載，其內容僅供合法持有本書的讀者使用，未經授權不得抄襲、轉載或任意散佈。

碁峰 www.gotop.com.tw

chapter 1

資料庫文件系統

所謂資料庫文件指的就是資料表的欄位說明，它是系統開發最基本的文件，多數軟體系統的功能是存取資料庫，一份正確的資料庫文件可以讓系統開發的工作更加順利，它的結構簡單，如圖 1-1：

資料庫文件　　管理系統

Table: Column(欄位檔)

序	欄位名稱	中文名稱	資料型態	Null	預設值	說明
1	Id	Id	varchar(10)			PKey
2	TableId	資料表 Id	varchar(10)			
3	Code	欄位代碼	varchar(30)			
4	Name	欄位名稱	nvarchar(30)			
5	DataType	資料型態	varchar(20)			
6	Nullable	可空值	bit			

圖 1-1　資料庫文件格式

許多人可能會使用 Word、Excel 來記錄這一類的內容，但隨著時間推進和系統的複雜度提高，它會越來越不容易維護。除此之外，這些欄位資料是一種有用的資源，可以拿來製作一些方便的工具，幫助系統開發。基於這些原因，我們開發了一套資料庫文件系統（以下稱 DbAdm），使用的開發

碁峰 www.gotop.com.tw

工具為 ASP.NET Core 6、Bootstrap 4、jQuery 3.4，及 Visual Studio 2022 Community。

DbAdm 在功能上屬於工具的一種，用來協助其他系統的開發工作，所以必須考慮不同語系的使用者，因此我們加入多國語的功能，包含的三種語系分別為：正體中文、簡體中文，以及英文。使用時可以在組態檔案 appsettings.json 設定所要使用的語系。在用途上，DbAdm 包含以下的功能：

- 維護資料庫文件相關的資料表。
- 從現有的 MSSQL 資料庫匯入欄位資訊。
- 產生資料庫文件 Word 檔案。
- 產生 CRUD 原始碼：所謂 CRUD 指的是新增、查詢、修改、刪除，它代表對資料庫的存取動作，是最常見的系統功能，我們將在「第 5 章 CRUD 產生器」介紹這個功能。
- 產生資料庫異動記錄：自動產生 Trigger 檔案來追蹤資料庫異動，這個功能我們在「第 17 章 資料異動記錄」介紹。

以上這些用途都會實作在 DbAdm 系統裡面，當你進入這個系統，左邊會顯示系統功能表，目前有四個項目，如圖 1-2：

圖 1-2 DbAdm 功能表

碁峰 www.gotop.com.tw

1-1 安裝

執行 DbAdm 之前，必須先進行安裝，在這裡我們使用安裝原始碼的方式，在 Visual Studio 的環境下執行，它有以下兩個步驟：

一、下載

DbAdm 包含以下兩個下載檔案，它們位於 Github 網站，任何人都可以自由下載：

- Base：裡面包含 Base、BaseWeb 兩個專案，下載的網址為 https://github.com/bruce68tw/Base。
- DbAdm：內容為 DbAdm 專案，下載的網址為 https://github.com/bruce68tw/DbAdm。

在你的本機建立一個目錄後（例如 d:\project），把 GitHub 下載的兩個壓縮檔放到此目錄，解壓縮兩個檔案後，移除目錄名稱後面的 master 文字，看到的結果如圖 1-3：

圖 1-3 DbAdm 專案目錄壓縮檔案

二、建立資料庫

進入 SQL Server Management Studio（SSMS），建立一個空白的資料庫，名稱為 Db，同時把它切換為目前的工作資料庫，在這裡我們使用的資料庫引擎為 MS SQL LocalDB 2016，你也可以使用 MS SQL Server 或是 MS

碁峰 www.gotop.com.tw

SQL Express。在 SSMS 下執行 DbAdm/_data/createDb.sql，這個檔案會建立 DbAdm 系統所有的資料表，並且產生資料內容，它包含九個資料表，如圖 1-4：

