

【發明摘要】

【中文發明名稱】 用於有效率存取資料之軟體工具及其方法

【中文】我們使用的方法就是現今很多研究者使用這種object stream的技術來儲存資料；而且我們認為object stream是一個有效率檔案建構方法，其原因如下三點：1、由於Java的object-oriented circles有persistence的特性，物件儲存在檔案的佈局(layout of a file)會完全跟在記憶體佈局的佈局一樣。2、所有相關的物件皆以陣列方式管理。3、只要我們能提供所要擷取資料的路徑，不論資料存在檔案內何處，這些資料都可以迅速擷取。

【指定代表圖】 第一圖

【發明說明書】

【中文發明名稱】 用於有效率存取資料之軟體工具及其方法

【技術領域】

【0001】 本發明係有關於物件及陣列儲存的架構，特別是指一種應用資料存取之存取路徑。

【先前技術】

【0002】 習知資料管理，如使用object stream的技術來儲存資料；而且object stream是一個有效率檔案建構方法，如下：1、由於Java的object-oriented circles有persistence的特性，物件儲存在檔案的佈局(layout of a file)會完全跟在記憶體中的佈局一樣。2、所有相關的物件皆以陣列方式管理。3、只要我們能提供所要擷取資料的路徑，不論資料存在檔案內何處，這些資料都可以迅速擷取。

【發明內容】

【0003】 爰此，本發明提出一種用於有效率存取資料之方法，該方法係為：

【0004】 (a)透過一物件儲存一相關資訊；
(b)上述物件經由一導向引擎儲存於相對應之一陣列；

- (c)再將上述儲存該物件的該陣列儲存於相對應之一類別；
- (d)藉由上述步驟取得該物件之相關取得儲存之一存取路徑；
- (e)經由上述存取路徑搜尋並取得該物件。

【0005】 進一步增加一步驟(f)上述物件係儲存於上述陣列中，且上述陣列在儲放於另一物件中，係以層層堆疊儲放之方式儲存上述物件以及陣列。

【0006】 進一步，上述導向引擎係為資料導向引擎以及陣列導向引擎。

【0007】 進一步，上述資料導向引擎係包含存取。

【0008】 進一步，上述陣列導向引擎包含：插入、修改、刪除及存取。

【0009】 本發明再提供一種用於有效率存取資料之軟體工具ISG(資訊系統產生器Information System generator簡稱ISG)之檔案建立以方便存取檔案內之資料，此檔案建立軟體工具本發明稱為An Efficient Building File System Generator簡稱BFS-G，BFS-G至少包含：一類別之建立至少一物件以及一儲存上述物件之陣列，構成一存取路徑，該物件儲存一相關資訊；上述物件係包含有至少一導向引擎；前述導向引擎係根據儲存上述相關資訊的存取路徑提供一回應，該回應係包含上述物件。

【0010】 進一步，上述軟體工具之機制係包含：呼叫

array_name_add(object_name element) : void、呼叫

array_name_expandCapacity() : void、呼叫remove_array_name() : void、

呼叫remove_array_name(int idx) : void、呼叫get_array_name_size() :

int、呼叫get_array_name (int idx) : object_name 、呼叫

array_name_isEmpty () : boolean、呼叫get_variable_name():data-type

以及呼叫set_variable_name(data-type element):void。

【0011】 進一步， BFS-G係為ISG軟體開發工具之一功能稱為檔案建立，ISG包括檔案建立、視窗連結、資料處理畫面、資料顯示、預覽和列印等6項功能，上述ISG之BFS-G係輸入一類別名稱及其屬性，進而產生一Java程式。

