

Net.Data



# 參考手冊



Net.Data



# 參考手冊

**注意事項**

使用這個資訊與其支援的產品之前，請確實閱讀第299頁的『附錄D. 注意事項』中的資訊。

# 目錄

前言	ix
關於 Net.Data	ix
關於本書	ix
誰應閱讀本書	x
關於本書中的範例	x
如何閱讀語法圖	x
<b>第1章 Net.Data 巨集語言結構</b>	<b>1</b>
Net.Data 巨集語法	1
常見語法元素	4
變數名稱	4
變數參照	4
字串	5
巨集語言結構	5
備註區塊	7
DEFINE 區塊或陳述式	9
ENVVAR 陳述式	13
EXEC 區塊或陳述式	14
FUNCTION 區塊	16
函數呼叫 (@)	23
HTML 區塊	26
IF 區塊	28
INCLUDE 陳述式	34
INCLUDE_URL 陳述式	36
LIST 陳述式	38
MACRO_FUNCTION 區塊	40
MESSAGE 區塊	44
REPORT 區塊	48
ROW 區塊	51
TABLE 陳述式	53
WHILE 區塊	55
<b>第2章 變數</b>	<b>59</b>
使用者定義的變數	60
條件式變數	60
環境變數	61
執行變數	61
隱藏變數	62
列示變數	63
表格變數	64
Net.Data 表格處理程序變數	65
Nn	66
NLIST	67
NUM_COLUMNS	68
NUM_ROWS	69
ROW_NUM	70
TOTAL_ROWS	71
V_columnName	72
VLIST	73

Vn . . . . .	74
Net.Data 報告變數 . . . . .	75
ALIGN . . . . .	76
DTW_DEFAULT_REPORT . . . . .	77
DTW_HTML_TABLE . . . . .	78
RPT_MAX_ROWS . . . . .	79
START_ROW_NUM . . . . .	80
Net.Data 語言環境變數 . . . . .	83
DATABASE . . . . .	84
DB_CASE . . . . .	85
DB2PLAN . . . . .	86
DB2SSID . . . . .	87
DTW_APPLET_ALTTEXT . . . . .	88
DTW_EDIT_CODES . . . . .	89
DTW_PAD_PGM_PARMS . . . . .	90
DTW_SAVE_TABLE_IN . . . . .	91
DTW_SET_TOTAL_ROWS . . . . .	92
LOCATION . . . . .	93
LOGIN . . . . .	94
NULL_RPT_FIELD . . . . .	95
PASSWORD . . . . .	96
SHOWSQL . . . . .	97
SQL_STATE . . . . .	98
TRANSACTION_SCOPE . . . . .	99
Net.Data 雜項變數 . . . . .	100
DTW_CURRENT_FILENAME . . . . .	101
DTW_CURRENT_LAST_MODIFIED . . . . .	102
DTW_DEFAULT_MESSAGE . . . . .	103
DTW_LOG_LEVEL . . . . .	104
DTW_MACRO_FILENAME . . . . .	105
DTW_MACRO_LAST_MODIFIED . . . . .	106
DTW_MBMODE . . . . .	107
DTW_MP_PATH . . . . .	108
DTW_MP_VERSION . . . . .	109
DTW_PRINT_HEADER . . . . .	110
DTW_REMOVE_WS . . . . .	111
RETURN_CODE . . . . .	112
<b>第3章 Net.Data 內建函數 . . . . .</b>	<b>113</b>
函數名稱 . . . . .	113
輸入與輸出參數 . . . . .	113
函數結果格式 . . . . .	113
函數參數規則 . . . . .	114
一般函數 . . . . .	114
DTW_ADDQUOTE . . . . .	116
DTW_CACHE_PAGE . . . . .	118
DTW_DATE . . . . .	121
DTW_EXIT . . . . .	123
DTW_GETCOOKIE . . . . .	124
DTW_GETENV . . . . .	126
DTW_GETINIDATA . . . . .	128
DTW_HTMLLENCODE . . . . .	129

DTW_QHTMLENCODE . . . . .	131
DTW_SENDMAIL . . . . .	132
DTW_SETCOOKIE . . . . .	136
DTW_SETENV. . . . .	139
DTW_TIME . . . . .	140
DTW URLESCSEQ . . . . .	142
算術函數 . . . . .	144
DTW_ADD . . . . .	145
DTW_DIVIDE . . . . .	147
DTW_DIVREM . . . . .	149
DTW_FORMAT . . . . .	151
DTW_INTDIV . . . . .	154
DTW_MULTIPLY. . . . .	156
DTW_POWER . . . . .	158
DTW_SUBTRACT . . . . .	160
字串函數 . . . . .	162
DTW_ASSIGN . . . . .	163
DTW_CHARTOHEX. . . . .	164
DTW_CONCAT . . . . .	165
DTW_DELSTR. . . . .	166
DTW_HEXTOCHAR. . . . .	167
DTW_INSERT . . . . .	168
DTW_LASTPOS . . . . .	170
DTW_LENGTH . . . . .	172
DTW_LOWERCASE. . . . .	173
DTW_POS . . . . .	175
DTW_REPLACE . . . . .	177
DTW_REVERSE . . . . .	179
DTW_STRIP . . . . .	180
DTW_SUBSTR. . . . .	182
DTW_TRANSLATE . . . . .	184
DTW_UPPERCASE . . . . .	186
字組函數 . . . . .	188
DTW_DELWORD. . . . .	189
DTW_SUBWORD. . . . .	191
DTW_WORD . . . . .	193
DTW_WORDINDEX. . . . .	194
DTW_WORDLENGTH . . . . .	195
DTW_WORDPOS. . . . .	196
DTW_WORDS . . . . .	198
表格函數 . . . . .	199
DTW_TB_APPENDROW . . . . .	200
DTW_TB_COLS . . . . .	201
DTW_TB_DELETECOL . . . . .	202
DTW_TB_DELETEROW . . . . .	203
DTW_TB_DLIST . . . . .	204
DTW_TB_DUMPH . . . . .	206
DTW_TB_DUMPV . . . . .	207
DTW_TB_GETN . . . . .	209
DTW_TB_GETV . . . . .	211
DTW_TB_HTMLENCODE. . . . .	213
DTW_TB_INPUT_CHECKBOX . . . . .	214

DTW_TB_INPUT_RADIO . . . . .	216
DTW_TB_INPUT_TEXT . . . . .	218
DTW_TB_INSERTCOL . . . . .	220
DTW_TB_INSERTROW . . . . .	221
DTW_TB_LIST . . . . .	222
DTW_TB_QUERYCOLNONJ. . . . .	224
DTW_TB_ROWS . . . . .	226
DTW_TB_SELECT . . . . .	227
DTW_TB_SETCOLS. . . . .	229
DTW_TB_SETN . . . . .	230
DTW_TB_SETV . . . . .	232
DTW_TB_TABLE. . . . .	234
DTW_TB_TEXTAREA . . . . .	236
純本文檔介面函數 . . . . .	238
存取純本文檔資料來源 . . . . .	238
純本文檔介面區隔字元 . . . . .	240
鎖定檔案 . . . . .	241
DTWF_APPEND . . . . .	242
DTWF_CLOSE. . . . .	244
DTWF_DELETE . . . . .	245
DTWF_INSERT . . . . .	247
DTWF_OPEN . . . . .	249
DTWF_READ . . . . .	251
DTWF_REMOVE . . . . .	254
DTWF_SEARCH . . . . .	255
DTWF_UPDATE . . . . .	258
DTWF_WRITE. . . . .	260
Web 登記函數 . . . . .	262
DTWR_ADDENTRY. . . . .	263
DTWR_CLEARREG . . . . .	265
DTWR_CLOSEREG . . . . .	266
DTWR_CREATEREG . . . . .	267
DTWR_DELENTY . . . . .	269
DTWR_DELREG . . . . .	270
DTWR_LISTREG . . . . .	271
DTWR_LISTSUB . . . . .	273
DTWR_OPENREG . . . . .	275
DTWR_RTVENTRY . . . . .	276
DTWR_UPDATEENTRY . . . . .	278
持續巨集函數 . . . . .	280
DTW_ACCEPT. . . . .	281
DTW_COMMIT . . . . .	283
DTW_ROLLBACK . . . . .	284
DTW_RTVHANDLE. . . . .	285
DTW_STATIC . . . . .	286
DTW_TERMINATE . . . . .	287
<b>附錄A. Net.Data 技術圖書庫 . . . . .</b>	<b>289</b>
<b>附錄B. DB2 WWW 連線. . . . .</b>	<b>291</b>
EXEC_SQL . . . . .	291
HTML_INPUT . . . . .	291



HTML_REPORT . . . . .	291
SQL . . . . .	291
SQL_MESSAGE . . . . .	292
SQL_REPORT . . . . .	292
SQL_CODE . . . . .	292
<b>附錄C. Net.Data 作業系統參考手冊 . . . . .</b>	<b>293</b>
<b>附錄D. 注意事項 . . . . .</b>	<b>299</b>
商標 . . . . .	300
名詞解釋 . . . . .	301
索引 . . . . .	303



## 前言

感謝您選購 IBM® 開發工具 -Net.Data®- 來建立動態網頁！經由納入各種資料來源及使用您已知的功能強大之程式設計語言，您可以利用 Net.Data 迅速地開發動態網頁。

---

## 關於 Net.Data

透過 IBM 的 Net.Data 產品，您可以使用關聯式與非關聯式資料庫管理系統（DBMS）的資料建立動態的網頁，這些系統包含 DB2、IMS、ODBC-enabled 資料庫、以及可經由 DRDA 來存取的資料庫，亦可使用以 Java、JavaScript、Perl、C、C++ 與 REXX 等程式設計語言寫成之應用程式來建立。在 Windows NT、AIX、OS/2、OS/390、OS/400、HP-UX、Sun Solaris、Santa Cruz Operating System (SCO) 及 Linux 等作業系統的機器上，Net.Data 產品系列提供類似的功能。

Net.Data 是一個巨集處理器，它以 Web 伺服器機器上的中間軟體之身份來執行。您可撰寫 Net.Data 應用程式（稱為 *macro*），Net.Data 解譯該應用程式以自行設定的內容建立動態網頁，這些自行設定的內容是根據使用者的輸入、資料庫的現行狀態、其它資料來源、現有的企業邏輯以及您指定給巨集的其它因數。

使用格式 URL（一致資源定址器）的要求從瀏覽器（如 Netscape Navigator 或 Internet Explorer）傳送到 Web 伺服器，該伺服器將要求轉送至 Net.Data 以供執行。Net.Data 會尋找並執行巨集，並且會建置一個 Web 網頁，而該 Web 網頁是依據您所寫入的函數來自行設定的。這些函數可以：

- 在使用但不限於 C、C++、RPG、COBOL、JAVA、Perl 或 REXX 程式設計語言撰寫的應用程式內封裝企業邏輯
- 存取資料庫，例如 DB2
- 存取其它資料原始檔如純本文檔

Net.Data 將此 Web 網頁傳給 Web 伺服器，該伺服器會透過網路轉送此網頁以便顯示在瀏覽器上。

Net.Data 可用於一些架構來使用介面如「超本文轉送通信協定」(HTTP) 和「通用閘道介面」(CGI) 的伺服器環境。HTTP 是瀏覽器與 Web 伺服器彼此交談的業界標準介面，CGI 是閘道應用程式如 Net.Data 的 Web 伺服器呼叫的業界標準介面。這些介面可讓您選取最喜歡的瀏覽器或 Web 伺服器來配合 Net.Data 使用。

爲了增進效能，Net.Data 支援多種 Web 伺服器「應用程式設計介面」(API)。此外，可以把 Net.Data 當作 Java servlet 來啓動。

---

## 關於本書

本書說明 Net.Data 語言架構、變數及函數的用法和語法。

本書可能會參照一些已發表但尚未上市的产品或特性。

相關資訊，如 Net.Data 巨集樣本、示範程式及最新的版本，可由下列全球資訊網（WWW）站台中獲得：

- <http://www.software.ibm.com/data/net.data>

- <http://www.as400.ibm.com/netdata>

## 誰應閱讀本書

規劃並撰寫 Net.Data 應用程式的人可以使用本書中的資訊，來瞭解 Net.Data 所提供的語言結構、變數與函數。

爲了瞭解本書中所討論的各種概念，您必須熟悉 Web 伺服器、簡易 SQL 陳述式、HTML（包括使用 HTML 套表）與 *Net.Data* 管理及程式設計手冊中的資訊。

## 關於本書中的範例

本書中所使用的範例將會儘量簡化，以說明特定的觀念。範例目的並不是要描述 Net.Data 結構所使用的全部方法。因此，有些範例只是程式碼片斷，並不能執行。

## 如何閱讀語法圖

下列爲本書中語法圖所引用的規則：

- 從左至右，從上至下，並遵循線條的路徑來讀取語法圖。

▶▶—— 符號表示陳述式的開始。

——▶ 符號表示陳述式將續接下一行。

▶—— 符號表示陳述式接自上一行。

——▶▶ 符號表示陳述式終止。

以 ▶—— 符號作爲開頭，而以 ——▶ 符號作爲結尾，是語法單位的圖案而不是完整陳述式。

- 必要的項目將會出現在水平線條（主路徑）上。

▶▶——必要的項目——▶▶

- 可選用的項目會出現在主路徑下方。

▶▶——必要的項目——  
└可選用的項目┐——▶▶

如果有某個可選用的項目出現在主路徑的上方，則該項目對陳述式的執行並無任何影響，且只爲了可讀性。

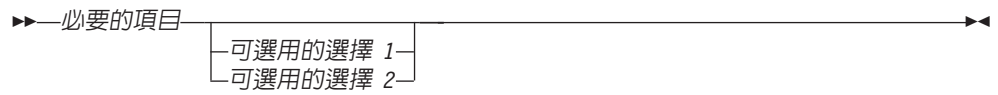
▶▶——必要的項目——  
┐可選用的項目┌——▶▶

- 如果您可以從兩個或兩個以上的項目中選擇，則它們會以垂直堆疊的方式呈現。

如果您必須選擇其中的一個項目，則該堆疊的項目會呈現在主路徑上。

▶▶——必要的項目——  
┐必要的選擇 1┌  
└必要的選擇 2┘——▶▶

如果您可選擇性地選擇其中一個項目，則整個堆疊將會出現在主路徑的下方。



如果其中一個項目是預設值，則它會出現在主路徑的上方，而其它的選項會顯示在下方。



- 若在主線上有一個指回左邊的箭頭，表示該項目可以重複。



如果重複箭頭包含一個標點符號，則您必需使用所指定的標點符號來區隔重複的項目。



在堆疊上方的重複箭頭表示您可以重複堆疊中的項目。

- 關鍵字會以大寫來表示（例如，FROM）。在 `Net.Data` 中，並不限定關鍵字是大寫或小寫。但是非關鍵字的詞彙將會以小寫字母方式呈現（例如，`column-name`）。它們是代表使用者提供的名稱或值。
- 如果有標示標點符號、括弧、數學運算子或其它符號，則您必須將它們當作語法的一部份來輸入。



## 第1章 Net.Data 巨集語言結構

本章說明在 Net.Data 巨集中使用的 Net.Data 巨集語法及語言結構。在 Net.Data 巨集中的語言結構含有關鍵字及陳述式或區塊，可設定不同的變數類型，並執行其它特殊作業，例如併入檔。

本章說明：

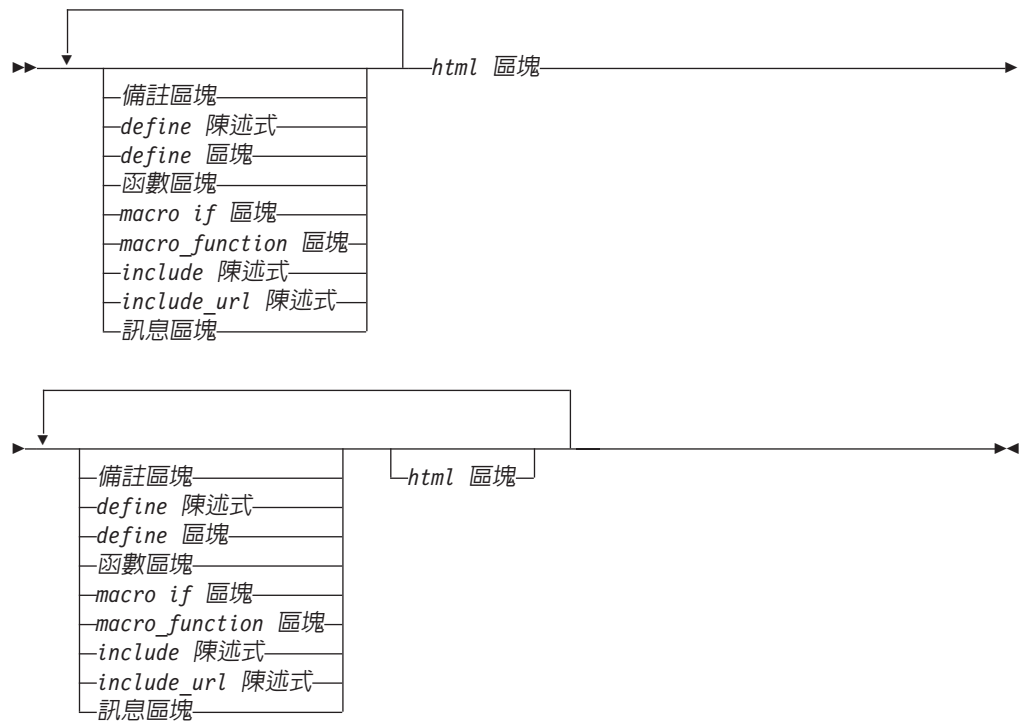
- 『Net.Data 巨集語法』
- 第4頁的『常見語法元素』
- 第5頁的『巨集語言結構』