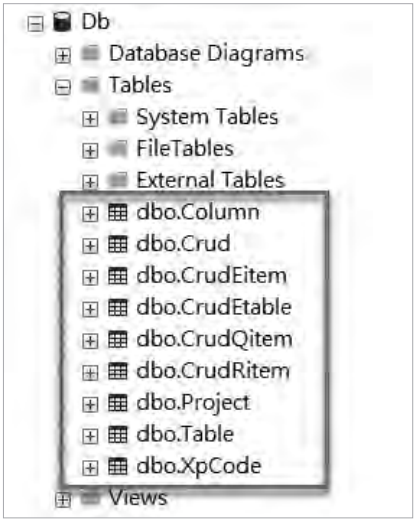

圖 1-4 Db 資料庫的資料表清單

完整的欄位內容可以參考 DbAdm/_data/Tables.docx 檔案，它同時也是利用這個系統所產生出來的，資料表內容說明如下：

- Column：欄位資料。
- Crud：CRUD 設定。
- CrudEitem：CRUD 維護資料表的欄位。
- CrudEtable：CRUD 維護資料表。
- CrudQitem：CRUD 查詢條件欄位。
- CrudRitem：CRUD 查詢結果欄位。
- Project：專案資料。
- Table：資料表。

碁峰 www.gotop.com.tw

- XpCode：雜項檔，這個資料表用來儲存 Key-Value 的對應資料，在性質上屬於系統用途，所以我們在前面加上 Xp，用來跟其他性質的資料表做區隔，依照字母排序時，這一類的資料表會顯示在最後面。

完成以上兩個步驟後，安裝的動作就結束了，進入 Visual Studio 開啟 DbAdm/DbAdm.sln 方案，你會看到四個專案（Project），如圖 1-5：

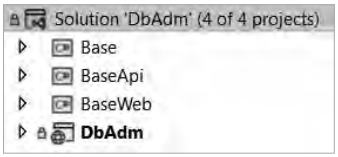

圖 1-5 DbAdm 方案的專案清單

這些專案的內容為：

- Base：底層的公用程式，任何專案都必須加入參照。
- BaseApi：Web API 專案的公用程式。
- BaseWeb：與 Web MVC 有關的公用程式，Web 專案必須加入參照。
- DbAdm：本系統的主程式。

重新編譯整個方案後，啟動 DbAdm 專案，系統預設會開啟「專案維護」作業表示安裝正常，目前我們建立了兩筆專案資料，如圖 1-6：

圖 1-6 DbAdm 專案維護作業的列表畫面

碁峰 www.gotop.com.tw

1-2 組態設定

DbAdm/appsettings.json 裡的 FunConfig 區段記錄系統執行時所需要的組態內容，如圖 1-7：

```
"FunConfig": {
  "Db": "Data Source=(localdb)\\mssqllocaldb;Initial Catalog=Db;Integrated
  "Locale": "zh-TW",
  "LogSql": "true",
  "LogDebug": "true",
  /* Smtp: 0(Host),1(Port),2(Ssl),3(Id),4(Pwd),5(FromEmail),6(FromName) */
  "Smtp": ""
}
```

圖 1-7 DbAdm appsettings.json 的 FunConfig 內容

它包含以下的欄位：

- Db：資料庫連線字串，你必須根據實際的狀況設定正確的內容，常用的屬性有 Data Source（Server）、Initial Catalog（Database）、User ID（Uid）、Password（Pwd）、Integrated Security（Trusted_Connection），括號內是另一組屬性名稱，兩組名稱可以互換。我們用這個字串內容來處理 ADO.NET 和 Entity Framework 的資料庫連線，其中的 MultipleActiveResultSets=True 是為了在一次連線中多次存取資料而設定。
- Locale：指定的多國語語系，目前允許的輸入值分別為：zh-TW（正體中文）、zh-CN（簡體中文）、en-US（英文），設定這個欄位，執行時系統即會呈現不同的語系。
- LogSql：是否記錄 SQL 的內容到 Log 檔案，預設 false，所有 Log 檔案會存放在 _log 目錄底下，這一類的檔案名稱字尾為 sql。
- LogDebug：是否記錄除錯資訊到 Log 檔案，預設 false，檔案名稱字尾為 debug。