【0012】 進一步，上述回應之路徑之轉譯程式如下：假設路徑是class_name/array1/array2/data_name，路徑之array1和array2是陣列，透過陣列 array1之所有物件參考拿出這些物件，再由這些物件內之陣列array2，使用此array2所存放之物件，這些物件就可以找到所有data_name資料內容。設定轉譯擷取data_name公式如下所示：

```
class_name the_first_object;
```

```
/* the_first_object表示從檔案讀出時，使用the_first_object暫存此物
```

```
件，此物件之類別名稱可從第一路徑取得，即是class_name。 */
```

```

data_type_of_data_name [] arrays = new
data_type_of_data_name[1000];

/* data_type_of_data_name表示data_name之資
料型態，此data_type_of_data_name可從
BFS-G得到，因為使用BFS-G檔案建立時已
有設定好data_name之資料型態為
data_type_of_data_name，所以BFS-G那裡可
以得到data_name之資料型態，本發明使用
陣列arrays暫存data_name之資料。 */

int point; // point作為陣列arrays之索引
point=0;
the_first_object = (class_name) ois.readObject();

/* 此物件the_first_object存有一陣列array1，
getarray1_size()表示陣列array1之物件個
數，下列表示依序從array1擷取物件*/

for(int i=0; i < the_first_object.getarray1_size(); i++){

/* 陣列array1存放類別class_of_object1之物
件，class_of_object1可從BFS-G得知，因使
用BFS-G時已設定好此陣列array1所儲存之
物件資料型態為class_of_object1。*/

class_of_object1 tmp1;

//tmp1暫存陣列array1所取出之物件參考

tmp1 = the_first_object.get_array1_class_of_object1(i);

```

```

//先擷取第一個物件從陣列array1索引i=0開始
for(int j=0; j < tmp1.getarray2_size(); j++) {
    class_of_object2 tmp2;
        // tmp2暫存陣列array2所取出之物件參考
    tmp2 = tmp1.get_array2_class_of_object2(j);
        //先擷取第一個物件從陣列array2索引j=0開始
    arrays[pointer++] =tmp2.getdata_name();
        //從物件tmp2取出data_name資料，依序暫存於陣
列arrays
    }
}
。

```

【0013】 進一步增加一Join Approach機制，Join Approach機制係藉由一下拉式選單的一登錄畫面，將多個檔案Join在一個檔案中。

【0014】 進一步，上述BFS-G增加一擴充機制。

【0015】 本發明的功效在於：

【0016】 1.本發明之該類別單元係為一多層式架構，藉由陣列和導向引擎機制，再藉用一路徑來轉譯程式碼，所轉譯之程式碼可加速資料存取之時間。

【0017】 2.本發明的擴充機制，用以防止該資料儲存空間在儲存過多的該物件後，無法再儲存於該陣列，故本發明具有可彈性擴充該資料儲存空間之功效。

【0018】 3.本發明的Join Approach機制，Join Approach機制係藉由一下拉式選單的一登錄畫面，將多個檔案Join在一個檔案中。

【圖式簡單說明】

【0019】 [第一圖]係為本發明之系統檔案架構之標準模式類別圖。

【0020】 [第二A圖]係為本發明之某一公司檔案架構圖，以合約/出貨單檔案為例。

【0021】 [第二B圖]係為本發明之某一公司檔案架構圖，以每批出貨開銷檔案為例。

【0022】 [第二C圖]係為本發明之某一公司檔案架構圖，以出貨開銷費用種類檔案為例。

【0023】 [第二D圖]係為本發明之某一公司檔案架構圖，以台灣地址檔案為例。

【0024】 [第三圖]係為本發明之流程圖。

【0025】 [第四圖]係為本發明之BFS-G檔案建立工具，使用此工具設定所有檔案之類別名稱，係以某一公司營運資訊系統17檔案類別名稱。

【0026】 [第五圖]係為合約/出貨單檔案之類別invoice_array為例，本發明之BFS-G為類別invoice_array設定其資料成員，按Translate to an object java code按鈕，則產生invoice_array.java程式碼，此程式碼與第二圖invoice_array類別圖所示相同。