### Net.Data 巨集語法

Net.Data 巨集是一個文字檔，由一連串的 Net.Data 巨集語言結構所組成，其中包括：

- 設定 Web 網頁的佈置
- 定義變數與函數
- 呼叫巨集中所定義的函數或由 Net.Data 傳遞給語言環境來進行處理的函數

每個陳述式是由一個或多個語言結構所組成，換句話說，就是由關鍵字、特殊的字元、字串、名稱、和變數所組成。下圖描述了語法上有效之 Net.Data 巨集的整體結構。關於整體結構中每一個元素的詳細資訊，請參閱第5頁的『巨集語言結構』。



Net.Data 巨集包含兩個部份：宣告部份與展示部份。您可以任何順序來多次使用這些部分。

- 宣告部分包含巨集中的變數及函數的定義。
- 展示部分包含 HTML 區塊，區塊中含有設定網頁佈置的 HTML 陳述式。這個部份並含有報告區段。

圖1顯示巨集的宣告及展示部份。

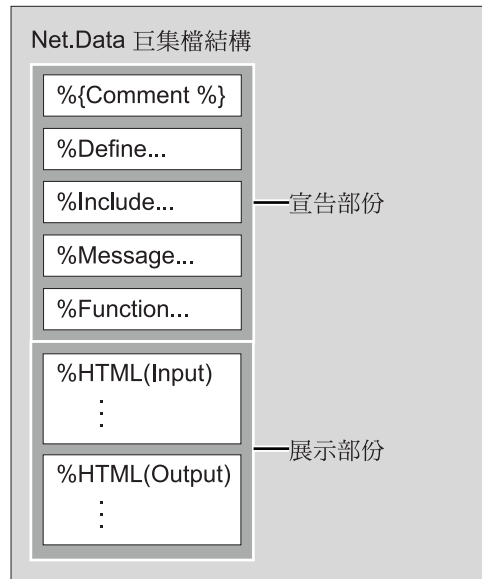


圖 1. 巨集結構

必須先定義在宣告或展示部份中使用的變數及函數，才可將其用於變數參照或函數呼叫。

第3頁的圖2示範巨集的各部份。宣告部份包含 DEFINE 及 FUNCTION 定義區塊。HTML 區塊有如輸入和輸出區塊。



```

%{ ***** Define 區塊 *****}
%DEFINE {
    page_title="Net.Data 巨集模版"
}%

%{ ***** 函數定義區塊 *****}
%FUNCTION(DTW_REXX) rexx1 (IN input) returns(result)
    { %EXEC{ompsamp.cmd %}
}%

%FUNCTION(DTW_REXX) today () RETURNS(result)
    {
        result = date()
    }
}%

%{ ***** HTML 區塊：輸入 *****}
%HTML (INPUT) {
    <html>
    <head>
    <title>$(page_title)<title>
    </head><body>
    <h1>輸入格式</h1>
    今天是@today()

    <FORM METHOD="post" ACTION="output">
    請輸入一些要傳給 REXX 程式的資料：
    <INPUT NAME="input_data" TYPE="text" SIZE="30">
    <p>
    <INPUT TYPE="submit" VALUE="Enter">

    <hr>
    <p>[<a href="/">首頁]
    </body></html>
}%

%{ ***** HTML 區塊：輸出 *****}
%HTML (OUTPUT) {
    <html>
    <head>
    <title>$(page_title)</title>
    </head><body>
    <h1>輸出頁</h1>
    <p>@rexx1(input_data)
    <p><hr>
    <p>[<a href="/">首頁</a> |
    <a href="input">前一頁</a>]
    </body></html>
}%

```

圖 2. 巨集模版格式

Net.Data 巨集語言是自由格式的語言，可讓您有彈性地撰寫您的巨集。除非特別指示，否則會忽略額外的空白字元。每一個 Net.Data 巨集語言結構，及其它幾個您可以使用來定義結構的元素，將在下列部分說明。Net.Data 巨集語言支援 DB2 WWW Connection 語言元素以便能夠與先前的版本相容。雖然這些語言元素已說明於第291頁的『附錄B. DB2 WWW 連線』，但仍建議您使用 Net.Data 語言結構。

這些範例顯示如何在巨集中使用語言結構、變數、函數和其它元素的方式。您可從 Net.Data Web 網頁中下載樣本及示範程式，以獲得更多的範例：

- <http://www.software.ibm.com/data/net.data>
- <http://www.as400.ibm.com/netdata>

## 常見語法元素

下列是在語言結構說明中經常使用的語法元素：

- 『變數名稱』
- 『變數參照』
- 第5頁的『字串』

### 變數名稱

目的：

定義變數。變數是一種物件，其值可於巨集執行期間變更。

變數名稱必須以字母或底線 ( \_ ) 為開頭，並且可包含任何英數字元、底線、# 字號或句點 ( . )。所有變數名稱都是區分大小寫，但除了 `N_columnName` 及 `V_columnName` 之外（關於這些例外狀況的詳細資訊，請參閱第65頁的『Net.Data 表格處理程序變數』。）。

### 變數參照

目的：

傳回變數值，並以 \$ 及 () 符號指定。例如，如果 `VAR = 'abc'`，則 `$(VAR)` 傳回值 'abc'。變數參照是在執行時被加以運算。定義一個 EXEC 陳述式或區塊的變數時，Net.Data 在讀取變數參照的同時，會執行指定的動作。

您可藉由併入變數參照、字串以及變數參照內的函數呼叫，動態產生變數名稱。如果您參照的動態產生變數未遵循變數名稱規則，Net.Data 會將此參照解析為空字串。

變數參照前後的多餘空格會被忽略，而函數呼叫、字串及變數參照之間的多餘空格是不被容許的。如果函數呼叫、字串與變數參照之間有換行字元，則會出現錯誤訊息。含有任何多餘空格的變數參照會傳回空字串。

語法：



註：

- 1 字串只能含有變數名稱所容許的字元：英數字元、底線 ( \_ )、# 字號或句點 ( . )。

#### 範例 1：變數參照

如果您已定義變數 `homeURL`：

```
%DEFINE homeURL="http://www.ibm.com/"
```

您可以將首頁參照為 `$(homeURL)` 並建立鏈結：

```
<a href="$(homeURL)">Home page</a>
```

## 範例 2：動態產生的變數參照

您可以動態產生變數參照來動態參照橫列中的欄位值：

```
%WHILE (INDEX < NUM_COLS) {  
  $(V$(INDEX))  
  @DTW_ADD(INDEX, "1", INDEX)  
  %}
```

## 範例 3：使用巢狀變數參照及函數呼叫的動態變數參照

```
%define my = "my"  
%define u = "lower"  
%define myLOWERvar = "hey"  
$($(my)@dtw_ruppercase(u)var)
```

變數參照會傳回 hey 值。

## 字串

字串中可使用任何英文字母、數字與標點符號。如果字串出現在雙引號內，則不容許使用換行字元。關於與語言結構同用時的限制，請參閱每一個語言結構中的字串參數說明。

雙引號 (") 內的字串，可以包含除了換行字元之外的字元。若是大括弧 ({ %}) 內的字串，則可以是包括換行字元的任何字元。例如，

```
%define multiline = {  
  first line  
  second line  
  %}
```

若要在引號內的字串之內設定雙引號，請使用兩對雙引號。作為函數引數的字串或作為比較表示式的項目可含有雙引號。例如，如果您定義一個字串值為：

```
%DEFINE result = " "Hello world!" " "
```

則結果值為：

```
"Hello world!"
```

HTML 陳述式是一個字串。

作為函數引數、函數項及變數值的字串，可含有變數參照和函數呼叫。在下列範例中，函數呼叫 myfunc2 具有一個含有變數參照與函數呼叫的字串參數。

```
%html(report) {  
  @myfunc2("abc$(var1)@myfunc()")  
  %}
```

將字串傳遞給函數 myfunc2 之前，Net.Data 會解析變數參照 \$(var1) 與函數呼叫 @myfunc()，而不是將它們解譯為字串部份。

---

## 巨集語言結構

本節說明在 Net.Data 巨集中使用的語言結構。

每一個語言結構說明，都有下列資訊：

**目的** 定義使用 Net.Data 巨集中語言結構的目的。

**語法** 提供語言結構之邏輯結構的圖解。

**參數** 定義語法圖內的所有元素，並提供其它語言結構的語法及範例的交叉參考。

**上下文** 解釋語言結構可用在 Net.Data 巨集結構中的何處。

**限制** 定義可以包含哪些元素，並指定用法限制。

**範例** 提供簡單的範例，並說明如何在 Net.Data 巨集中使用關鍵字陳述式或區塊。

下列是用於巨集中的結構；關於語法及範例的詳細資訊，請參閱每一個結構說明。

- 第7頁的『備註區塊』
- 第9頁的『DEFINE 區塊或陳述式』
- 第13頁的『ENVVAR 陳述式』
- 第14頁的『EXEC 區塊或陳述式』
- 第16頁的『FUNCTION 區塊』
- 第23頁的『函數呼叫 (@)』
- 第26頁的『HTML 區塊』
- 第28頁的『IF 區塊』
- 第34頁的『INCLUDE 陳述式』
- 第36頁的『INCLUDE\_URL 陳述式』
- 第38頁的『LIST 陳述式』
- 第40頁的『MACRO\_FUNCTION 區塊』
- 第44頁的『MESSAGE 區塊』
- 第48頁的『REPORT 區塊』
- 第51頁的『ROW 區塊』
- 第53頁的『TABLE 陳述式』
- 第55頁的『WHILE 區塊』

## 備註區塊

### 目的

替 Net.Data 巨集的函數製作文件。因為 COMMENT 區塊可以用在巨集中的任何部份，所以沒有在其它語法圖內加以陳述。

COMMENT 區塊也可用在 Net.Data 起始設定檔中。

### 語法

►►—%{—本文—%}—

### 值

**本文** 一或多行上的任何字串。Net.Data 不處理所有備註內容。

### 上下文

備註可放置在 Net.Data 巨集或 Net.Data 起始設定檔中任何兩個 Net.Data 語言結構之間

### 限制

接受任何本文或字元；但是備註區塊不可為巢狀。

### 範例

#### 範例 1：基本備註區塊

```
%{  
這是備註區塊。不限制行數及字元。Net.Data 不處理它的內容。  
%}
```

#### 範例 2：FUNCTION 區塊中的備註

```
%function(DTW_REXX) getAddress(IN name,  %{ 客戶名稱 %}  
                                IN phone,  %{ 客戶電話號碼 %}  
                                OUT address %{ 客戶地址 %}  
                                )  
  
{  
    ....  
%}
```

#### 範例 3：HTML 區塊中的備註

```
%html(report) {  
  
  %{ 執行查詢並儲存結果於表格 %}  
  @myQuery(resultTable)  
  
  %{ 建置套表以顯示一頁資料 %}  
  <form method="POST" action="report">  
  
  %{ 傳送表格給 REXX 函數，以傳送資料輸出 %}  
  @displayRows(START_ROW_NUM, submit, resultTable, RPT_MAX_ROWS)  
  
  %{ 將 START_ROW_NUM 作為隱藏變數傳給下一筆呼叫 %}  
  <input name="START_ROW_NUM" type="hidden" value="${START_ROW_NUM}">
```

```
%{ 建置下一個或前一個按鈕 %}
%if (submit == "both" || submit == "next_only")
  <input name="submit" type="submit" value="下一個">
%endif
%if (submit == "both" || submit == "prev_only")
  <input name="submit" type="submit" value="前一個">
%endif
</form>
%}
```

#### 範例 4：DEFINE 區塊中的備註

```
%define {
  START_ROW_NUM = "1"          %{ 輸出表格的起始列列號 %}
  RPT_MAX_ROWS = "25"         %{ 表格列數的最大值 %}
  resultTable = %table        %{ 保留查詢結果的表格 %}
%}
```

#### 範例 5：Net.Data 起始設定檔中的備註

```
%{ changes: removed RETURN_CODE parm and DTW_DEFAULT ENVIRONMENT statement %}
...
ENVIRONMENT (DTW_SQL) dtwsq1 (IN LOCATION, DB2SSID, DB2PLAN, TRANSACTION_SCOPE)
ENVIRONMENT (DTW_ODBC) odbcd11 (IN LOCATION, TRANSACTION_SCOPE)
ENVIRONMENT (DTW_PERL) perl111 ()
ENVIRONMENT (DTW_REXX) rexxd11 ()
ENVIRONMENT (DTW_FILE) filed11 ()
ENVIRONMENT (DTW_APPLET) appld11 ()
ENVIRONMENT (DTW_SYSTEM) sysd11 ()
```

## DEFINE 區塊或陳述式

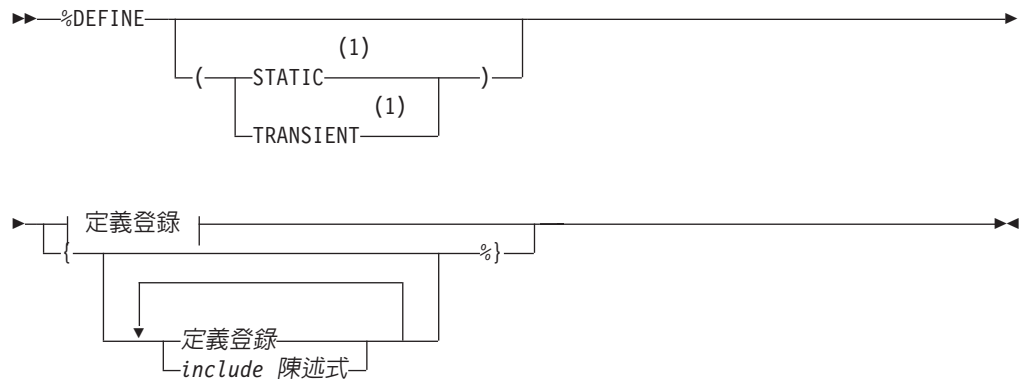
## 目的

DEFINE 區段在巨集中的宣告部份定義變數名稱，可以是陳述式或是區塊。

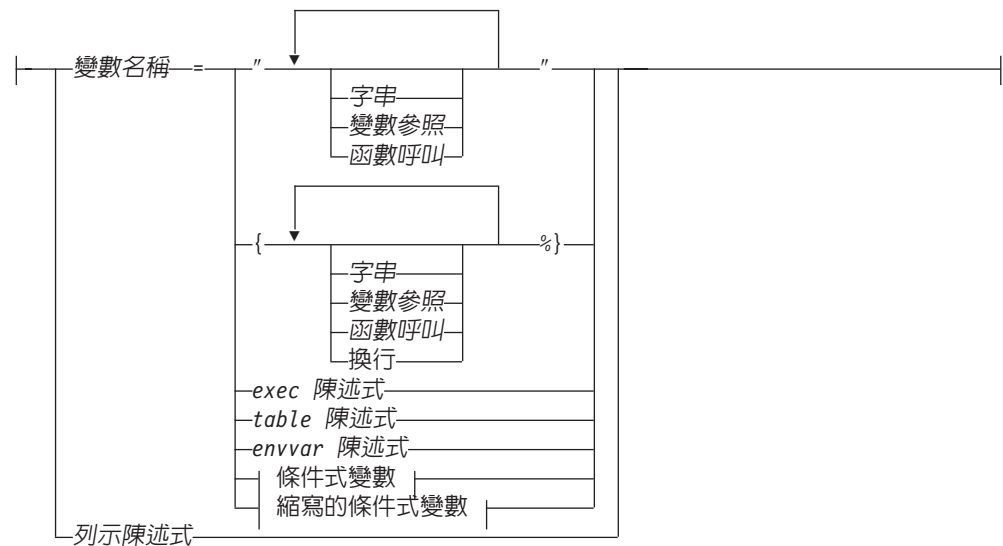
- 使用陳述式，可以一次定義一個變數
- 使用區塊，可以定義數個變數

變數定義使用雙引號 ("")，可以在單一行，或使用大括弧加上百分比符號 ({ %})，可以分成多行。定義了變數之後，您可以在巨集的任何地方參考它。

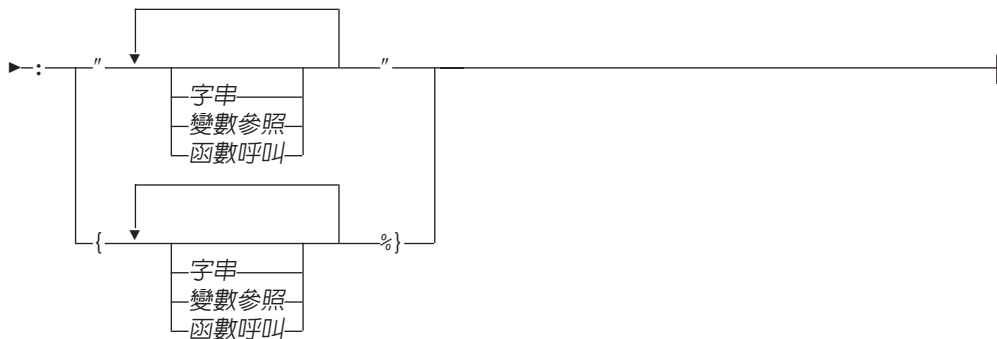
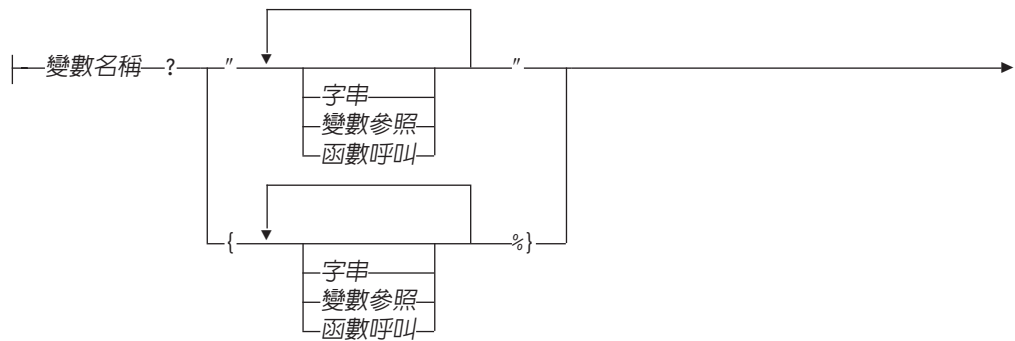
## 語法