碁峰 www.gotop.com.tw

- Smtp：寄送 email 的組態設定，字串內容包含六個欄位，中間以逗號分隔，因為這個系統無此功能，所以為空白。在「第 15 章　簡單報表」我們使用了 SMTP 功能。

1-3 Startup.cs 說明

啟動 DbAdm 時會執行 Startup.cs 的 ConfigureServices 函數（或稱類別方法），它會設定系統許多底層程式所需要的功能，程式內容如下：

```
//1.config MVC
services.AddControllersWithViews()
    //view Localization
    .AddViewLocalization(LanguageViewLocationExpanderFormat.Suffix)
    //use pascal for newtonSoft json
    .AddNewtonsoftJson(options => { options.UseMemberCasing(); })
    //use pascal for MVC json
    .AddJsonOptions(options => { options.JsonSerializerOptions.
PropertyNamingPolicy = null; });

//2.set Resources path
services.AddLocalization(options => options.ResourcesPath = "Resources");

//3.http context
services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();

//4.user info for base component
services.AddSingleton<IBaseUserService, BaseUserService>();

//5.ado.net for mssql
services.AddTransient<DbConnection, SqlConnection>();
services.AddTransient<DbCommand, SqlCommand>();

//6.appSettings "FunConfig" section -> _Fun.Config
var config = new ConfigDto();
Configuration.GetSection("FunConfig").Bind(config);
_Fun.Config = config;
```

碁峰 www.gotop.com.tw

程式解說

(1) 設定 MVC 的執行環境，這裡包含三個屬性：第一是啟用 View 檔案的多國語功能，第二是解決 Newtonsoft Json 大小寫問題，在這裡我們使用 Newtonsoft 來處理 Json 資料，當 ASP.NET Core 回傳 Json 資料到前端網頁時，系統會自動轉換成小 Camel 格式，這會造成許多不便和錯誤，加上 UseMemberCasing 這個選項可以讓資料維持原本的大小寫格式。第三是解決 MVC Json 大小寫問題，從 Controller 傳回 JsonResult 資料時會有類似的大小寫問題，必須加上這個選項。

(2) 設定多國語檔案的所在目錄為 Resources。

(3) 註冊 HttpContext：在系統內的許多地方我們需要存取 Request、Response、Cookie、Session 這些物件，它們必須透過 HttpContent 來存取，所以我們先註冊 HttpContext，然後在 BaseWeb/Services/_Http.cs 公用程式中可以很方便存取這些物件。

(4) 設定讀取登入者基本資料的服務程式為 BaseUserService 類別，系統許多核心程式會需要讀取這個基本資料。

(5) 我們使用 ADO.NET 來處理 CRUD 功能中對資料庫的存取，這兩行程式用來註冊 SqlConnect、SqlCommand 類別，它同時表示我們所要存取的是 MSSQL 資料庫。

(6) 讀取組態檔資料：把系統組態 appsettings.json 檔案中 FunConfig 區段的欄位資料儲存到 _Fun.Config 變數裡，方便系統在任何地方讀取這些組態內容。

同時在 Configure 函數中我們也做了一些調整，內容如下：

```
//1.initial & set locale
_Fun.Init(env.IsDevelopment(), app.ApplicationServices, DbTypeEnum.MSSql);

//2.set locale
_Locale.SetCultureAsync(_Fun.Config.Locale);
```