【0027】 [第六圖]係為係為合約/出貨單檔案之類別proforma_invoice_data為例，本發明之BFS-G為類別proforma_invoice_data設定其資料成員，按Translate to a java code按鈕，則產生proforma_invoice_data.java程式碼，此程式碼與第二圖proforma_invoice_data類別圖所示相同。

【0028】 [第七圖]係為合約/出貨單檔案之類別pi_data為例，本發明之BFS-G為類別pi_data設定其資料成員，按Translate to a java code按鈕，則產生pi_data.java程式碼，此程式碼與第二圖pi_data類別圖所示相同。

【0029】 [第八圖]係為每批出貨開銷檔案之類別goods_payout_data為例，本發明之BFS-G為類別goods_payout_data設定其資料成員，按Translate to an array code按鈕，則產生

goods_payout_array.java和goods_payout_data.java程式碼，這兩套程式碼與第二圖goods_payout_array類別圖和goods_payout_data類別圖所示相同。

【0030】 [第九圖]係以登錄每批出貨開銷為例，描述如何Join每批出貨開銷檔案和出貨開銷費用種類檔案。

【實施方式】

【0031】 綜合上述技術特徵，本發明用於有效率存取資料之軟體工具及其方法的主要構造及技術特徵，可在下述實施例清楚呈現。

【0032】 本發明之結構與實施例請參閱第一圖至第九圖所示，本發明之方法係包括：**(a)**透過一物件儲存一相關資訊；**(b)**上述物件經由一導向引擎add() 儲存於相對應之一陣列；**(c)**上述儲存該物件的該陣列，此陣列是存在一類別之物件內，此物件可從檔案讀出；**(d)**藉由上述步驟取得該物件之相關儲存之一存取路徑，此路徑是從檔案讀出一物件開始，第一路徑是此物件之類別名稱，第二路徑是此物件之一陣列，則可存取陣列之所有物件之資料；**(e)**經由上述存取路徑搜尋並取得該物件。

【0033】 特別說明：上述物件以及陣列係以層層堆疊儲放之方式儲存上述物件以及陣列，例如至少一個上述物件儲存於上述陣列中，上述陣列再儲存於另外一個物件中，而上述另外之物件係再儲存於另外一個陣列之中，以上述方式層層堆疊儲放上述物件以及陣列；

且上述導向引擎係為資料導向引擎以及陣列導向引擎，其中上述資料導向引擎係包含存取的功能，而上述陣列導向引擎包含：插入、修改、刪除及存取之功能，至於多層類別圖資料擷取之路徑是 `class_name/array1/array2/.../arrayn/data_name`。

【0034】 進一步， BFS-G之檔案建立功能係包括：於一類別建立至少一物件以及一儲存上述物件之陣列，構成一存取路徑，該物件儲存一相關資訊；上述物件係包含導向引擎。

【0035】 前述導向引擎係根據儲存上述相關資訊的存取路徑提供一回應，該回應係包含上述物件。

【0036】 進一步，上述軟體工具係為BFS-G，BFS-G係輸入一類別名稱及其屬性，進而產生一Java程式。

【0037】 進一步， BFS-G轉譯之導向引擎係包含：呼叫 `array_name_add(object_name element) : void`、呼叫 `array_name_expandCapacity() : void`、呼叫 `remove_array_name() : void`、呼叫 `remove_array_name(int idx) : void`、呼叫 `get_array_name_size() : int`、呼叫 `get_array_name (int idx) : object_name`、呼叫 `array_name_isEmpty () : boolean`、呼叫 `get_variable_name():data-type` 以及呼叫 `set_variable_name(data-type element):void`。

【0038】 進一步， BFS-G之擴充機制是呼叫上述 `array_name_expandCapacity() : void`。

【0039】 綜合上述技術特徵，續請參閱第一圖檔案架構之標準模式類別圖，圖中的class_name、array_name、variable_name、class_name_of_object和data_type表示使用BFS-G轉譯程式時的參數名稱各有其意義而其意思與英文所表示相同，class_name表示一類別的名稱，array_name表示一陣列名稱，variable_name表示一屬性名稱，class_name_of_object表示儲存在該陣列裡物件類別名稱，data_type表示某一變數的資料型態和element表示任何變數名稱。