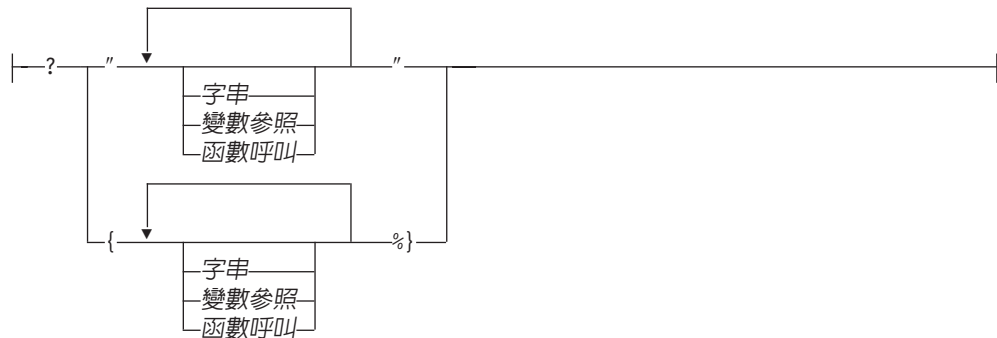
定義登録:



條件式變數：



縮寫的條件式變數：



註：

- 1 STATIC 與 TRANSIENT 均是持續巨集的關鍵字，目前僅能在 OS/400 作業系統上使用它們。

值

**%DEFINE**

此關鍵字負責定義變數。

**STATIC**

指定變數在持續異動內的巨集呼叫之間保留它的值的關鍵字。這是持續巨集的預設值。

**TRANSIENT**

指定變數在巨集呼叫之間不會保留它的值的關鍵字。這是非持續巨集的預設值。



定義登錄：

### 變數名稱

定義變數的名稱。有關語法的詳細資訊，請參閱第4頁的『變數名稱』。

### 字串

字串中可使用任何英文字母、數字與標點符號。如果字串出現在雙引號內，則不容許使用換行字元。

### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

### exec 陳述式

EXEC 陳述式。為參考變數或呼叫函數時所執行的外部程式名稱。有關語法及範例，請參閱第14頁的『EXEC 區塊或陳述式』。

### 表格陳述式

TABLE 陳述式。定義相關資料的集合，有一個相同記錄或橫列的陣列、以及一個說明各列欄位的直欄名稱陣列。有關語法及範例，請參閱第53頁的『TABLE 陳述式』。

### envvar 陳述式

ENVVAR 陳述式。參照環境變數。有關語法及範例，請參閱第13頁的『ENVVAR 陳述式』。

### 條件式變數

設定一個根據另一個變數或字串值而產生的變數值。

### 縮寫的條件式變數

設定一個根據另一個變數或字串值而產生的變數值。條件式變數的縮短格式。

### 列示陳述式

LIST 陳述式。定義一些變數，可用來建置一個有定界符號的值列示。有關語法及範例，請參閱第38頁的『LIST 陳述式』。

### include 陳述式

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。

## 上下文

DEFINE 區塊或陳述式必須在巨集的 IF 區塊之內，或在 Net.Data 巨集宣告部份的其它區塊之外。

## 限制

- 可以包含下列元素：
  - 備註區塊
  - 條件變數
  - LIST 陳述式

- TABLE 陳述式
  - 變數參照
  - INCLUDE 陳述式
  - EXEC 陳述式
  - 函數呼叫
  - ENVVAR 陳述式
- 變數本身的定義內無法使用變數。例如，下列變數定義是不容許的：  

```
%DEFINE var = "The value is $(var)."
```

## 範例

### 範例 1：簡式變數定義

```
%DEFINE var1 = "orders"
%DEFINE var2 = "$(var1).html"
```

執行時，變數參照 `$(var2)` 被運算為 `orders.html`。

### 範例 2：字串內含雙引號

```
%DEFINE hi = "say "hello""
%DEFINE empty = ""
```

顯示時，變數 `hi` 的值為 `say "hello"`。變數 `empty` 是空值。

### 範例 3：多重變數的定義

```
%DEFINE{ DATABASE = "testdb"
          home = "http://www.software.ibm.com"
          SHOWSQL = "YES"
          PI = "3.14150"
%}
```

### 範例 4：變數的多行定義

```
%DEFINE text = {This variable definition
                 spans two lines
%}
```

**範例 5：**這個條件式變數的範例示範如果結果值不包含任何 NULL 值時，變數 `var` 如何取用引號 (") 內的結果值。

```
%DEFINE var = ? "Hello! $(V)@MyFunc()"
%}
```

## ENVVAR 陳述式

### 目的

定義一個變數為 `DEFINE` 區塊中的環境變數。當參考 `ENVVAR` 變數時，`Net.Data` 以相同名稱傳回環境變數的現行值。

### 語法

►►—%ENVVAR—◄◄

### 上下文

`ENVVAR` 陳述式可以在 `DEFINE` 區塊或陳述式中。

### 值

#### %ENVVAR

此關鍵字用於在 `DEFINE` 區塊中定義一個變數成為環境變數。這個變數可在巨集的任何部份取得環境變數值。

### 限制

`ENVVAR` 陳述式不可以包含其它元素。

### 範例

**範例 1：**在本例中，`ENVVAR` 定義一變數，此變數被參考到時會傳回現行值給環境變數 `SERVER_SOFTWARE`，亦即 Web 伺服器名稱。

```
%DEFINE SERVER_SOFTWARE = %ENVVAR
```

```
%HTML(REPORT) {  
伺服器是 $(SERVER_SOFTWARE)。  
%}
```

## EXEC 區塊或陳述式

### 目的

當參考變數或呼叫函數時，指定一個執行的外部程式名稱。

當 Net.Data 在巨集中發現一個執行變數時，它會使用下列方法尋找被參考到的可執行程式：

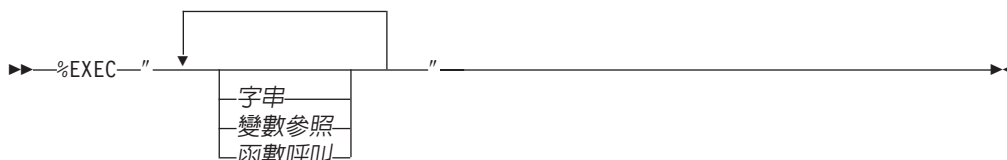
1. 它會搜尋 Net.Data 起始設定檔案中的 EXEC\_PATH。有關 EXEC\_PATH 的詳細資訊，請參閱您作業系統的 *Net.Data* 管理和程式設計指南中架構一章。
2. 如果 Net.Data 找不到該程式，它會搜尋系統所定義的目錄。如果找到了可執行的程式，Net.Data 會執行該程式。

**授權要訣：**確定 Net.Data 執行時所用的使用者 ID，有權存取 EXEC 陳述式或區塊所參照的任何檔案。有關設定 Web 伺服器對 Net.Data 檔案的存取權之詳細資訊，請參閱您作業系統的 *Net.Data* 管理和程式設計指南中架構這一章。

EXEC 陳述式及區塊用於兩個不同的上下文，二者的語法不同，視使用位置而定。EXEC 陳述式用於 DEFINE 區塊，EXEC 區塊用於 FUNCTION 區塊。

### 語法

EXEC 陳述式用於 DEFINE 區塊：



EXEC 區塊語法用於 FUNCTION 區塊：



### 值

#### %EXEC

此為一個關鍵字，指定參照變數或呼叫函數時，須執行的外部程式名稱。Net.Data 遇到 EXEC 陳述式定義的變數參照時，會處理 EXEC 陳述式宣告的變數內容。

#### 字串

字串中可使用任何英文字母、數字與標點符號。如果字串出現在雙引號內，則不容許使用換行字元。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

## 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

## 上下文

在這些上下文中可找到 EXEC 區塊或陳述式：

- DEFINE 區塊
- FUNCTION 區塊

## 限制

EXEC 區塊或陳述式可以包含這些元素：

- 備註區塊
- 字串
- 變數參照
- 函數呼叫

下列 Net.Data 提供的語言環境支援 EXEC 陳述式：

- REXX
- 系統
- Perl

## 範例

**範例 1：**被變數參考的可執行檔

```
%DEFINE mycall = %EXEC "MYEXEC.EXE $(empno)"

%HTML (report){
<P>這是您要求的的報告：
<HR>$(mycall)
%}
```

這個範例會針對每次對變數 mycall 的參照而執行 MYEXEC.EXE。

**範例 2：**被函數參考的可執行檔

```
%FUNCTION(DTW_REXX) my_rexx_pgm(INOUT a, b, IN c, INOUT d){
  %EXEC{ mypgm.cmd 這是測試 %}
%}
```

當函數 my\_rexx\_pgm 被呼叫時，這個範例執行 mypgm.cmd。

## FUNCTION 區塊

### 目的

定義 Net.Data 從巨集呼叫的次常式。FUNCTION 區塊中的可執行檔陳述式，可以包含直接由語言環境解譯過來的語言陳述式，也可以指示呼叫外部程式。

**FUNCTION 區塊中的 EXEC 區塊：** 如果您在 FUNCTION 區塊中使用的 EXEC 區塊，它必須是 FUNCTION 區塊中的唯一可執行檔陳述式。將可執行檔陳述式傳給語言環境之前，Net.Data 把 EXEC 陳述式中的程式檔案名稱，附加到起始設定檔之 EXEC\_PATH 架構陳述式決定的路徑名稱中。最後將產生的字串，傳給要執行的語言環境。

語言環境處理 EXEC 區塊所用的方法，取決於特定的語言環境；REXX、System 及 Perl Net.Data 提供的語言環境支援 EXEC 區塊。

**在語言陳述式中使用特殊字元：** 當符合 Net.Data 語言結構語法的字元於函數區塊的語言陳述式區段中使用，作為語法有效的內含程式碼 (如 REXX 或 Perl) 的一部份時，可能會將它們誤譯為 Net.Data 語言結構，而導致在巨集內發生錯誤或無法預測的結果。

例如，Perl 函數必須使用 COMMENT 區塊定界字元 (%{)。當執行巨集時， %{ 字元會被解譯為 COMMENT 區塊的開頭。然後，Net.Data 會尋找 COMMENT 區塊的尾端，所以當它讀取函數區塊的尾端時，它會認為找到 COMMENT 區塊的尾端。Net.Data 隨後會繼續尋找函數區塊的尾端，當它找不到時，便會發出錯誤。

請使用下列一種方法，使用 Net.Data 特殊字元作為內含的程式碼的一部份，不讓 Net.Data 將它們解譯為特殊字元：

- 使用 EXEC 陳述式呼叫程式碼，而不是將程式碼列入。
- 使用變數參照來指定特殊字元。

例如，下列 Perl 函數含有代表 COMMENT 區塊區隔字元 (%{) 作為 Perl 語言陳述式一部份的字元：

```
%function(DTW_PERL) func() {  
    ...  
    for $num_words (sort bynumber keys %{ $Rtitles{$num} }) {  
        &make_links($Rtitles{$num}){$num_words};  
    }  
    ...  
}%}
```

若要確定 Net.Data 將 %{ 字元解譯為 Perl 原始程式，而不是 Net.Data COMMENT 區塊區隔字元，請用下列方式之一重寫函數：

- 使用 %EXEC 陳述式：

```
%function(DTW_PERL) func() {  
    %EXEC{ func.pr1 %}  
}%}
```

- 使用變數參照來指定 %{ 字元：

```
%define percent_openbrace = "%{"  
  
%function(DTW_PERL) func() {  
    ...  
    for $num_words (sort bynumber keys ${percent_openbrace} $Rtitles{$num} )) {
```

```

        &make_links($Rtitles{$num}{$num_words});
    }
    ...
%}

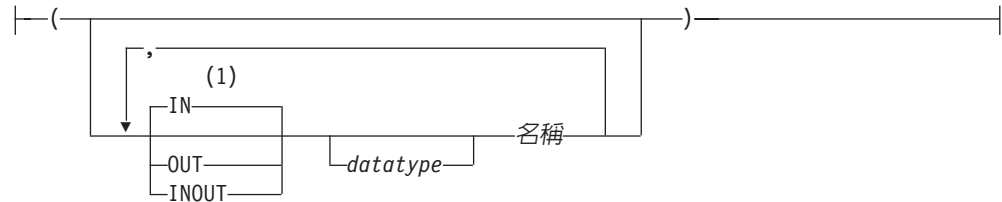
```

## 語法

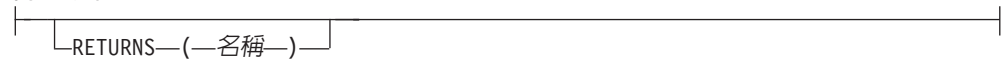
► %FUNCTION—(—語言環境—)—函數名稱— 參數傳送規格 —————►

┌;—————┐  
└傳回規格—{|— 函數本體 —|}%┘—————►

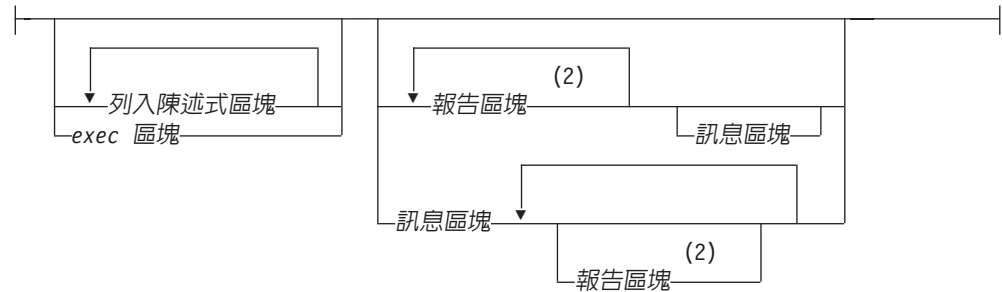
### 參數傳送規格:



### 傳回規格:



### 函數本體:



### 註:

- 1 在參數列示開頭沒有指定的參數類型時，引用預設值參數 IN 類型。無參數類型的參數使用最新指定在參數列示上的類型，或若無指定類型時使用 IN 類型。例如，在參數列示 -( -parm1 - , INOUT -parm2 - , -parm3 - , OUT -parm4 - , -parm5 - )中，參數 -parm1、-parm3 及 -parm5 沒有參數型類。因為沒有指定最初的參數類型，所以參數 -parm1 是 IN 類型。參數 -parm3 是 INOUT 類型，因為它是最新指定的參數類型。同樣地，參數 -parm5 有 OUT 的類型，因為它是最近在參數列示中指定的參數類型。
- 2 重複的報表區塊對下列有效：
  - -處理儲存程序時所在的 SQL 及 ODBC 語言環境，這些儲存程序對 OS/390 作業系統傳回多重結果設定。

- 呼叫 OS/400、OS/2、Windows NT 及 UNIX 作業系統之任何語言環境的函數。

## 值

### %FUNCTION

指定 Net.Data 從巨集呼叫次常式的關鍵字。

### 語言環境

處理函數本體的語言環境。有關詳細資訊，請參閱 *Net.Data 管理和程式設計指南*。

### 函數名稱

定義的函數名稱，是一個以英文字母或底線開頭的英文或數字字串，可以是英文字母、數字或底線字元的任意組合。

### 參數傳送規格：

**IN** 指定 Net.Data 傳送輸入資料給語言環境。IN 是預設值。

### OUT

指定語言環境傳回輸出資料給 Net.Data。

### INOUT

指定 Net.Data 傳送輸入資料給語言環境，及語言環境傳回輸出資料給 Net.Data。

### datatype

指定參數的資料類型。參數的資料類型。有關儲存程序支援的資料類型列示，請參閱 *Net.Data 參考手冊* 的作業系統附錄。

### 名稱

一個以英文字母或底線開頭的英文或數字字串，可以是英文字母、數字或底線字元的任意組合。

### 傳回規格：

### RETURNS

宣告函數完成之後包含語言環境指定的函數值之變數。

### 函數本體：

#### 列入陳述式區塊

函數定義中指定之語言環境的有效語法陳述式，例如：REXX、SQL 或 Perl。有關您所使用語言環境的說明，請參閱 *Net.Data 管理和程式設計指南*。有關語法及用法，請參閱程式設計語言的參考手冊。代表列入陳述式區塊的字串可以包含 Net.Data 變數參照及函數呼叫，而於執行列入陳述式區塊（程式）前，這些變數參照與函數呼叫便會先被求值。