碁峰 www.gotop.com.tw

```
//3.exception handle
if (env.IsDevelopment())
{
     app.UseDeveloperExceptionPage();
}
else
{
     app.UseExceptionHandler("/Home/Error");
     ...
}
```

程式解說

(1) 呼叫 _Fun 的 Init 函數進行系統的初始化，_Fun 是最底層的公用類別，所有可執行的專案在啟動時都必須執行。

(2) 設定系統預設的語系。

(3) 使用系統預設的例外處理機制，在開發模式下會顯示詳細的錯誤內容；在正式模式下則會導到 Error.cshtml 頁面，參考「第 19 章 Log 與例外處理」。

1-4 專案目錄說明

以下是 DbAdm 專案目錄清單，其中底線開頭的目錄名稱代表作為特殊用途：

- _data：包含許多工作檔案，其中 createDb.sql 用來建立本系統的資料表以及產生資料內容；Tables.docx 是利用本系統所產生的資料庫文件。
- _log：系統運行所產生 Log 檔案，參考「第 19 章 Log 與例外處理」。
- _template：各種功能所需要的範本檔案。

碁峰 www.gotop.com.tw

- Controllers：Controller 類別檔案。
- Enums：系統所需要的 Enum（列舉）類別，如果檔名結尾是 Enum，表示是數字型態，如果是 Estr，則表示為字串型態，例如：InputTypeEstr.cs。
- Models：系統所需要 Model 類別，檔名後面若為 Dto，表示 Data Transfer Object，Vo 表示 View Object。
- Resources：多國語資料檔案，這裡用於 View 頁面。
- Services：服務類別。
- Views：網頁檔案。
- wwwroot：Web 目錄，Visual Studio 自動產生。
- Tables：使用 Database First 所產生出來的 Entity Model，方法是在 Nuget Console 下執行 Scaffold-DbContext 指令，如以下內容：

```
Scaffold-DbContext "Name=FunConfig:Db" Microsoft.EntityFrameworkCore.SqlServer
 -Project DbAdm -context MyContext -OutputDir Tables -Force -NoPluralize
```

執行成功後，會在 Tables 目錄下產生 MyContext.cs 檔案以及多個 Entity Model 類別。要注意的是，每次重新產生後都必須修改 MyContext.cs 的內容，如圖 1-8，Scaffold-DbContext 的參數內容可以參考「6-1 節 ASP.NET Core」中的 Entity Framework 段落。