【0040】 第一圖所表示的檔案可能只有一個類別圖或多層類別圖，雖然類別圖內宣告的屬性和陣列都表示 *，但在宣告屬性的部分至少要有一個宣告。

【0041】 而上述導向引擎係提供資料導向引擎和陣列導向引擎，資料導向引擎是用來處理一般變數variable_name，其資料型態為data_type，譬如取出變數variable_name資料值：(1)呼叫getvariable_name(): data_type；會傳回此屬性variable_name的資料內容。

【0042】 如設定variable_name資料值：(2)呼叫setvariable_name(data_type element): void；將variable_name的資料值element設定在此資料屬性variable_name裡。

【0043】 下列為BFS-G轉譯getvariable_name(): data_type；和setvariable_name(data_type element): void；程式碼之語法；

```
public String getvariable_name(){
```

```

        return this.variable_name; //回傳variable_name之值
    }

    public void setvariable_name(data_type element) {
        this.variable_name = element; //將element值設定於variable_name
    }
}

```

。

【0044】 特別說明：第一圖中的*表示零至多個。

【0045】 本發明再以該陣列導向引擎來說明下列七種機制。

【0046】 該陣列導向引擎運作需要下列屬性之輔助，

array_name_count表示該陣列中該物件的個數、陣列的名稱為

array_name和陣列儲存物件的類別名稱稱為class_name_of_object。

【0047】 第一個為呼叫 array_name_add(class_name_of_object element) : void;，此呼叫表示該物件會依序存放於該陣列內，而該陣列的名稱是array_name，且將存入之物件參考是element，第一筆物件參考會放在陣列的索引0之位置，然後第二筆物件參考放在索引1之位置，以此類推下去，若上述陣列放滿物件時會呼叫

array_name_expandCapacity()讓儲存的容量擴充為原來該陣列的兩倍；BFS-G轉譯array_name_add(class_name_of_object element)程式碼之語法如下所述：

```

public void array_name_add(class_name_of_object element)
{
    //檢查該物件的個數是否超過該陣列容量

```

```

If (array_name_count == array_name.length)
    array_name_expandCapacity();
//如果是，則擴充該陣列的容量為原本容量的兩倍
array_name[array_name_count] = element;
//將物件參考加入array_name_count索引所示之陣列位置
array_name_count++;
}

```

。

【0048】 第二個為呼叫array_name_expandCapacity(): void; 將該陣列array_name的容量擴充為兩倍; BFS-G轉譯

array_name_expandCapacity ()程式碼之語法如下所述:

```

public void array_name_expandCapacity() {
    class_name_of_object [] larger =
        class_name_of_object [array_name.length*2];
        /* 產生一個新陣列larger和size為陣列
           array_name容量的兩倍。 */
    for(int i=0; i < array_name.length ; i++)
        larger[i] = array_name[i];
        /* 上述將該陣列array_name所有之該
           物件參考複製到新陣列larger。 */
    array_name = larger;
    /* 將新陣列larger為array_name，因此array_name擴充為兩倍大。 */
}

```

。

【0049】 第三個為呼叫remove_array_name(): void; 刪除該陣列array_name內的所有物件; BFS-G轉譯remove_array_name ()程式碼之語法如下所述:

```
public void remove_array_name(){
    for(int ii=0; ii < array_name_count; ii++)
        array_name[ii] = null; // remove每一陣列內之該物件
    array_name_count=0;
}
```