#### exec 區塊

此為 EXEC 區塊。為參考變數或呼叫函數時所執行的外部程式名稱。有關語法及範例，請參閱第14頁的『EXEC 區塊或陳述式』。

#### 報告區塊

REPORT 區塊。函數呼叫的輸出的格式化指示。您可以在報告中使用表頭及註腳資訊。有關語法及範例，請參閱第48頁的『REPORT 區塊』。

#### 訊息區塊

MESSAGE 區塊。為一組回覆碼、相關訊息以及傳回函數呼叫時 Net.Data 執行的動作。有關語法及範例，請參閱第44頁的『MESSAGE 區塊』。



## 上下文

在這些上下文中可找到 FUNCTION 區塊：

- IF 區塊
- Net.Data 巨集宣告部份的任何區塊或陳述式之外。

## 限制

- FUNCTION 區塊可以包含這些元素：
  - 備註區塊
  - EXEC 區塊
  - MESSAGE 區塊
  - REPORT 區塊
  - 列入陳述式區塊
- 不含任何 Net.Data 變數參照或函數呼叫的最長連續列入陳述式區塊字串，其長度限制如下：
  - 對於 OS/2 及 Windows NT 或 UNIX 而言：無限制
  - OS/390：256KB
  - OS/400：256KB
- 列入陳述式區塊中的 SQL 陳述式可以有下列長度。您的資料庫可能有不同的限制；請參閱資料庫文件，以了解資料庫是否有較小的限制。如果 IBM DB2 資料庫限制不同於 Net.Data 限制，則如下所示：
  - 對於 OS/2、Windows NT 及 UNIX 而言：64 KBDB2 具有下列限制：
  - DB2 Universal Database V6 或以上版本：64 KB
  - DB2 Universal Database V5.2 或以下版本：32 KB
  - 若是 OS/390：32 KB
  - 若是 OS/400：32 KB

## 範例

下面是一般範例，並未涵蓋所有的語言環境。有關在特定的語言環境中使用 FUNCTION 區塊的詳細資訊，請參閱 *Net.Data 語言環境參考手冊*。

### 範例 1：REXX 子字串函數

```
%DEFINE lstring = "longstring"
%FUNCTION(DTW_REXX) substring(IN x, y, z) RETURNS(s) {
  s = substr("$x)", $(y), $(z));
}%
%DEFINE a = {@substring(lstring, "1", "4")}% { assigns "long" to a %}
```

在求 *a* 的值時，找到 @ 子字串函數呼叫並執行子字串 FUNCTION 區塊。FUNCTION 區塊中的可執行陳述式會被置換，然後字串 *s* = substr("longstring", 1, 4) 被傳送到 REXX 直譯器執行。因為指定了 RETURNS 子句，所以 @子字串指定的值在 *a* 的運算式中被置換成『long』，即為 *s* 的值。

### 範例 2：呼叫外部 REXX 程式

- Net.Data 巨集：

```
%FUNCTION(DTW_REXX) my_rexx_pgm(INOUT a, b, IN c, OUT d) {
%EXEC{ mypgm.cmd 這是測試 %}
%}
%HTML(INPUT) {
<P> 原始變數值：$(w) $(x) $(z)
<P> @my_rexx_pgm(w, x, y, z)
<P> 已修改之變數值：$(w) $(x) $(z)
%}
```

變數 *w* 與 *x* 對應至函數中的 INOUT 參數 *a* 與 *b*。它們的值與對應至 IN 參數 *c* 的值 *y*，應已從 HTML 表格輸入或 DEFINE 陳述式定義。變數 *a* 與 *b* 在參數 *a* 與 *b* 傳回值時，會被指定新值。當 OUT 參數 *d* 傳回一個值時，即定義了變數 *z*。

- REXX 程式 mypgm.cmd：

```
/* 範例 2 的 REXX 程式範例 */
/* 測試引數 */
num_args = arg();
say '有' num_args 'arguments';
do i = 1 to num_args;
    say 'arg' i '是' arg(i)
end;
/* 設定 Net.Data 傳來的變數 */
d = a || b || c; /* 連結 a, b, c 形成 d */
a = ''; /* 將 a 重設為空字串 */
b = ''; /* 將 b 重設為空字串 */
return;
```

- 自 mypgm.cmd 輸出：

```
有 1 引數
引數 1 是 "這是測試"
```

EXEC 陳述式告訴 REXX 語言環境去通知 REXX 直譯器開始執行外部 REXX 程式 mypgm.cmd。因為 REXX 語言環境可以直接與 REXX 程式共用 Net.Data 變數，所以由它在執行 mypgm.cmd 前，會指定 REXX 變數 *a*、*b* 及 *c* 為 Net.Data 變數 *w*、*x* 及 *y* 的值。mypgm.cmd 可以直接在 REXX 陳述式中使用變數 *a*、*b* 及 *c*。當程式終止時，REXX 變數 *a*、*b* 及 *d* 從 REXX 程式取回，它們的值被指定給 Net.Data 變數 *w*、*x* 及 *z*。因為 RETURNS 子句沒有被用在 my\_rexx\_pgm FUNCTION 區塊的定義之內，所以 @my\_rexx\_pgm 函數呼叫的值為空字串，『』，（如果回覆碼為 0）或為 REXX 程式回覆碼（如果回覆碼不是零）。

### 範例 3：SQL 查詢及報告

```
%FUNCTION(DTW_SQL) query_1(IN x, IN y) {
    SELECT customer.num, order.num, part.num, status
    FROM customer, order, shippingpart
    WHERE customer.num = '$(x)'
        AND customer.ordernumber = order.num
        AND order.num = '$(y)'
        AND order.partnumber = part.num
%REPORT{
    <P>這是您的訂購狀態：
    <P>$(NLIST)
<UL>
%ROW{
    <LI>$(V1) $(V2) $(V3) $(V4)
    %}
</UL>
    %}
%}
%DEFINE customer_name="IBM"
```

```
%DEFINE customer_order="12345"
%HTML(REPORT) {
  @query_1(customer_name, customer_order)
%}
```

@query\_1 函數呼叫在 SELECT 陳述式中以 IBM 取代 \$(x)，以 12345 取代 \$(y)。因為 SQL 函數 query\_1 的定義中沒有定義輸出表格變數，所以使用預設的表格（有關詳細資料，請參閱 TABLE 表格區塊）。REPORT 區塊中參照的 NLIST 及 Vi 變數，是由預設的表定義所定義。REPORT 區塊產生的報告被放置在呼叫 query\_1 函數的輸出 HTML。

#### 範例 4：執行 Perl script 的系統呼叫

- Net.Data 巨集：

```
%FUNCTION(DTW_SYSTEM) today() RETURNS(result) {
  %exec{ perl "today.prl" %}
%}
%HTML(INPUT) {
  @today()
%}
```

- Perl 程式 today.prl：

```
$date = 'date';
chop $date;
open(DTW, "> $ENV{DTWPIPE}") || die "無法開啓：$!";
print DTW "result = \"$date\"\n";
```

「系統」語言環境解譯 FUNCTION 區塊中的可執行檔陳述式時，是經由 C 語言 system() 函數呼叫，傳給作業系統。這個方法無法向 REXX 語言環境般，直接向可執行檔陳述式傳收 Net.Data 變數。「系統」語言環境是以下列方法傳收變數：

- 把輸入參數當作系統環境變數，以 putenv() 函數來傳送，以執行程式來取回。不同語言參照不同的變數。UNIX cshell script 參考環境變數時，會在環境變數名稱之前加上 '\$'，例如 \$x。Perl 語言 script 參考環境變數時，會引用相關陣列 %ENV，例如 %ENV{'x'}。DOS 批次 (.BAT) 則參考含括在百分比符號內的變數名稱，例如 %x%。
- 輸出參數被傳回語言環境時，是經由寫入一個管路，而該管路名稱是環境變數 DTWPIPE 所傳送的，但除了 OS/400 平台之外，因為該平台上的輸出參數是以系統環境參數傳回系統環境。寫入具名管線的資料格式為 name="value"，與 DEFINE 陳述式中的格式一樣。若以這種方式編寫輸出參數對應的變數名稱，新值會置換現行值。如果您編寫的變數名稱未對應到輸出參數，系統不予處理。

遇到 @today 函數呼叫時，Net.Data 將對可執行檔陳述式進行變數替代。本例的可執行檔陳述式中沒有 Net.Data 變數，所以不替代變數。可執行檔陳述式及參數傳給「系統」語言環境後，此語言環境立即建立一個具名管線，並將環境變數 DTWPIPE 設定為具名管線。

然後，再以 C system() 函數呼叫，呼叫外部程式。外部程式以唯寫方式開啓管路，再以標準資料流檔的方式將輸出參數的值寫入管路中。外部程式藉由寫入 STDOUT 產生 HTML 輸出。本例將系統日期程式的輸出，指定至 FUNCTION 區塊 RETURNS 子句識別的變數結果中。結果變數的這個值，則會置換 HTML 區塊中的 @today() 函數呼叫。

#### 範例 5：Perl 語言環境

```
%FUNCTION(DTW_PERL) today() RETURNS(result) {
  $date = 'date';
  chop $date;
```

```

    open(DTW, "> $ENV{DTWPIPE}") || die "無法開啓: $!";
    print DTW "result = \"$date\\\"\\n";
%}
%HTML(INPUT) {
    @today()
%}

```

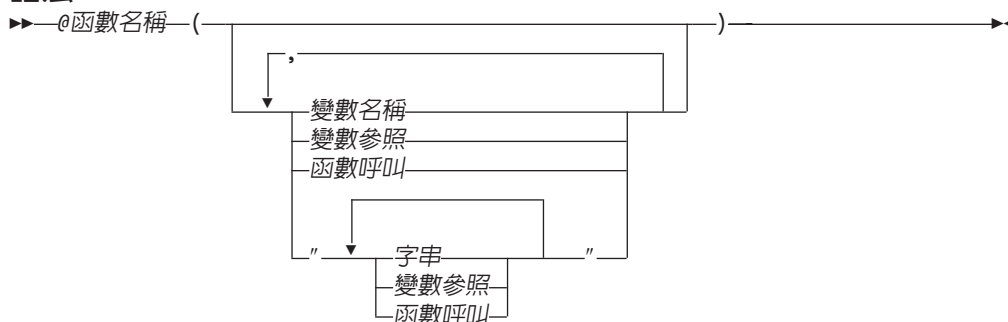
您可以比較本例與「範例 4」中 EXEC 陳述式的使用方式。範例 4 的「系統」語言環境不瞭解如何解譯 Perl 程式，但知道如何呼叫外部程式。EXEC 陳述式要它以外部程式的方式呼叫 perl 程式。實際的 Perl 語言陳述式，是由外部 Perl 程式解譯。範例 5 的 Perl 語言環境可以直接解譯 Perl 語言陳述式，所以沒有 EXEC 陳述式。

## 函數呼叫 (@)

### 目的

呼叫 FUNCTION 或 MACRO\_FUNCTION 區塊，或具有指定引數的內建函數。如果該函數不是內建函數，您必需先在 Net.Data 巨集定義函數後，才能指定函數呼叫。

### 語法



### 值

#### @函數名稱

任何現有函數的名稱。一個以英文字母或底線開頭的英文或數字字串，可以是英文字母、數字或底線字元的任意組合。

#### 變數名稱

定義變數的名稱。有關語法的詳細資訊，請參閱第4頁的『變數名稱』。

#### 字串

字串中可使用任何英文字母、數字與標點符號，但不能使用換行字元。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

#### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。

### 上下文

在這些上下文中可找到函數呼叫：

- HTML 區塊
- REPORT 區塊
- ROW 區塊
- DEFINE 區塊
- IF 區塊
- MACRO\_FUNCTION 區塊
- MESSAGE 區塊
- WHILE 區塊

- 函數呼叫陳述式
- Net.Data 巨集宣告部份的區塊之外

## 限制

- 函數呼叫可以包含這些元素：
  - 備註區塊
  - 字串
  - 函數呼叫
  - 變數參照
- OUT 或 INOUT 參數值不能含有變數參照、函數呼叫或文字字串。

## 範例

### 範例 1：呼叫結構化查詢語言函數 formQuery

```
%FUNCTION(DTW_SQL) formQuery(){
SELECT $(queryVal) from $(tableName)
%}

%HTML (input){
<P>想察看 $(tableName) 的哪些直欄？
<FORM METHOD="POST" ACTION="report">
<INPUT NAME="queryVal" TYPE="CHECKBOX" VALUE="名稱">名稱
<INPUT NAME="queryVal" TYPE="CHECKBOX" VALUE="郵件">電子郵件
<INPUT NAME="queryVal" TYPE="CHECKBOX" VALUE="傳真">傳真
<INPUT TYPE="SUBMIT" VALUE="提出要求">
%}

%HTML (report){
<P>這些是您所選取的直欄：
<HR>@formQuery()
%}
```

### 範例 2：以輸入及輸出參數呼叫 REXX 函數

```
%FUNCTION(DTW_REXX) my_rexx_pgm(INOUT a, b, IN c, OUT d) {
%EXEC{ mypgm.cmd 這是測試 %}
%}
%HTML(INPUT) {
<P> 原始變數值：$(w) $(x) $(z)
<P> @my_rexx_pgm(w, x, y, z)
<P> 已修改之變數值：$(w) $(x) $(z)
%}
```

### 範例 3：以使用變數參照及函數呼叫的輸入參數呼叫 REXX 函數

```
%FUNCTION(DTW_REXX) my_rexx_pgm(IN a, b, c, d, OUT e) {
...
%}
%HTML(INPUT){
<p> @my_rexx_pgm$(myA), @getB(), @retrieveC(), $(myD), myE)
%}
```

### 範例 4：闡釋 INOUT 參數用法的巨集。

```
%DEFINE a = "initial value of a"
%FUNCTION(DTW_REXX) func1(INOUT x) {
Say 'value at start of function:<br>
Say 'x =' x
Say '
```

```

<p>
'
    x = "new value of a"
    %REPORT {

<p>
value at start of report block:<br>
    x = $(x)<br>
    @dtw_assign(x, "newest value of a")
    value at end of report block:<br>
    x = $(x)<br>
    %}
%}
%HTML(report) {
    initial values:<br>
    a = $(a)<br>
    @func1(a)
    value after function call:<br>
    a = $(a)<br>
%}

```

輸出結果：

```

initial values:
a = initial value of a
value at start of function:
x = initial value of a
value at start of report block:
x = new value of a
value at end of report block:
x = newest value of a
value after function call:
a = newest value of a

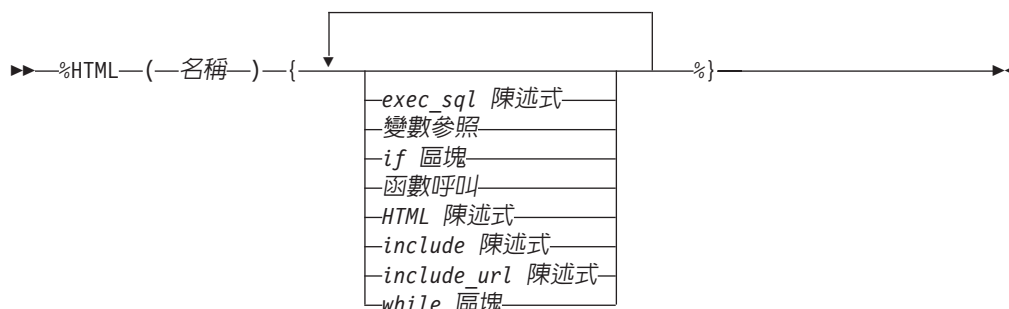
```

## HTML 區塊

### 目的

定義網頁的呈現方式。所要執行的 HTML 區塊名稱，會在呼叫 Net.Data 時於 URL 上指定。HTML 區塊可以包含大部份 Net.Data 巨集語言陳述式及任何有效的展示陳述式，例如 HTML 和 Javascript。

### 語法



### 值

#### %HTML

指區塊包含在從屬站瀏覽器顯示之 HTML 標籤和文字的關鍵字。

#### 名稱

一個以英文字母或底線開頭的英數字字串，可以是英文字母、數字或底線字元的任意組合，包括句點。

#### EXEC\_SQL 陳述式

這是為了提供相容性才支援的 DB2WWW 版次 1 語言元素。請參閱第291頁的『附錄B. DB2 WWW 連線』或「DB2 全球資訊網版次 1」文件。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

#### IF 區塊

此為 IF 區塊。執行條件字串處理。條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有單一加號 (+) 或減號 (-) 置於前端。關於語法及範例，請參閱第28頁的『IF 區塊』。

#### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

#### HTML 陳述式

包含任何英數字元，以及從屬站瀏覽器的格式化 HTML 標籤。

#### include 陳述式

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。



### **include\_url 陳述式**

INCLUDE\_URL 陳述式。讀取另一個檔案，併入指定此陳述式的 Net.Data Web 巨集中。指定的檔案，可能在區域或遠端伺服器中。有關語法及範例，請參閱第36頁的『INCLUDE\_URL 陳述式』。

### **WHILE 區塊**

WHILE 區塊。以條件式字串處理程序執行迴路。有關語法及範例，請參閱第55頁的『WHILE 區塊』。

## **上下文**

在這些上下文中可找到 HTML 區塊：

- IF 區塊
- Net.Data 巨集宣告部份的區塊之外

## **限制**

HTML 區塊可以包含這些元素：

- 備註區塊
- EXEC\_SQL 陳述式
- IF 區塊
- HTML 陳述式
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式
- WHILE 區塊
- 變數參照
- 函數呼叫