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    if (!optionsBuilder.IsConfigured)
    {
        optionsBuilder.UseSqlServer(_Fun.Config.Db);
    }
}
```

圖 1-8 DbAdm MyContext.cs 手動修改內容

碁峰 www.gotop.com.tw

1-5　功能清單

啟動 DbAdm 系統後，主畫面左側會顯示功能清單，每一個功能主要包含兩個畫面，第一個是列表畫面，以分頁的方式面顯示多筆資料的查詢結果；第二個是編輯畫面，用來新增、修改或是檢視資料內容。多數的資料列表其畫面結構和操作十分類似，我們將在「第 2 章　CRUD 列表畫面」介紹它的內容和實作方法；以下是 DbAdm 的功能說明。

一、專案維護

如圖 1-9，每筆資料後面有三個連結，同時也是這個作業的主要功能：

專案維護

新增 +

專案	資料庫	功能	維護	資料狀態
DbAdm	Db	匯入結構 | 產生文件 | 產生異動Sql	✎ ✕	正常
HrAdm	Hr	匯入結構 | 產生文件 | 產生異動Sql	✎ ✕	正常

每頁顯示 10 筆, 第 1 至 2 筆, 總共 2 筆　|< < 1 > >|

圖 1-9　專案維護作業

連結功能說明如下：

- 匯入結構：把來源資料庫的欄位資訊匯入本系統，如果某個欄位資料已經存在，則系統會做更新處理。
- 產生文件：產生並且下載資料庫文件 Word 檔案。
- 產生異動 SQL：產生並且下載資料庫異動紀錄所需要的 Trigger 檔案，這部分更多的內容請參考「第 17 章　資料異動記錄」。

碁峰 www.gotop.com.tw

除了上述的功能，這個作業用來維護 Project 資料表，它的編輯畫面如圖 1-10：

圖 1-10 專案維護作業的編輯畫面

欄位說明：

- 專案代碼：專案唯一代碼，不可重複，它同時也是產生 CRUD 檔案時類別的命名空間（Namespace）。
- 資料庫：資料庫的實際名稱。
- 資料狀態：資料狀態是否為啟用。
- 專案路徑：產生的 CRUD 檔案所存放的目錄。
- DB 連線字串：資料庫連線字串。

二、資料表維護

如圖 1-11，其中的「新增」和「匯出」按鈕是 CRUD 的基本功能；「產生文件」可以讓你針對選取的多筆資料表產生 Word 文件，文件格式與「專案維護」作業相同。

碁峰 www.gotop.com.tw

圖 1-11 資料表維護作業

另外，這個作業用來維護 Table 和 Column 兩個資料表，如圖 1-12，其中畫面上方是單筆的 Table，畫面下方是多筆 Column，兩個資料表之間存在一對多的關聯：

圖 1-12 資料表維護功能編輯畫面

碁峰 www.gotop.com.tw

畫面上方的欄位說明：

- 專案：所屬專案。
- 資料表：資料庫裡面的資料表名稱。
- 資料表名稱：資料表顯示名稱。
- 異動記錄：是否產生異動記錄 Trigger 檔案，在「專案維護」執行「產生異動 SQL」時，系統只會讀取這個欄位有勾選的資料表。
- 資料狀態：如果狀態不為啟用，則任何輸出將不會包含此筆資料。

畫面下方的欄位說明：

- 欄位：資料庫裡面的欄位名稱。
- 欄位名稱：欄位顯示名稱，產生 CRUD 檔案時，系統會使用這個資料做為欄位的標題（Label），所以在設定內容時，必須要多考慮畫面上呈現出來的意義。
- 資料型態：匯入時產生。
- 可空值：匯入時產生。
- 預設值：匯入時產生。
- 排序：匯入時產生，產生資料庫文件時，欄位會用這個資料來排序。
- 說明：欄位說明。
- 資料狀態：如果狀態不為啟用，則任何輸出將不會包含此筆資料。

碁峰 www.gotop.com.tw

三、欄位維護

這個功能相對簡單，用來維護 Column 單一資料表，如圖 1-13：

圖 1-13 欄位維護作業的編輯畫面

除了以上三個作業的功能，如果你還想為現有的 MS SQL 資料庫產生文件，可以遵循以下步驟：

1. 在「專案維護」作業新增一筆資料，同時輸入正確的欄位資料，然後進入列表畫面執行「匯入結構」功能，系統會把資料寫入 Table 和 Column 這兩個資料表。
2. 分別進入「資料表維護」和「欄位維護」作業，填寫必要的欄位。
3. 回到「專案維護」作業，執行「產生文件」功能，系統即會產生並且下載這個 Word 檔案。

碁峰 www.gotop.com.tw

四、CRUD 維護

如圖 1-14，這個作業的主要目的是自動產生 CRUD 原始碼，減少程式員 Coding 的成本，點擊畫面上的「產生 CRUD」按鈕，系統即會為所選取的多筆資料產生對應的 CRUD 檔案，每個 CRUD 功能會產生六個 MVC 檔案，我們將在「第 5 章　CRUD 產生器」介紹這個功能；除此之外，這個作業會維護五個 CRUD 相關的資料表。

圖 1-14　CRUD 維護作業

1-6　程式解說

針對 DbAdm 系統中比較複雜的功能，在此解釋它的程式邏輯，為避免說明過於瑣碎，我們在程式碼前後加入 #region，以區塊的方式講解它們的內容：

碁峰 www.gotop.com.tw

一、匯入資料庫結構

使用者在「專案維護」作業執行「匯入結構」功能時，系統最後會呼叫 DbAdm/Services/ImportDbService.cs 的 Run 函數，它是這個功能的主要程式，用途是把來源資料庫的欄位資訊寫入 Table 和 Column 資料表，程式結構如圖 1-15：