。

【0050】 第四個為呼叫remove_array_name(int idx): void; 會刪除該陣列array_name之某物件; BFS-G轉譯remove_array_name(int idx):void; 程式碼之語法如下所述:

```
public void remove_array_name (int idx) {
    int j;
    array_name[idx] = null; //該陣列索引idx指向之物件設為null
    for (j=idx+1 ;j < array_name_count ;j++){
        array_name[j-1] = array_name[j];
        /* 將該陣列內之每一物件參考
           考往上移一位 (從刪除物件
           之下一物件) */
    }
}
```

// 則該物件已刪除，物件個數也減一

```
array_name_count--;  
}
```

。

【0051】 第五個為呼叫get_array_name_size():int;回傳該陣列array_name的該物件個數;BFS-G轉譯get_array_name_size():int;程式碼之語法如下所述:

```
public int get_array_name_size() {  
    return array_name_count;  
    //回傳該陣列array_name之該物件的個數array_name_count  
}
```

。

【0052】 第六個為呼叫get_array_name_class_name_of_object(int idx):class_name_of_object;從該陣列擷取一個該物件;BFS-G轉譯get_array_name_class_name_of_object(int idx):class_name_of_object;程式碼之語法如下所述:

```
public class_name_of_object  
    get_array_name_class_name_of_object(int idx){  
    return array_name[idx];  
    /* 回傳索引idx所指向之該類別  
    class_name_of_object之該物件。*/  
}
```

。

【0053】 第七個為 呼叫array_name_isEmpty () : boolean ; 會回傳true或false , 如果該陣列是空的沒有存放任何該物件則會回傳true , 否則回傳false 。 BFS-G轉譯 array_name_isEmpty () : boolean ; 程式碼之語法如下所述 :

```
public boolean array_name_isEmpty() {  
    return array_name_count==0;  
        //如果該陣列是空的則會回傳true  
}
```

【0054】 續請參閱第二A圖、第二B圖、第二C圖以及第二D圖 , 係以某公司營運資訊系統之檔案系統中之四個檔案為例 , 說明本發明之檔案系統架構和其構思 , 此四個檔案分別為 : 合約/出貨單檔案、每批出貨開銷檔案、出貨開銷費用種類以及台灣地址檔案。

【0055】 本發明之合約/出貨單檔案係儲存商業合約和出貨通知單 , 其包括三層類別 ; 類別invoice_array、類別proforma_invoice_data和類別pi_data。

【0056】 該類別invoice_array的該物件包括該陣列pi_proforma_invoice和該陣列si_shipping_invoice。

【0057】 該陣列pi_proforma_invoice存放商業合約和該陣列si_shipping_invoice存放出貨通知單。

【0058】 由於商業合約和出貨通知單皆存放相同類型之資料，本發明將相同類型之資料存放在類別proforma_invoice_data的物件包括公司名稱、客戶名稱、pi編號、報價條件、預定出貨日…等；因為該類別proforma_invoice_data之該物件之產品編號、產品敘述、數量、單位、單價…等有多筆相同類型之資料如同類別proforma_invoice_data之物件特性相同，因之將存放在類別pi_data之物件，再將該物件存放於與其相關之類別proforma_invoice_data的該物件之pi_array陣列。

【0059】 本發明之檔案類別圖的屬性宣告語法為
variable_name : data_type ; 以上述語法宣告pi編號、公司名稱、客戶名稱、報價條件、預定出貨日，如下所示pi_number : int; 、
pi_company : String、pi_customer : String、 pi_term : String和pi_date : date，因為pi編號、公司名稱、客戶名稱、報價條件和預定出貨日之variable_name，已先設計為pi_number、pi_company、pi_customer、pi_term和pi_date和其data_type已設計為整數、字串、字串、字串和日期之資料型態(請參閱第二A圖)。

【0060】 本發明檔案類別圖之方法及步驟請參閱第三圖所示，透過該物件儲存相關資訊，如：屬性等，上述物件經由上述資料導向引擎將一資料儲存於所在之屬性或陣列導向引擎將物件儲存於相關之該陣列中，再經由上述儲存該物件的陣列儲存到相關的該類別