## **範例**

**範例 1：**帶有標題及表尾之併入檔的 HTML 區塊

```
%HTML(example1){
%INCLUDE"header.html"
<P>您可以放置<EM>任何</EM> HTML 於 HTML 區塊中。
SQL 函數呼叫大致如下：
@xmp1()
%INCLUDE"footer.html"
%}
```

**範例 2：**包含句點的 HTML 區塊名稱

```
%HTML(my.report){
%INCLUDE"header.html"
<P>您可以放置<EM>任何</EM> HTML 於 HTML 區塊中。
SQL 函數呼叫大致如下：
@xmp1()
%INCLUDE"footer.html"
%}
```

## IF 區塊

## 目的

執行條件字串處理。IF 區塊提供測試一或多個條件的能力，然後基於條件測試的結果執行一區塊的陳述式。您可以在 Net.Data 巨集內宣告部份、HTML 區塊、MACRO\_FUNCTION 區塊、REPORT 區塊、WHILE 區塊及 ROW 區塊中使用 IF 區塊，亦可在另一個 IF 區塊內建立巢狀。

條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有單一加號 (+) 或減號 (-) 置於前端。

**限制：**Net.Data 不支援非整數的數值比較；例如，浮點數字。

**巢狀 IF 區塊：**IF 區塊語法的規則由巨集中區塊的位置決定。如果 IF 區塊在宣告部份任何其它區塊之外的 IF 區塊之中建立巢狀，則其之外區塊能使用的任何元素它也可以使用。 如果 If 區塊在一個 IF 區塊中的其它區塊之中建立巢狀，則它依循所在之區塊的語法規則。

在下列範例，巢狀 IF 區塊必須遵循當它在 HTML 區塊之中時的規則。

```

    %IF 區塊
...
    %HTML 區塊
...
    %IF 區塊

```

您最多可以用巢狀方式設定 1024 個 IF 區塊。

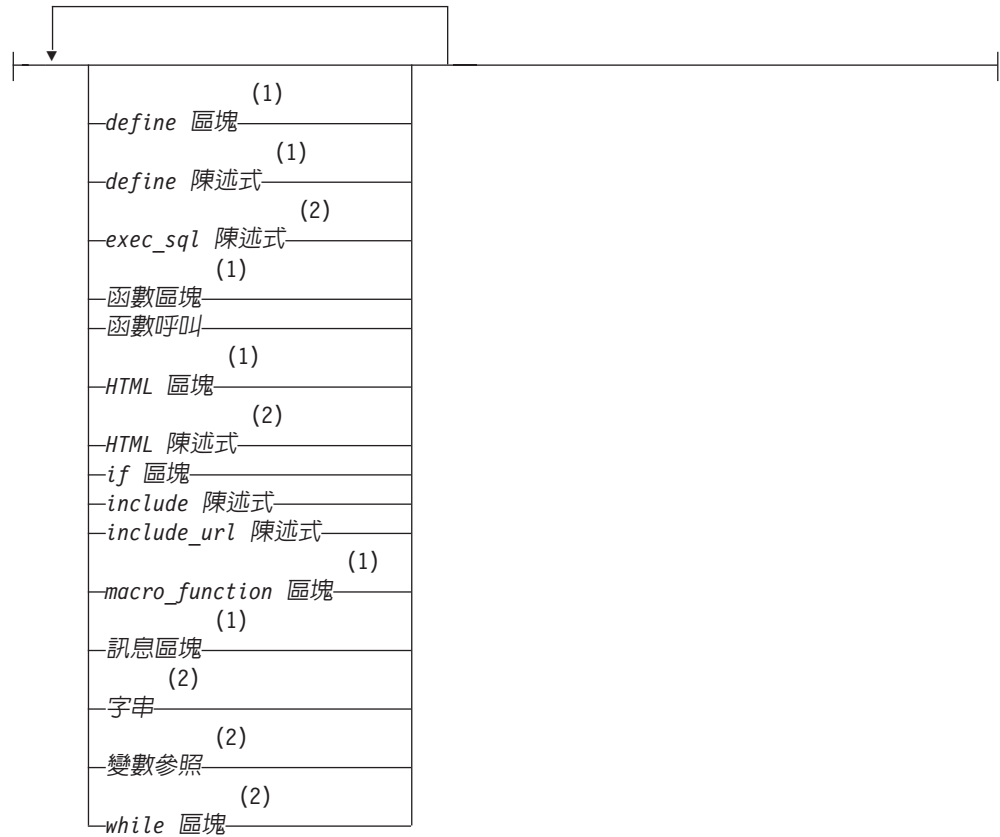
## 語法

►►%IF 條件列示 陳述式區塊 else\_if 規格 %ENDIF◄◄

條件列示：

(	(—條件列示—)	)
	條件列示—&&—條件列示	
	條件列示—  —條件列示	
	!—條件列示	
	條件	
	詞彙	

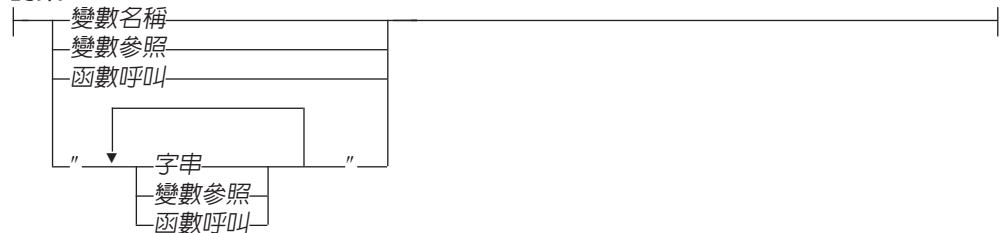
陳述式區塊：



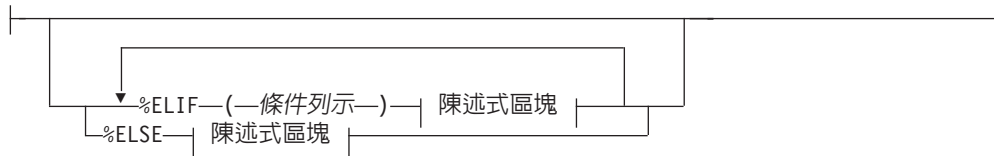
條件:



詞彙:



else\_if 規格:



註:

- 1 當 IF 區塊位於巨集的宣告部份中任何其它區塊之外時，這個語言結構有效。
- 2 當 IF 區塊位於 HTML 區塊、MACRO\_FUNCTION 區塊、REPORT 區塊、ROW 區塊或 WHILE 區塊中時，這個語言結構有效。

## 值

### %IF

此關鍵字指定條件字串處理。

### 條件列示

比較條件及詞彙的值。使用布林運算子，可以連接條件列示。您可以在另一個條件列示中建立巢狀條件列示。

### 陳述式區塊

下列有效的 Net.Data 巨集結構。請參閱圖解及附註，以決定巨集結構有效的上下文。

#### DEFINE 陳述式

DEFINE 區塊或陳述式。定義變數，設定架構變數。變數名稱必須以一個字母或底線開頭，可以使用任何英數字元或底線。有關語法及範例，請參閱第9頁的『DEFINE 區塊或陳述式』。

#### EXEC\_SQL 陳述式

這是為了提供相容性才支援的 DB2WWW 版次 1 語言元素。請參閱第291頁的『附錄B. DB2 WWW 連線』或「DB2 全球資訊網版次 1」文件。

#### FUNCTION 區塊

一種關鍵字，指定一個可以從 Net.Data 巨集呼叫的次常式。FUNCTION 區塊中的可執行檔陳述式，可以包含直接由語言環境解譯的語言陳述式，也可以指示呼叫外部程式。有關語法及範例，請參閱第16頁的『FUNCTION 區塊』。

#### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

#### HTML 區塊

包含任何英數字元，以及從屬站瀏覽器的格式化 HTML 標籤。

#### HTML 陳述式

包含任何英數字元，以及從屬站瀏覽器的格式化 HTML 標籤。

#### IF 區塊

此為 IF 區塊。執行條件字串處理。條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有單一加號 (+) 或減號 (-) 置於前端。

### **include 陳述式**

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。

### **include\_url 陳述式**

INCLUDE\_URL 陳述式。讀取另一個檔案，併入指定此陳述式的 Net.Data Web 巨集中。指定的檔案，可能在區域或遠端伺服器中。有關語法及範例，請參閱第36頁的『INCLUDE\_URL 陳述式』。

### **macro\_function 區塊**

一種關鍵字，指定一個可以從 Net.Data 巨集呼叫的次常式。在 MACRO\_FUNCTION 區塊中的可執行的陳述式可以包含 Net.Data 巨集語言來源陳述式。有關語法及範例，請參閱第40頁的『MACRO\_FUNCTION 區塊』。

### **訊息區塊**

MESSAGE 區塊。為一組回覆碼、相關訊息以及傳回函數呼叫時 Net.Data 執行的動作。有關語法及範例，請參閱第44頁的『MESSAGE 區塊』。

### **字串**

字串中可使用任何英文字母、數字與標點符號。若是條件列示之詞彙內的字串，則可以是包括換行字元的任何字元。若是可執行區塊碼內的字串，則可以是包括換行字元的任何字元。

### **變數參照**

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

### **WHILE 區塊**

WHILE 區塊。以條件式字串處理程序執行迴路。有關語法及範例，請參閱第55頁的『WHILE 區塊』。

### **條件**

使用比較運算子，比較兩個項目。如果下列兩個條件為真，則 IF 條件被作為數字比較：

- 條件運算子是下列其中一項：<,<=,>,>=,==,!=
- 兩個詞彙都是代表有效整數的字串，而所謂有效整數是數位字串，之前選用性地有加號 (+) 或減號 (-)，且沒有其它白色空間。

如果其中一個條件非真，則執行一般字串比較。

### **詞彙**

一個變數名稱、字串、變數參照或函數呼叫。

### **%ELIF**

此關鍵字可以啟動替代處理路徑，可以包含條件列示及大部份的 Net.Data 巨集陳述式。

### **%ENDIF**

此關鍵字可以關閉 %IF 區塊。

### **%ELSE**

此關鍵字可以在不符合其它條件列示時，執行相關陳述式。

## 上下文

在這些上下文中可找到 IF 區塊：

- Net.Data 巨集宣告部份中任何其它區塊之外
- HTML 區塊
- IF 區塊
- MACRO\_FUNCTION 區塊
- REPORT 區塊
- ROW 區塊
- WHILE 區塊

## 限制

當 IF 區塊位於巨集的宣告部份中任何其它區塊之外時，可以包含這些元素。

- 備註區塊
- DEFINE 區塊
- DEFINE 陳述式
- FUNCTION 區塊
- 函數呼叫
- HTML 區塊
- IF 區塊
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式
- MACRO\_FUNCTION 區塊
- MESSAGE 區塊
- 變數參照

當 IF 區塊位於 HTML 區塊、MACRO\_FUNCTION 區塊、REPORT 區塊或 WHILE 區塊中時，可以包含這些元素。

- 備註區塊
- EXEC\_SQL 陳述式
- 函數呼叫
- IF 區塊
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式
- HTML 陳述式
- 字串
- 變數參照
- WHILE 區塊

您最多可以用巢狀方式設定 1024 個 IF 區塊。

## 範例

### 範例 1：Net.Data 巨集宣告部份中的 IF 區塊

```
%DEFINE a = "1"
%DEFINE b = "2"
...
%IF ($(DTW_HTML_TABLE) == "YES")
%define OUT_FORMAT = "HTML"
%ELSE
%define OUT_FORMAT = "CHARACTER"
%ENDIF

%HTML(REPORT){
    ...
%}
```

### 範例 2：HTML 區塊內的 IF 區塊

```
%HTML(REPORT){
@myFunctionCall()
%IF ($RETURN_CODE) == $(failure_rc))
    <P> 函數呼叫失敗，其回覆碼為 $(RETURN_CODE)。
%ELIF ($RETURN_CODE) == $(warning_rc))
    <P> 函數呼叫繼續，其警告碼為 $(RETURN_CODE)。
%ELIF ($RETURN_CODE) == $(success_rc))
    <P>函數呼叫順利完成。
%ELSE
    <P>函數呼叫傳回不明回覆碼 $(RETURN_CODE)。
%ENDIF
%}
```

### 範例 3：數字比較

```
%IF (ROW_NUM < "100")
    <p>表格尚未填滿...
%ELIF (ROW_NUM == "100")
    <p>表格已滿...
%ELSE
    <p>表格溢位...
%ENDIF
```

數字比較之所以能完成是由於隱含的表格變數 ROW\_NUM 總是傳回整數值，且這個被比較的值也是整數。

### 範例 4：巢狀 IF 區塊

```
%IF (MONTH == "January")
    %IF (DATE = "1")
        新年快樂！
    %ELSE
        哈，又是另一天。
    %ENDIF
%ENDIF
```

## INCLUDE 陳述式

### 目的

讀取且納入檔案到指定陳述式的 Net.Data 巨集中。

Net.Data 在起始設定檔 INCLUDE\_PATH 陳述式指定的目錄中，尋找併入檔。

您併入檔的使用方式，可以與大部份的高階語言相同。您可以插入共用標題及註腳、定義共用變數設定、或將 FUNCTION 區塊定義的共用次常式程式庫納入 Net.Data 巨集中。

在處理巨集時，Net.Data 只執行一次 INCLUDE 陳述式，並在巨集中 INCLUDE 陳述式的位置插入參考檔案的內容。分析參考檔案名稱的任何變數參照的時刻最先是在執行 INCLUDE 陳述式時，而不是在執行參考檔案內容時。

當 INCLUDE 陳述式位於 ROW 或 WHILE 區塊中時，Net.Data 不會重複執行 INCLUDE 陳述式。Net.Data 第一次執行 ROW 或 WHILE 區塊時會執行 INCLUDE 陳述式，納入參考檔案內容到區塊，然後用參考檔案內容重複執行 ROW 或 WHILE 區塊。

**授權要訣：**確定 Net.Data 執行時所用的使用者 ID，有權存取 INCLUDE 陳述式所參照的任何檔案。有關設定 Web 伺服器使用 Net.Data 檔案之存取權的詳細資料，請參閱 *Net.Data 管理及程式設計指南* 中的架構章節部分。

**要訣：**如果您要從一個區域 Web 伺服器併入 HTML 檔案，可使用如範例 3 所顯示的 INCLUDE\_URL 結構。只要使用示範的語法，您就不須更新 Net.Data 起始設定檔案中的 INCLUDE\_PATH，去設定 Web 伺服器已經知道的目錄。

### 語法



### 值

#### %INCLUDE

此關鍵字用於顯示被讀取且納入 Net.Data 巨集中的檔案。

#### 名稱

一個以英文字母或底線開頭的英文或數字字串，可以是英文字母、數字或底線字元的任意組合。

#### 字串

字串中可使用任何英文字母、數字與標點符號，但不能使用換行字元。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。



## 上下文

在這些上下文中可找到 INCLUDE 陳述式：

- DEFINE 區塊
- HTML 區塊
- REPORT 區塊
- ROW 區塊
- IF 區塊
- MESSAGE 區塊
- MACRO\_FUNCTION 區塊
- WHILE 區塊
- Net.Data 巨集宣告部份的區塊之外

## 限制

INCLUDE 陳述式可以包含這些元素：

- 備註區塊
- 字串
- 變數參照

不容許字串中的函數呼叫。

您最多可以用巢狀方式設定 10 個 INCLUDE 陳述式。

## 範例

**範例 1：**HTML 區塊中的 INCLUDE 陳述式

```
%HTML(start){  
%INCLUDE "header.hti"  
...  
%}
```

**範例 2：**REPORT 區塊中的 INCLUDE 陳述式

```
%REPORT {  
  %INCLUDE "report_header.txt"  
  %ROW {  
    %INCLUDE "row_include.txt"  
  }  
  %INCLUDE "report_footer.txt"  
%}
```

**範例 3：**INCLUDE 陳述式中的變數參照

```
%define library = "/qsys.lib/mylib.lib/"  
%define filename = "macros.file/incfile.mbr"  
  
%include "$(library)$(filename)"
```

## INCLUDE\_URL 陳述式

### 目的

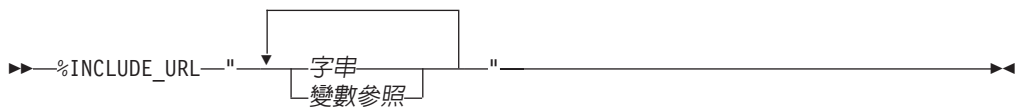
讀取並納入另一個檔案到指定此陳述式的 Net.Data 輸出結果中。指定的檔案，可能在區域或遠端伺服器中。

使用 INCLUDE\_URL 陳述式，您可以從一個巨集呼叫另一個巨集，而不須應用程式使用者選取「提出」按鈕。

在處理巨集時，Net.Data 只執行一次 INCLUDE\_URL 陳述式，並在巨集中 INCLUDE\_URL 陳述式的位置插入參考檔案的內容。分析參考檔案名稱的任何變數參照的時刻最先是在執行 INCLUDE\_URL 陳述式時，而不是在執行參考檔案內容時。

當 INCLUDE\_URL 陳述式位於 ROW 或 WHILE 區塊中時，Net.Data 不會重複執行 INCLUDE\_URL 陳述式。Net.Data 第一次執行 ROW 或 WHILE 區塊時會執行 INCLUDE\_URL 陳述式，納入參考檔案內容到區塊，然後用參考檔案內容重複執行 ROW 或 WHILE 區塊。