```
1 reference
public async Task<ResultDto> RunAsync(string projectId)
{
    var result = new ResultDto();

    1.get dbo.Project row

    2.create temp table: #tmpTable, #tmpColumn

    3.write #tmpTable(from Information_Schema.Tables)

    4.write #tmpColumn(from Information_Schema.Columns)

    5.insert/update dbo.Table from #tmpTable

    6.insert/update dbo.Column from #tmpColumn

lab_exit:
    if (dbSrc != null)
        await dbSrc.DisposeAsync();

    await db.DisposeAsync();
    return result;
}
```

圖 1-15　匯入資料庫結構功能的程式結構

碁峰 www.gotop.com.tw

程式解說

(1) 函數為非同步，利用傳入專案 Id 讀取對應的 Project 資料表。

(2) 建立暫存資料表：每次程式啟動時我們會建立兩個暫存資料表 #tmpTable、#tmpColumn，用來儲存從系統資料庫讀取出來的資料表和欄位資訊，在程式執行完畢時，這些暫存資料表會自動刪除。

(3) 寫入資料表 #tmpTable：系統所有的資料表以及欄位可以透過 Information.Tables 和 Information.Columns 這兩個系統檢視表來讀取，同時傳入 Project.Id。

(4) 寫入資料表 #tmpColumn。

(5) 從 #tmpTable 寫入 Table 資料表，如果這是一個新的資料表，則系統會寫入一筆記錄，如果這個資料表已經不存在資料庫了，那麼系統會設定這一筆 Table.Status = 0，表示此筆資料已經停用。

(6) 從 #tmpColumn 寫入 Column 資料表。

二、產生資料庫文件

使用者執行「專案維護」或「資料表維護」的「產生文件」功能時，系統執行的主要程式為 DbAdm/Services/GenDocuService.cs 的 Run 函數，傳入 Project.Id 或是多個 Table.Id，表示資料來源是某個專案或是多個資料表，程式結構如圖 1-16：

碁峰 www.gotop.com.tw

```
public async Task<bool> RunAsync(string projectId, string[] tableIds = null)
{
    1.check input & template file

    2.read column rows & group by

    3.get memory stream for download file

    //binding stream && docx
    using (var docx = WordprocessingDocument.Open(ms, true))
    {
        4.get body/row template string

        //table list loop
        var bodyLeft = bodyTpl.TplStr.Substring(0, rowTpl.StartPos);
        var bodyRight = bodyTpl.TplStr.Substring(rowTpl.EndPos);
        var fileStr = "";   //file string to echo
        for (var i = 0; i < tableLen; i++)
        {
            5.get table string

            6.add page break if need
        }

        7.get file string

        remark testing code
    }

    //8.download file
    await _Web.ExportByStream(ms, "Tables.docx");
    return true;

lab_error:
    await _Log.ErrorAsync("GenDocuService.cs RunAsync() failed: " + error);
    return false;
}
```

圖 1-16 產生資料庫文件功能的程式結構

碁峰 www.gotop.com.tw

程式解說

(1) 檢查傳入參數和範本檔案是否存在。

(2) 利用傳入參數來讀取 Column 資料表並且分群排序。

(3) 使用 Memory Steam 作為下載檔案的資料來源，這個功能將不會在主機產生任何實體檔案。

(4) 讀取範本檔的內容。

(5) 從範本檔和 Column 資料表產生單一資料表文件的字串內容。

(6) 在每個資料表文件後面加上分頁符號。

(7) 產生完整的文件檔案字串。

(8) 下載文件檔案。

碁峰 www.gotop.com.tw