中，並產生一存取路徑，可藉由上述存取路徑得知該物件或該陣列儲存的位址。

【0061】 本發明之第四圖至第八圖為BFS-G所提供之介面，藉由上述介面輸入參數後，即可透過BFS-G將參數轉譯屬性之宣告碼和導向引擎碼之Java程式。

【0062】 請參閱第四圖所示，BFS-G提供設定檔案內之所有類別名稱，如：Class Name的欄位，如goods_payout_data、goods_payout_kind_data、invoice_array、proforma_invoice_data、pi_data以及taiwan_address_data；而類別goods_payout_data、類別goods_payout_kind_data以及類別taiwan_address_data所產生物件皆各自需要一陣列來管理，所以BFS-G的檔案建立會自動產生一陣列程式goods_payout_array.java、goods_payout_kind_array.java和taiwan_address_array.java，譬如使用BFS-G之第八圖介面，登錄類別goods_payout_data之屬性後按Translate to an array code按鈕，即產生兩支Java程式碼；goods_payout_array.java和goods_payout_data.java。

【0063】 第四圖下版面Date type of the Array的欄位係表示陣列存放之物件之類別名，例如第四圖的上版面之Class Name的欄位：類別invoice_array的物件有兩個陣列皆儲存類別proforma_invoice_data的物件參考，所以在下版面欄位Date type of the

Array登錄類別profroma_invoice_data；而第四圖下方之showing additional attributes of record 3 in the buttom panel係表示滑鼠才剛點選上版面之Class Name的欄位第三筆之invoice_array，也表示Date type of the Array的欄位顯示的即是類別invoice_array之物件之陣列存放之物件之類別名稱profroma_invoice_data。

【0064】 如第五圖所示係為設定上述第四圖的一類別，上版面之Class name選擇類別invoice_array，下列之設定乃依據第二A圖之類別圖類別invoice_array之設計而設定，Field name 設定為pi_profroma_invoice和si_shipping_invoice，因兩者皆為陣列，Data Type設定為Array，其The # of dimension表示陣列之維數，因在第二A圖設計為pi_profroma_invoice: profroma_invoice_data[]; []表示陣列pi_profroma_invoice一維陣列，如[][]表示二維，因之此設定為1維，size設定100，Array data type表示陣列存放之物件之類別名，兩者皆存放類別profroma_invoice_data之物件，Array data type欄位係設定為profroma_invoice_data，按Translate an object java code按鈕，即產生Java程式碼；invoice_array.java。

【0065】 如第六圖所示，係為設定上述第四圖的一類別profroma_invoice_data，Field name係設定類別所有屬性，此類別有一陣列pi_array和一般屬性pi_number、pi_company、pi_customer、

pi_date、pi_expect_date、pi_term、pi_payout、pi_bank、pi_total以及si_total，陣列pi_array之設定參閱【0064】，其他一般屬性之設定，只需設定Field name 和Data type，這些資料可從第二A圖之類別proforma_invoice_data得知，按Translate to a java code按鈕，即產生Java程式碼；proforma_invoice_data.java。

【0066】 如第七圖所示，係為設定上述第四圖的一類別pi_data，Field name係設定pi_gu_number、pi_gu_statement、pi_amount、pi_unit、pi_price以及pi_presently...等，其Data type可從第二A圖之類別pi_data得知，按Translate to a java code按鈕，即產生Java程式碼；pi_data.java。

【0067】 如第八圖所示，係為設定上述第四圖的一類別goods_payout_data，Field name係設定gp_si_number、gp_gpk_kinds、以及gp_sum...等，其Data type可從第二B圖之類別goods_payout_data得知，按Translate to an array code按鈕，即產生兩支Java程式碼；goods_payout_array.java和goods_payout_data.java。