### 語法



### 值

#### %INCLUDE\_URL

此關鍵字用於顯示將從區域或遠端伺服器讀取且納入一個檔案 Net.Data 巨集中。

#### 字串

字串中可使用任何英文字母、數字與標點符號，但不能使用換行字元。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

### 上下文

在這些上下文中可找到 INCLUDE\_URL 陳述式：

- DEFINE 區塊
- HTML 區塊
- IF 區塊
- MACRO\_FUNCTION 區塊
- MESSAGE 區塊
- REPORT 區塊
- ROW 區塊
- WHILE 區塊
- Net.Data 巨集宣告部份的區塊之外

## 限制

INCLUDE\_URL 陳述式可以包含這些元素：

- 備註區塊
- 字串
- 變數參照

在 OS/390，INCLUDE\_URL 檔案最大是 256 KB。其它作業系統則無此限制。

使用 INCLUDE\_URL 陳述式時，勿遞迴呼叫現行巨集檔，以免起始一連串無限的巨集要求。

INCLUDE\_URL 沒有在 OS/400 環境中受支援。

## 範例

**範例 1：**從另一個伺服器併入 HTML 檔案

```
%include_url "http://www.ibm.com/path/myfile.html"
```

**範例 2：**藉由呼叫遠端伺服器名稱從該伺服器併入 HTML 檔案

```
%include_url "myserver/path/myfile.html"
```

其中 myserver 即為伺服器名稱。

**範例 3：**從區域 Web 伺服器併入 HTML 檔案

```
%include_url "/path/myfile.html"
```

**要訣：**只要使用這個方法，您就不須更新 Net.Data 架構檔中的 INCLUDE\_URL 路徑，去設定 Web 伺服器已經知道的目錄。如果字串不是以斜線開始，Net.Data 會假設該字串是一個伺服器名稱，並嘗試用對應的名稱從伺服器取回檔案。

**範例 4：**從遠端伺服器併入其它 Net.Data 巨集

```
%REPORT{  
<P>現行新選擇為 @DTW_rTIME() :  
%include_url "http://www.ibm.com/cgi-bin/db2www/hotpic.mac/report?custno=$(custno)"
```

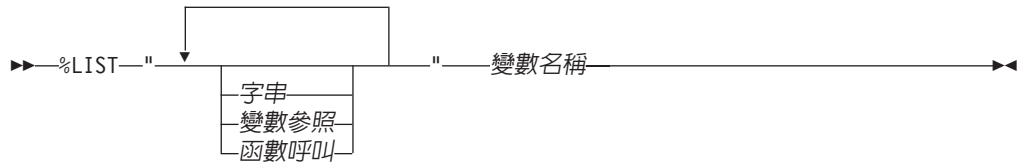
在這個範例中，呼叫的是巨集 hotpic.mac，而 custno 當作變數傳送。如果字串是以斜線開始，則 Net.Data 將從區域 Web 伺服器取回 INCLUDE 檔案。

## LIST 陳述式

### 目的

建置一個定界限的值列示。當您用多重項目建立結構 SQL 查詢時，可以使用 LIST 陳述式，就像某些 WHERE 或 HAVING 子句一般。

### 語法



### 值

#### %LIST

此關鍵字指定使用變數來建置以分隔符號區隔的值的列示。

#### 字串

字串中可使用任何英文字母、數字與標點符號，但不能使用換行字元。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

#### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

#### 變數名稱

定義變數的名稱。有關語法的詳細資訊，請參閱第4頁的『變數名稱』。

### 上下文

在這些上下文中可找到 LIST 陳述式：

- DEFINE 陳述式

### 限制

LIST 陳述式可以包含這些元素：

- 備註區塊
- 變數參照
- 函數呼叫
- 字串

### 範例

範例 1：變數列示

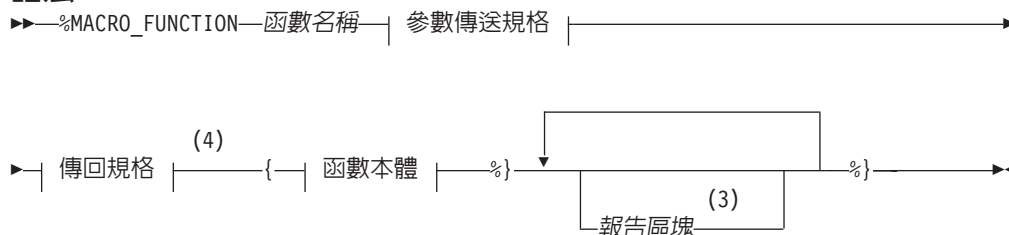
```
%DEFINE{  
DATABASE="custcity"  
%LIST " OR " conditions  
conditions="cond1='Sao Paolo'"  
conditions="cond2='Seattle'"  
conditions="cond3='Shanghai'"  
whereClause=conditions ? "WHERE $(conditions)" : ""  
%}
```

## MACRO\_FUNCTION 區塊

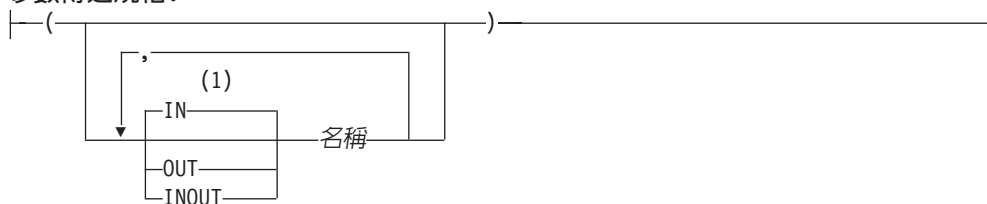
### 目的

定義一個可以從 Net.Data 巨集呼叫的次常式。在 MACRO\_FUNCTION 區塊中的可執行的陳述式必須是 Net.Data 巨集語言來源陳述式。

### 語法



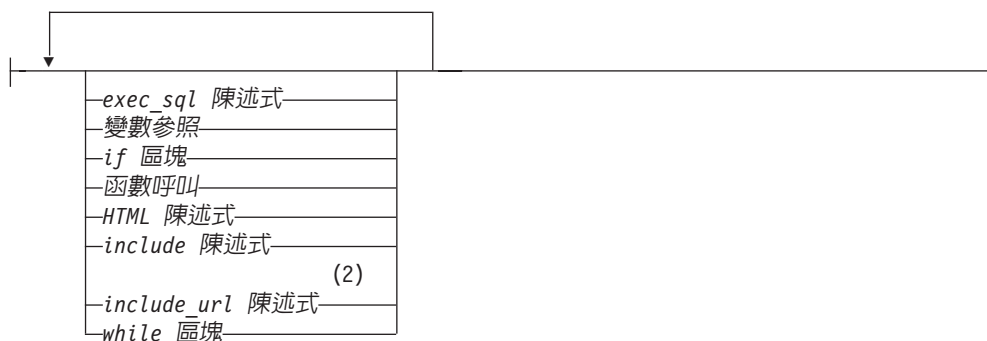
### 參數傳送規格：



### 傳回規格：



### 函數本體：



### 註：

- 1 在參數列示開頭沒有指定的參數類型時，引用預設值參數 IN 類型。無參數類型的參數使用最新指定在參數列示上的類型，或若無指定類型時使用 IN 類型。例如，在參數列示 `-( -parm1 - , INOUT -parm2 - , -parm3 - , OUT -parm4 - , -parm5 - )` 中，參數 `-parm1`、`-parm3` 及 `-parm5` 沒有參數型類。因為沒有指

定最初的參數類型，所以參數 *-parm1* 是 IN 類型。參數 *-parm3* 是 INOUT 類型，因為它是最新指定的參數類型。同樣地，參數 *-parm5* 有 OUT 的類型，因為它是最近在參數列示中指定的參數類型。

- 2 OS/400 不支援 INCLUDE\_URL 陳述式。
- 3 OS/400、OS/2、Windows NT 及 UNIX 支援 MACRO\_FUNCTION 區塊中的 REPORT 區塊。
- 4 RETURNS 陳述式僅在 OS/400 上才有支援。

## 值

### %MACRO\_FUNCTION

一種關鍵字，指定一個可以從 Net.Data 巨集呼叫的次常式。在 MACRO\_FUNCTION 區塊中的可執行的陳述式必須包含 Net.Data 可以直接解譯的語言陳述式。

### 函數名稱

定義的函數名稱。一個以英文字母或底線開頭的英文或數字字串，可以是英文字母、數字或底線字元的任意組合。

### 參數傳送規格：

**IN** 指定 Net.Data 傳送輸入資料給語言環境。IN 是預設值。

### OUT

指定語言環境傳回輸出資料給 Net.Data。

### INOUT

指定 Net.Data 傳送輸入資料給語言環境，及語言環境傳回輸出資料給 Net.Data。

### 名稱

一個以英文字母或底線開頭的英文或數字字串，可以是英文字母、數字或底線字元的任意組合。*name* 可以代表 Net.Data 表格或結果設定。

### 傳回規格：

### RETURNS

函數完成後宣告含有函數值的變數。

### 函數本體：

### exec\_sql

這是為了提供相容性才支援的 DB2WWW 版次 1 語言元素。請參閱第291頁的『附錄B. DB2 WWW 連線』或「DB2 全球資訊網版次 1」文件。

### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

### IF 區塊

此為 IF 區塊。執行條件字串處理。條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有單一加號 (+) 或減號 (-) 置於前端。

## 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

## HTML 陳述式

包含任何英數字元，以及從屬站瀏覽器的格式化 HTML 標籤。

## include 陳述式

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。

## include\_url 陳述式

INCLUDE\_URL 陳述式。讀取另一個檔案，納入指定此陳述式的 Net.Data 巨集中。指定的檔案，可能在區域或遠端伺服器中。有關語法及範例，請參閱第36頁的『INCLUDE\_URL 陳述式』。

## WHILE 區塊

WHILE 區塊。以條件式字串處理程序執行迴路。有關語法及範例，請參閱第55頁的『WHILE 區塊』。

## 報告區塊

REPORT 區塊。函數呼叫的輸出的格式化指示。您可以在報告中使用表頭及註腳資訊。有關語法及範例，請參閱第48頁的『REPORT 區塊』。

## 上下文

在這些上下文中可找到 MACRO\_FUNCTION 區塊：

- IF 區塊
- Net.Data 巨集宣告部份的區塊之外

## 限制

MACRO\_FUNCTION 區塊可以包含這些元素：

- 備註區塊
- EXEC\_SQL 陳述式
- HTML 陳述式
- IF 區塊
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式
- OS/400 上不被支援
- REPORT 區塊
- WHILE 區塊
- 變數參照
- 函數呼叫

## 範例

範例 1：指定訊息處理的巨集函數



```
%MACRO_FUNCTION setMessage(IN rc, OUT message) {
%IF (rc == "0")
    @dtw_assign(message, "函數呼叫順利完成。")
%ELIF (rc == "-1")
    @dtw_assign(message, "函數失敗，記憶體不足。")
%ELIF (rc == "-2")
    @dtw_assign(message, "函數失敗，參數無效。")
%ENDIF
%}
```

## 範例 2：指定表頭資訊的巨集函數

```
%MACRO_FUNCTION setup(IN browserType) {
%{ 在巨集中每一個 HTML 區塊頂端呼叫這個函數 %}
%INCLUDE "header_info.html"
@dtw_rdate()
%IF (browserType == "IBM")
    @setupIBM()
%ELIF (browserType == "MS")
    @setupMS()
%ELIF (browserType == "NS")
    @setupNS()
%ELSE
    @setupDefault()
%ENDIF
%}
```

## MESSAGE 區塊

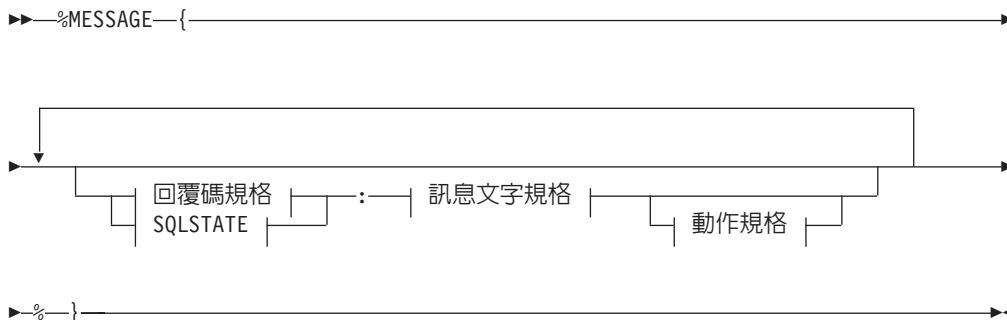
### 目的

根據函數回覆碼，指定顯示的訊息及採行的建議動作。

在 MESSAGE 區塊中定義一組回覆碼、對應訊息、以及建議動作。函數呼叫完成後，*Net.Data* 比較它的回覆碼與 MESSAGE 區塊中定義的回覆碼。如果函數的回覆碼符合 MESSAGE 區塊中定義的回覆碼，*Net.Data* 立即顯示訊息，並評估動作，判斷應該繼續處理或跳出 *Net.Data* 巨集。

MESSAGE 區塊可以是整體範圍，也可以是區域單一 FUNCTION 區塊。巨集最外層次定義的 MESSAGE 巨集，作為整體範圍。如果定義多個整體 MESSAGE 區塊，僅最後一個區塊有效。如果 MESSAGE 區塊定義在 FUNCTION 區域中，則該區域在所屬 FUNCTION 區塊的區域範圍中。有關回覆碼處理程序規則，請參閱 *Net.Data* 管理程式設計指南 中的 MESSAGE 部分。

### 語法



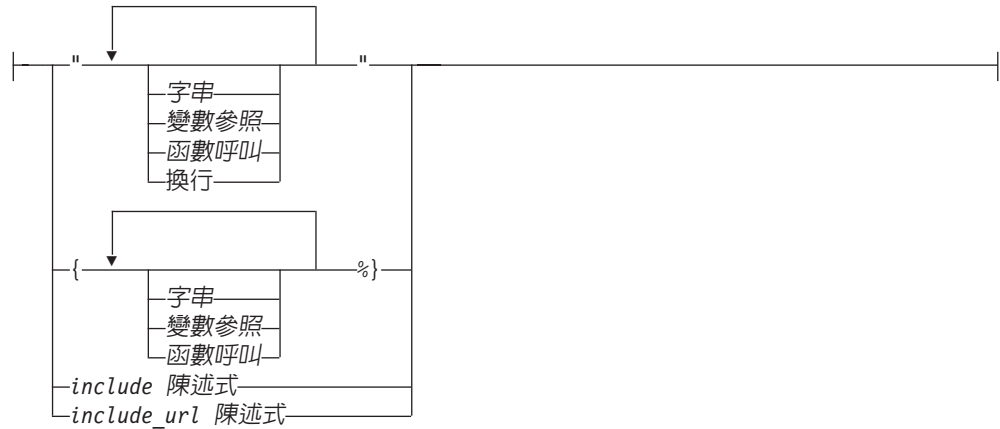
#### 回覆碼規格:



#### SQLSTATE:



#### 訊息文字規格:



#### 動作規格:



## 值

### %MESSAGE

此區塊關鍵字定義一組回覆碼、相關訊息、以及傳回函數呼叫時 Net.Data 執行的動作。

#### 回覆碼規格

一個正整數或負整數。Net.Data RETURN\_CODE 變數值符合回覆碼規格值時，訊息陳述式的剩餘資訊可以用來處理函數呼叫。您也可以為未特別在 MESSAGE 區塊中輸入的回覆碼，指定訊息。

#### +DEFAULT

此關鍵字可用來指定預設正訊息碼。若 RETURN\_CODE 大於 0，並且未指定完全相符時，則 Net.Data 使用此訊息陳述式中的資訊將用來處理此函數呼叫。

#### -DEFAULT

此關鍵字可用來指定預設負訊息碼。若 RETURN\_CODE 小於 0，並且未指定完全相符時，則 Net.Data 使用此訊息陳述式中的資訊將用來處理此函數呼叫。

#### DEFAULT

此關鍵字可用來設定預設訊息碼。若下列條件皆符合時，Net.Data 使用此訊息陳述式中的資訊將用來處理此函數呼叫：

- 若 RETURN\_CODE 大於或小於零，且不為零
- 若沒有完全相符指定的回覆碼時
- 若 RETURN\_CODE 大於或小於零，而 +DEFAULT 或 -DEFAULT 沒有指定的回覆碼時

#### 訊息碼

此訊息碼指定處理期間可能發生的錯誤及警告。為一串從 0 到 9 的數字。

## SQLSTATE

此關鍵字提供應用程式一般錯誤條件的一般碼。SQLSTATE 值是根據併入在 SQL 標準中的 SQLSTATE 規格建立，而撰寫程式碼方式在所有 IBM 上所施行的 SQL 都一樣。

### state\_id

指 SQLSTATE。一個 5 字元 (位元組) 的英數字串，其格式為 *ccsss*，其中 *cc* 表示類別，而 *sss* 表示次類別。

## 訊息文字規格

當 RETURN\_CODE 符合現行訊息陳述式中的回覆碼值時，被傳送給 Web 瀏覽器的字串。

### 字串

字串中可使用任何英文字母、數字與標點符號。如果字串出現在雙引號內，則不容許使用換行字元。

### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

## 動作規格

決定當 RETURN\_CODE 符合現行訊息陳述式中的回覆碼值時，Net.Data 採行的動作。

### EXIT

此關鍵字指定當發生訊息碼對應的錯誤或警告時，立即跳出巨集。這是預設值。

### CONTINUE

此關鍵字指定當發生訊息碼對應的錯誤或警告時，繼續處理。

### include 陳述式

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。INCLUDE 陳述式可以出現在 MESSAGE 的任何地方。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。

### include\_url 陳述式

INCLUDE\_URL 陳述式。讀取另一個檔案，納入指定此陳述式的 Net.Data 巨集中。指定的檔案，可能在區域或遠端伺服器中。有關語法及範例，請參閱第36頁的『INCLUDE\_URL 陳述式』。

## 上下文

在這些上下文中可找到 MESSAGE 區塊：

- FUNCTION 區塊
- IF 區塊
- Net.Data 巨集宣告部份所有區塊或陳述式之外

## 限制

MESSAGE 區塊可以包含這些元素：

- 備註區塊
- 函數呼叫
- 變數參照
- HTML 陳述式
- 字串
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式

對於 **OS/390**、**OS/2**、**Windows NT** 及 **UNIX** 作業系統而言：無法從 SQL 函數內呼叫 SQL 函數。

## 範例

### 範例 1：區域 MESSAGE 區塊

```
%{ FUNCTION 區塊中的區域訊息區塊 %}  
%FUNCTION(DTW_REXX) my_function() {  
  %EXEC { my_command.cmd %}  
  %MESSAGE{  
    -601: {<H3>已建立表格，請返回並輸入您的名稱。</H3>  
    <P><a href="input">傳回</a>  
  %}  
  default: "<H3>無法繼續，因為錯誤 $(RETURN_CODE)</H3>"%} : exit  
  %}
```

### 範例2：整體 MESSAGE 區塊

```
%{ 整體訊息區塊 %}  
  %MESSAGE {  
    -100 : "回覆碼 -100 的訊息" : exit  
    100 : "回覆碼 100 訊息" : continue  
    +default : {  
      此為超出一行的長訊息。  
      您可以在此訊息中使用包括鏈結和套表的  
      HTML 標籤。 %} : continue  
    %}  
  %}  
  
%{ FUNCTION 區塊中的區域訊息區塊 %}  
%FUNCTION(DTW_REXX) my_function() {  
  %EXEC { my_command.cmd %}  
  %MESSAGE {  
    -100 : "回覆碼 -100 的訊息" : exit  
    100 : "回覆碼 100 訊息" : continue  
    -default : {  
      此為超出一行的長訊息。  
      您可以在此訊息中使用包括鏈結和套表的  
      HTML 標籤。 %} : exit  
    %}  
  %}
```

### 範例 3：包含 INCLUDE 陳述式的 MESSAGE 區塊

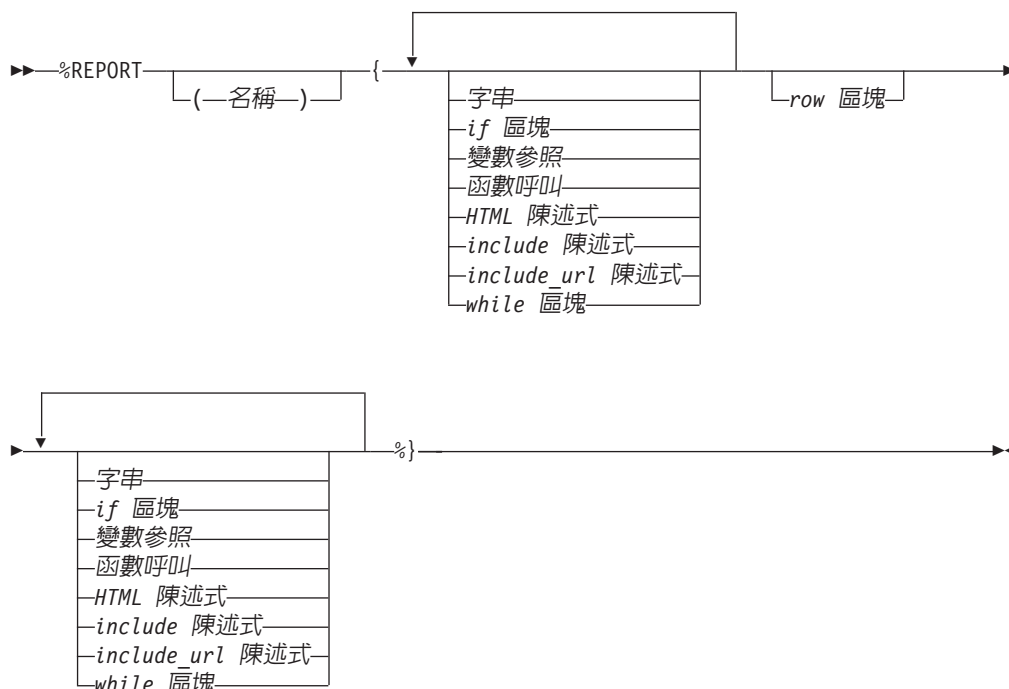
```
%message {  
  %include "rc1000.msg"  
  %include "rc2000.msg"  
  %include "defaults.msg"  
  %}
```

## REPORT 區塊

### 目的

製作函數呼叫之輸出的格式。您可以輸入一個表格名稱參數，來設定這個表格使用被命名表格內的資料。否則，所產生的表格會使用函數參數列示中第一個輸出表格上的資料，或是當參數列示上沒有表格名稱時，使用預設的表格資料。

### 語法



### 值

#### %REPORT

此關鍵字指定製作函數呼叫輸出指示的格式。您可以在報告中使用表頭及註腳資訊。

#### 名稱

此值代表 Net.Data 表格或結果設定。有關詳細資訊，請參閱 *Net.Data* 管理 & 程式設計手冊中的「報表區塊」章節。

#### 字串

字串中可使用任何英文字母、數字與標點符號。

#### IF 區塊

此為 IF 區塊。執行條件字串處理。條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有一個加號 (+) 或減號 (-) 置於前端。關於語法及範例，請參閱第28頁的『IF 區塊』。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

## 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或一個具有指定引數的 Net.Data 內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

## HTML 陳述式

包含任何英數字元，以及從屬站瀏覽器的格式化 HTML 標籤。

## include 陳述式

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。

## include\_url 陳述式

INCLUDE\_URL 陳述式。讀取另一個檔案，納入指定此陳述式的 Net.Data 巨集中。指定的檔案，可能在區域或遠端伺服器中。有關語法及範例，請參閱第36頁的『INCLUDE\_URL 陳述式』。

## row 區塊

ROW 區塊。顯示一次函數呼叫傳回的資料各列的 HTML 格式化資料。有關語法及範例，請參閱第51頁的『ROW 區塊』。

## WHILE 區塊

WHILE 區塊。以條件式字串處理程序執行迴路。有關語法及範例，請參閱第55頁的『WHILE 區塊』。

## 上下文

在這些上下文中可找到 REPORT 區塊：

- FUNCTION 陳述式或區塊

## 限制

報告區塊可以包含這些元素：

- 備註區塊
- IF 區塊
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式
- ROW 區塊
- WHILE 區塊
- 函數呼叫

對於 OS/390、OS/2、Windows NT 及 UNIX 作業系統而言：無法從 SQL 函數內呼叫 SQL 函數。

- HTML 陳述式
- 字串
- 變數參照

若是 OS/390：MACRO\_FUNCTION 區塊中不容許有 REPORT 區塊。

## 範例

範例 1：顯示一列姓名及位置的雙欄 HTML 表格

```

%FUNCTION(DTW_SQL) mytable() {
%REPORT{
<H2>查詢結果</H2>
<P>選取名稱以顯示明細。
<TABLE BORDER=1>
<TR><TD>姓名</TD><TD>位置</TD>
%ROW{
  <TR>
  <TD>
  <a href="/cgi-bin/db2www/name.mac/details?name=$(V1)&location=$(V2)">$(V1)</a></TD>
  <TD>$(V2)</TD>
  %}
  </TABLE>
%}

```

選取表格中的一個名稱，可呼叫 *name.mac* Net.Data 巨集的明細 HTML 區塊，將這二個值當作 URL 的一部份來傳送給它。您可以在本例的 *name.mac* 中使用這些值來尋找名稱的其它明細。

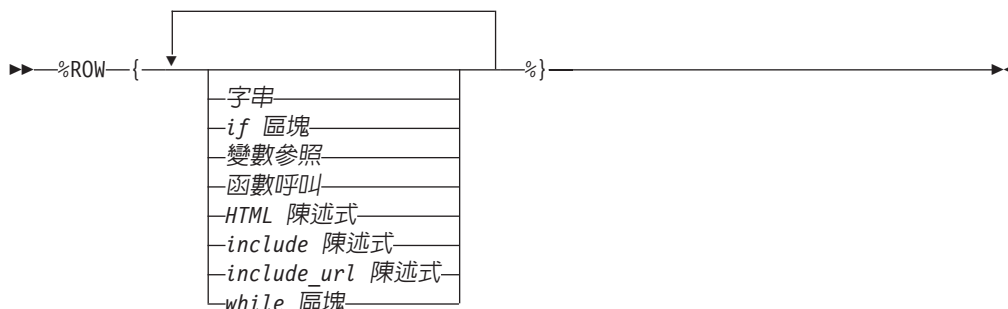


## ROW 區塊

### 目的

處理函數呼叫傳回的每一個表格列。 Net.Data 只為各列處理一個 ROW 區塊中的陳述式。

### 語法



### 值

#### %ROW

此關鍵字指定函數呼叫傳回的資料各列的 HTML 格式化資料，只顯示一次。

#### 字串

字串中可使用任何英文字母、數字與標點符號。

#### IF 區塊

此為 IF 區塊。執行條件字串處理。條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有單一加號 (+) 或減號 (-) 置於前端。關於語法及範例，請參閱第28頁的『IF 區塊』。

#### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

#### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或具有指定引數的內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

#### HTML 陳述式

包含任何英數字元，以及從屬站瀏覽器的格式化 HTML 標籤。

#### include 陳述式

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。

#### include\_url 陳述式

INCLUDE\_URL 陳述式。讀取另一個檔案，納入指定此陳述式的 Net.Data 巨集中。指定的檔案，可能在區域或遠端伺服器中。有關語法及範例，請參閱第36頁的『INCLUDE\_URL 陳述式』。

## WHILE 區塊

WHILE 區塊。以條件式字串處理程序執行迴路。有關語法及範例，請參閱第55頁的『WHILE 區塊』。

## 上下文

在這些上下文中可找到 ROW 區塊：

- REPORT 區塊

## 限制

ROW 區塊可以包含這些元素：

- 備註區塊
- IF 區塊
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式
- WHILE 區塊
- 函數呼叫

對於 OS/390、OS/2、Windows NT 及 UNIX 作業系統而言：無法從 SQL 函數內呼叫 SQL 函數。

- 變數參照
- HTML 陳述式
- 字串

## 範例

範例 1：顯示一列姓名及位置的雙欄 HTML 表格

```
%REPORT{
<H2>查詢結果</H2>
<P>選取名稱以顯示明細。
<TABLE BORDER=1>
<TR><TD>姓名</TD><TD>位置</TD>

%ROW{
  <TR>
  <TD>
<a href="/cgi-bin/db2www/name.mac/details?name=$(V1)&location=$(V2)">$(V1)</a></TD>
<TD>$(V2)</TD>
%}

  </TABLE>
%}
```

選取表格中的一個名稱，可呼叫 *name.mac* Net.Data 巨集的明細 HTML 區塊，將這二個值當作 URL 的一部份來傳送給它。您可以在本例的 *name.mac* 中使用這些值來尋找名稱的其它明細。

## TABLE 陳述式

### 目的

定義一套相關資料的變數。變數包含一組橫列與直欄，其中有一列直欄表頭，說明每一列的欄位。表格陳述式只能用於 DEFINE 陳述式或區塊。

當 TABLE 變數被參考到時，如果 DTW\_HTML\_TABLE 變數設定為 YES，Net.Data 會將表格內容顯示為一般字元表格或 HTML 表格。

### 語法

►►%TABLE—| 上限 |—————►►

### 上限：

|  
| ( — 號碼 — )  
| — ALL —

### 值

#### %TABLE

此關鍵字指定相關資料的集合定義，裡面有一個相同記錄或橫列的陣列、以及一個介紹各列欄位的直欄名稱陣列。

#### 上限

可以併入表格的列數。如果未指定上限值，則表格可含有無限制的列數。

#### 號碼

為一串從 0 到 9 的數字。「零」可以作為表格不受限制的列數。

#### ALL

此關鍵字容許表格列數不受限制。

### 上下文

在這些上下文中可找到 TABLE 陳述式：

- DEFINE 陳述式

### 限制

TABLE 陳述式可以包含這些元素：

- 備註區塊
- 號碼

### 範例

**範例 1：** 列數上限為 30 的 Net.Data 表格

```
%DEFINE myTable1=%TABLE(30)
```

**範例 2：** 使用所有列數預設值的 Net.Data

```
%DEFINE myTable2=%TABLE
```

**範例 3：** 指定所有列數的 Net.Data 表格

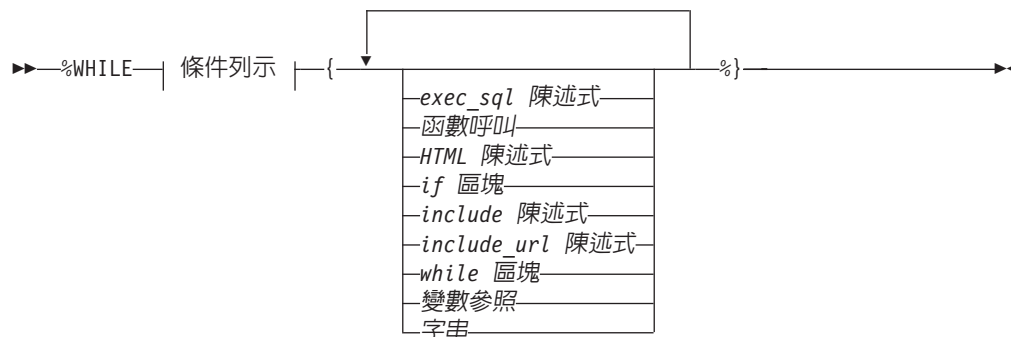
```
%DEFINE myTable3=%TABLE(ALL)
```

## WHILE 區塊

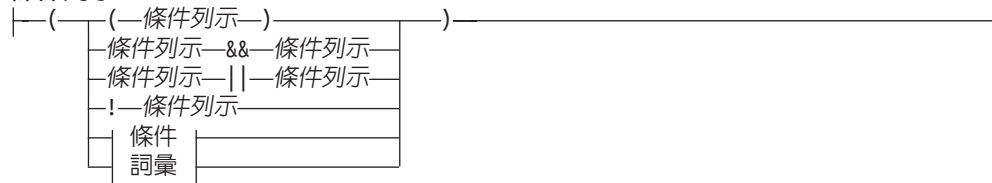
## 目的

提供根據條件式字串處理程序的迴路結構。您可以在 **HTML** 區塊、報告區塊、**ROW** 區塊、**IF** 區塊及 **MACRO\_FUNCTION** 區塊中使用 **WHILE** 區塊。條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有單一加號 (+) 或減號 (-) 置於前端。

## 語法



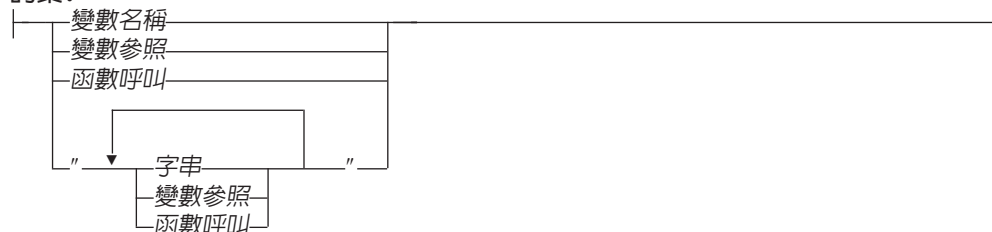
條件列示：



條件：



詞彙：



## 值

### %WHILE

此關鍵字指定迴路處理程序。

### 條件列示

比較條件及詞彙的值。使用布林運算子，可以連接條件列示。您可以在另一個條件列示中建立巢狀條件列示。

### 條件

使用比較運算子，比較兩個項目。如果下列兩個條件為真，則 IF 條件被作為數字比較：

- 條件運算子是下列其中一項：<,<=,>,>=,==,!=
- 兩個詞彙都是代表有效整數的字串，而所謂有效整數是數位字串，之前選用性地有加號 (+) 或減號 (-)，且沒有其它白色空間。

如果其中一個條件非真，則執行一般字串比較。

### 詞彙

一個函數呼叫的變數名稱、字串及變數參照。

### EXEC\_SQL 陳述式

這是為了提供相容性才支援的 DB2WWW 版次 1 語言元素。請參閱第291頁的『附錄B. DB2 WWW 連線』或「DB2 全球資訊網版次 1」文件。

### 函數呼叫

呼叫一或多個 FUNCTION 或 MACRO\_FUNCTION 區塊，或具有指定引數的內建函數。有關語法及範例，請參閱第23頁的『函數呼叫 (@)』。

### HTML 陳述式

包含任何英數字元，以及從屬站瀏覽器的格式化 HTML 標籤。

### IF 區塊

此為 IF 區塊。執行條件字串處理。條件列示中的字串值，如果是代表整數且前端或尾端沒有白色空間時，將被作為數字來比較。它們可以有一個加號 (+) 或減號 (-) 置於前端。關於語法及範例，請參閱第28頁的『IF 區塊』。