【0068】 第九圖係表示上述本發明之Join Approach機制，Join Approach機制係藉由登錄畫面之下拉式選單，如下拉選單的來源由一檔案得來且也儲存選單之索引，則可將多個檔案Join在一個檔案中，第九圖描述如何Join每批出貨開銷檔案和出貨開銷費用種類檔案，

因為欄位開銷種類係一下拉選單，選單資料由出貨開銷費用種類檔案之屬性開銷種類所提供，此登錄之資料儲存在每批出貨開銷檔案之類別goods_payout_data之物件，此物件有一屬性gp_gpk_kinds_index乃儲存下拉式選單之索引，此公司資訊系統可藉此gp_gpk_kinds_index索引至出貨開銷費用種類檔案之陣列g_oods_payout_kind_array拿出類別goods_payout_kind_data之物件所有資料。

【0069】 綜合上述實施例之說明，當可充分瞭解本發明之操作、使用及本發明產生之功效，惟以上所述實施例僅係為本發明之較佳實施例，當不能以此限定本發明實施之範圍，即依本發明申請專利範圍及創作說明內容所作簡單的等效變化與修飾，皆屬本發明涵蓋之範圍內。

【符號說明】

【0070】

無

【發明申請專利範圍】

【第1項】 一種用於有效率存取資料之方法，該方法係為：

- (a)透過一物件儲存一相關資訊；
- (b)上述物件經由一導向引擎儲存於相對應之一陣列；
- (c)再將上述儲存該物件的該陣列儲存於相對應之一類別；
- (d)藉由上述步驟取得該物件之相關儲存之一存取路徑；
- (e)經由上述存取路徑搜尋並取得該物件。

【第2項】 如申請專利範圍第1項所述之用於有效率存取資料之方法，進一步增加一步驟(f)上述物件係儲存於上述陣列中，且上述陣列在儲放於另一物件中，係以層層堆疊儲放之方式儲存上述物件以及陣列。

【第3項】 如申請專利範圍第1項所述之用於有效率存取資料之方法，上述導向引擎係為資料導向引擎以及陣列導向引擎。

【第4項】 如申請專利範圍第3項所述之用於有效率存取資料之方法，上述資料導向引擎係包含存取。

【第5項】 如申請專利範圍第3項所述之用於有效率存取資料之方法，上述陣列導向引擎包含：插入、修改、刪除及存取。

【第6項】 一種用於有效率存取資料之軟體工具，該軟體工具至少包含：

於一類別建立至少一物件以及一儲存上述物件之陣列，構成一存取路徑，該物件儲存一相關資訊；

上述物件係包含有至少一導向引擎；

前述導向引擎係根據儲存上述相關資訊的存取路徑提供一回應，該回應係包含上述物件。

【第7項】 如申請專利範圍第6項所述之用於有效率存取資料之軟體工具，上述軟體工具之機制係包含：呼叫array_name_add(object_name element)：void、呼叫array_name_expandCapacity()：void、呼叫remove_array_name()：void、呼叫remove_array_name(int idx)：void、呼叫get_array_name_size()：int、呼叫get_array_name (int idx)：object_name、呼叫array_name_isEmpty ()：boolean、呼叫get_variable_name():data-type以及呼叫set_variable_name(data-type element):void。

【第8項】 如申請專利範圍第6項所述之用於有效率存取資料之軟體工具，該軟體工具係為ISG軟體開發工具之檔案建立功能，本發明稱為BFS-G (Building file system generator)，上述BFS-G軟體開發工具係輸入一類別名稱及其屬性，進而產生一Java程式。

【第9項】 如申請專利範圍第6項所述之用於有效率存取資料之軟體工具，上述回應之路徑之轉譯程式如下：假設路徑是class name/array1/array2/data_name設定轉譯擷取data_name公式。

【第10項】 如申請專利範圍第6項所述之用於有效率存取資料之軟體工具，進一步增加一Join Approach機制，上述Join Approach機制係藉由一下拉式選單的一登錄畫面，將多個檔案Join在一個檔案中。