### include 陳述式

INCLUDE 陳述式。讀取及納入檔案到 Net.Data 巨集。有關語法及範例，請參閱第34頁的『INCLUDE 陳述式』。

### include\_url 陳述式

INCLUDE\_URL 陳述式。讀取另一個檔案，併入指定此陳述式的 Net.Data Web 巨集中。指定的檔案，可能在區域或遠端伺服器中。有關語法及範例，請參閱第36頁的『INCLUDE\_URL 陳述式』。

### WHILE 區塊

WHILE 區塊。以條件式字串處理程序執行迴路。有關語法及範例，請參閱第55頁的『WHILE 區塊』。

### 變數參照

傳回變數值，並以 \$ 及 () 符號指定。例如：如果 VAR='abc'，則 \$(VAR) 傳回值 'abc'。有關語法的詳細資訊，請參閱第4頁的『變數參照』。

## 字串

字串中可使用任何英文字母、數字與標點符號。條件列示之詞彙內的字串，可以包括換行字元的任何字元。

## 變數名稱

定義變數的名稱。有關語法的詳細資訊，請參閱第4頁的『變數名稱』。

## 上下文

在這些上下文中可找到 **WHILE** 區塊：

- HTML 區塊
- REPORT 區塊
- ROW 區塊
- MACRO\_FUNCTION 區塊
- IF 區塊
- WHILE 區塊

## 限制

**WHILE** 區塊可以包含這些元素：

- 備註區塊
- EXEC\_SQL 陳述式
- IF 區塊
- WHILE 區塊
- 字串
- HTML 陳述式
- 函數呼叫
- 變數參照
- INCLUDE 陳述式
- INCLUDE\_URL 陳述式

## 範例

**範例 1：**產生表格列數的 **WHILE** 區塊

```
%DEFINE loopCounter = "1"

%HTML(build_table) {
%WHILE (loopCounter <= "100") {
  %{ 產生表格標籤及欄位標題 %}
  %IF (loopCounter == "1")
    <TABLE BORDER>
    <TR>
      <TH>Item #
      <TH>Description
    </TR>
  %ENDIF

  %{ 產生個別的列 %}
  <TR>
    <TD>
      <TD>$(loopCounter)
```

```

<TD>@getDescription(loopCounter)
</TR>

%{ 產生終止表格標籤 %}
%IF (loopCounter == "100")
    </TABLE>
%ENDIF

%{ 增量迴路計數器 %}
@dtw_add(loopCounter, "1", loopCounter)
%}
%}

```



---

## 第2章 變數

Net.Data 提供兩種變數型類：使用者定義變數及 Net.Data 變數。

### 第60頁的『使用者定義的變數』

為您的應用程式而定義的變數。您可以定義執行下列作業的變數：

- 第60頁的『條件式變數』  
根據另一個變數或字串值而指定一變數值。
- 第61頁的『環境變數』  
使用 ENVVAR 語言結構去參考環境變數。
- 第61頁的『執行變數』  
使用 EXEC 語言結構，由變數參照的函數來呼叫其它程式。
- 第62頁的『隱藏變數』  
隱藏 HTML 原始程式中的變數參照。
- 第63頁的『列示變數』  
使用 LIST 語言結構建置一個以分隔符號區隔的值串。
- 第64頁的『表格變數』  
在函數之間往返傳送一列值。用於報告輸出。

### Net.Data 變數

用於雜項處理程序與檔案操作、表格處理、報告格式製作，及語言環境的變數。

您可以定義或修改部份變數的值，但其它變數則必需由 Net.Data 定義。變數的說明會指定是否可讓您定義值。關於如何定義值的詳細資訊，請參閱各變數的說明。

下列變數類型由 Net.Data 定義：

- 第65頁的『Net.Data 表格處理程序變數』  
由 Net.Data 定義，可供您處理 Net.Data 表格。使用這些變數來存取 SQL 查詢及函數呼叫的資料。除非另外指定，否則只有在 REPORT 或 區塊 ROW 區塊中才能辨識它們。
- 第75頁的『Net.Data 報告變數』  
可幫助您使用函數自行設定報告。您可以在任何 Net.Data 巨集區塊中，定義或參照報告變數。
- 第83頁的『Net.Data 語言環境變數』  
可幫助您使用語言環境自行設定 FUNCTION 區塊的處理方式。
- 第100頁的『Net.Data 雜項變數』  
由 Net.Data 定義，用來影響 Net.Data 處理程序、尋找函數呼叫的狀態、與取得關於資料庫查詢結果的資訊。部份雜項變數是由 Net.Data 設定且無法被變更。

有許多 Net.Data 變數的輸出，都是根據它所執行的作業系統而定。

在 Net.Data 巨集中，常數最高可達 256 KB。因此，您無法在巨集中起始設定一個長度大於 256 KB 的變數或將其設定為預設值。

在本章中，每一個變數所適用的作業系統都會被指定。下列定義的是作業系統的縮寫：

<b>HP-UX</b>	Hewlett Packard UNIX 作業系統
<b>SCO</b>	Santa Cruz Operation OpenServer
<b>SUN</b>	Solaris 作業系統
<b>Win NT</b>	Microsoft Windows NT 作業系統

---

## 使用者定義的變數

這個部分將說明使用者定義的變數。您可以在巨集中定義這些變數。

### 條件式變數

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

條件式變數的值，是根據另一變數或字串的值，做條件式設定。它又稱為 *ternary* 作業。

條件式變數的語法為：

`test ? trueValue : falseValue`

其中：

**test**      要測試的條件。

**trueValue**  
            測試為真時所採用的值。

**falseValue**  
            測試為偽時所採用的值。

**範例 1：**用兩個可能的值所定義的條件式變數

`varA = varB ? "value_1" : "value_2"`

若 `varB` 存在，則 `varA=value_1`，否則 `varA=value_2`。

**範例 2：**用一個變數參照所定義的條件式變數

`varname = ? "${value_1}"`

此例中，如果 `value_1` 為空值，則 `varname` 也為空值，否則 `varname` 設定為 `value_1`。

**範例 3：**與 `LIST` 陳述式及 `WHERE` 子句一起使用的條件式變數

```
%DEFINE{
%list " AND " where_list
where_list = ? "custid = ${cust_inp}"
where_list = ? "product_name LIKE '${(prod_inp)}%' "
where_clause = ? "WHERE ${where_list}"
```

```
%}

%FUNCTION(DTW_SQL) mySelect(){
    SELECT * FROM proddtable $(where_clause)
%}
```

條件式和 LIST 變數一起使用時，其效果最佳。這個範例將告訴您，如何在 DEFINE 區塊中設定 WHERE 子句。 *cust\_inp* 和 *prod\_inp* 這兩個變數是從 Web 瀏覽器傳來的 HTML 輸入變數，通常是從 HTML 套表傳來。*where\_list* 變數是一個 LIST 變數，它是由兩個條件式陳述式所組成，每一個陳述式都包含來自 Web 瀏覽器的變數。

如果 Web 瀏覽器傳回 *cust\_inp*及 *prod\_inp* 兩變數所用的值，例如 IBM 及 755C，則 *where\_clause* 為：

```
WHERE custid = IBM AND product_name LIKE '755C%'
```

如果 *cust\_inp* 或 *prod\_inp* 其中一個變數為空值，或者沒有定義，則 WHERE 子句會變成省略這個空值。例如：如果 *prod\_inp* 為空值，則 WHERE 子句將如下所示：

```
WHERE custid = IBM
```

如果兩者皆為空值或都沒有定義，則 *where\_clause* 變數為空值，且含有 *\$(where\_clause)* 的 SQL 查詢中，也不會出現任何 WHERE 子句。

## 環境變數

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

環境變數讓您使用 Net.Data ENVVAR 語言結構來參考存在於 Net.Data 在其下執行之處理程序的環境變數。

**範例 1：** 被指定一個環境變數值的變數

```
%define SERVER_NAME=%ENVVAR
```

```
...
```

```
伺服器是 $(SERVER_NAME)
```

環境變數 *SERVER\_NAME* 的值是現行伺服器名稱，在此例中，即為 *www.software.ibm.com*。

```
伺服器是 www.software.ibm.com
```

有關 ENVVAR 陳述式的詳細資訊，請參閱第13頁的『ENVVAR 陳述式』。

## 執行變數

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

透過執行變數特性，可讓您用變數參照呼叫其它程式。執行變數是透過 EXEC 語言元素，被定義在 Net.Data 巨集中。有關 EXEC 語言元素的詳細資訊，請參閱第14頁的『EXEC 區塊或陳述式』。

當 Net.Data 在巨集中發現一個執行變數時，它會使用下列方法尋找被參考到的可執行程式：

1. 它會搜尋 Net.Data 起始設定檔案中的 EXEC\_PATH。有關 EXEC\_PATH 的詳細資訊，請參閱 *Net.Data 管理程式設計指南* 中的架構章節。
2. 如果 Net.Data 找不到該程式，它會搜尋系統所定義的目錄。如果找到了可執行的程式，Net.Data 會執行該程式。

#### 範例 1：執行變數定義

```
%DEFINE runit=%exec "testProg"
```

變數 *runit* 被定義去執行可執行的程式 *testProg*；*runit* 變成了一個可執行的變數。

當 Net.Data 在 Net.Data 巨集中發現執行變數參照時，它會執行可執行的程式。例如，當 Net.Data 巨集中進行了一個對變數 *runit* 的執行變數參照時，程式 *testProg* 會被執行。

有一個簡單的方法是從另一個變數定義來參考一個執行變數。範例 2 示範這個方法。變數 *date* 被定義為一個執行變數，而 *dateRpt* 被定義為一個包含變數的執行變數參照。

#### 範例 2：作為變數參照的執行變數

```
%DEFINE date=%exec "date"  
%DEFINE dateRpt="Today is $(date)"
```

當 Net.Data 分析變數參照 *\$(dateRpt)* 時，Net.Data 會搜尋可執行的 *date*、執行該程式，並傳回下列結果：

```
Today is Tue 11-07-1995
```

執行變數永遠不會被設定其所呼叫之執行程式的輸出值。以上述範例來說，日期的值為空值。若您在 DTW\_ASSIGN 函數呼叫中使用它，以將它的值給指定另一個變數，則在做了這項指定之後，新變數的值也是為空值。執行變數的唯一目的是用來呼叫其所定義的程式。

您也可以藉由在變數定義中指定參數及程式名稱，然後將參數傳給要執行的程式。

#### 範例 3：使用參數的執行變數

```
%DEFINE mph=%exec "calcMPH $(distance) $(time)"
```

*distance* 及 *time* 的值被傳送到程式 *calcMPH*。

## 隱藏變數

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

有了隱藏變數，您可以參照變數，同時又在 HTML 原始程式中隱藏實際的變數值。若使用隱藏變數，請執行下列步驟：

1. 為每一個您要隱藏的字串，定義一個變數。
2. 在參照變數的 HTML 區塊中以雙貨幣符號取代單貨幣符號來參照變數。例如，採用 *\$(X)* 而不是 *\$(X)*。

請勿使用動態建構的變數名稱來參照隱藏變數。

### 範例 1：在 HTML 套表的隱藏變數

```
%HTML(INPUT){
<FORM ...>
<P>請選取要檢視的欄位：
<SELECT NAME="Field">
<OPTION VALUE="$(name)"> 名稱
<OPTION VALUE="$(addr)"> 位址
.
.
.
</FORM>
%}

%DEFINE{
name="customer.name"
addr="customer.address"
%}

%FUNCTION(DTW_SQL) mySelect(){
SELECT $(Field) FROM customer
%}
.
.
.
```

當 HTML 套表顯示在 Web 瀏覽器上時，`$(name)` 和 `$(addr)` 便會分別被置換成 `$(name)` 和 `$(addr)`，如此一來，實際的表格和直欄名稱就不會在 HTML 套表顯示出來，而沒有任何人可以察覺真正的變數名稱已被隱藏。當客戶提出該套表時，會呼叫 HTML(REPORT) 區塊。當 @mySelect() 呼叫 FUNCTION 區塊時，SQL 陳述式中的 `$(Field)` 會被換成在 SQL 查詢中的 `customer.name` 或 `customer.addr`。

## 列示變數

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

您可以使用 LIST 語言結構建置一個有分隔設定的值串。在您希望以多個項目建構一個 SQL 查詢時，(例如，常見於一些 WHERE 或 HAVING 子句中)，這類變數尤其有用。

空白是具有其意義的。您通常會在值的兩邊留空白。大部份的查詢會使用布林或算術運算子 (例如，AND、OR 或 >)。有關語法及範例，請參閱第38頁的『LIST 陳述式』。

### 範例 1：使用條件式、隱藏及列示變數

```
%HTML(INPUT){
<FORM METHOD="POST" ACTION="/cgi-bin/db2www/example2.max/report">
請選取一或多個都市：<BR>
<INPUT TYPE="checkbox" NAME="conditions" VALUE="$(cond1)">Sao Paulo<BR>
<INPUT TYPE="checkbox" NAME="conditions" VALUE="$(cond2)">Seattle<BR>
<INPUT TYPE="checkbox" NAME="conditions" VALUE="$(cond3)">Shanghai<BR>
<INPUT TYPE="submit" VALUE="Submit Query">
</FORM>
%}

%DEFINE{
DATABASE="custcity"
%LIST " OR " conditions
cond1="cond1='Sao Paolo'"
cond2="cond2='Seattle'"
cond3="cond3='Shanghai'"
whereClause= ? "WHERE $(conditions)" : ""
}
```

```
%}  
  
%FUNCTION(DTW_SQL) mySelect(){  
SELECT name, city FROM citylist  
$(whereClause)  
%}  
  
%HTML(REPORT){  
@mySelect()  
%}
```

在 HTML 套表中，如果沒有勾選選項框，則 *conditions* 為空值，所以 *whereClause* 在查詢中也是空值。否則，*whereClause* 會有選取的值，該值以布林運算子 OR 加以區隔。例如，若三個城市全選，則 SQL 查詢會是：

```
SELECT name, city FROM citylist  
WHERE cond1='Sao Paolo' OR cond2='Seattle' OR cond3='Shanghai'
```

**範例 2：值區隔符號**

```
%DEFINE %LIST " | " VLIST  
%REPORT{  
%ROW{  
<EM>$(ROW_NUM):</EM> $(VLIST)  
%}  
%}
```

在此例中，表格處理程序變數 *VLIST* 是使用兩個雙引號和一個 OR 條欄 ( | )，作為值區隔符號。字值串是由用引號括住的值加以分隔。

**表格變數**

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

表格變數會定義一組相關資料的集合。包含一組橫列與直欄（含一列直欄表頭）。您可以使用表格變數，將幾組值傳給函數。您可以在函數的 **REPORT** 區塊中參照表格 (列) 的個別元素，或使用表格內建函數。表格變數通常作為 SQL 函數的輸出及報告的輸入使用，但您也可以將它們當作 **IN**、**OUT** 或 **INOUT** 參數傳送給任何非 SQL 函數。但表格只能當作 **OUT** 參數，傳給 SQL 函數。有關語法及範例，請參閱第53頁的『**TABLE** 陳述式』。

參照 **TABLE** 變數時，*Net.Data* 將表格內容顯示成一般字元表格，或 *DTW\_HTML\_TABLE* 變數設為 **YES**，則顯示成 **HTML** 表格。

**範例 1：傳送給 REXX 程式的 SQL 結果組**

```
%DEFINE{  
DATABASE = "iddata"  
MyTable = %TABLE(ALL)  
DTW_DEFAULT_REPORT = "NO"  
%}  
  
%FUNCTION(DTW_SQL) Query(OUT table) {  
select * from survey  
%}  
  
%FUNCTION(DTW_REXX) showTable(INOUT table) {  
Say '列數：'table_ROWS  
Say '欄數：'table_COLS  
do j=1 to table_COLS
```

```

        Say "下面是所有的直欄值" table_N.j " : "
        do i = 1 to table_ROWS
            Say "<B>"i"</B>: " table_V.i.j
        end
    end
%}

%HTML(report){
<HTML>
<PRE>
@Query(MyTable)
<p>
@showTable(MyTable)
</PRE>
</HTML>
%}

```

該 HTML 報告區塊會呼叫 SQL 查詢、將結果儲存在表格變數中、然後將該變數傳給 REXX 函數。

---

## Net.Data 表格處理程序變數

除非另有附註，否則 Net.Data 定義這些變數乃是用於 REPORT 及 ROW 區塊。請使用這些變數參照您的查詢所傳回的值。

- 第66頁的『Nn』
- 第67頁的『NLIST』
- 第68頁的『NUM\_COLUMNS』
- 第69頁的『NUM\_ROWS』
- 第70頁的『ROW\_NUM』
- 第71頁的『TOTAL\_ROWS』
- 第72頁的『V\_columnName』
- 第73頁的『VLIST』
- 第74頁的『Vn』

## Nn

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

含有對直欄  $n$  的函數呼叫或查詢所傳回的直欄名稱。

您可參照在 REPORT 及 ROW 區塊中的  $Nn$ 。

### 範例

**範例 1：**直欄名稱的變數參照

直欄 2 的名稱是 \$(N2)。

**範例 2：**使用 DTW\_ASSIGN 儲存直欄名稱的值，用於 REPORT 區塊之外

```
%define coll=""
...
%function (DTW_SQL) myfunc() {
    select * from atable
%report {
    @dtw_assign(coll, N1)
    %row{ %}
    %}
%}

%html(report) {
@myfunc()
第一直欄的名稱是 $(coll)
%}
```

下面範例示範如何藉由 DTW\_ASSIGN 將這個變數使用於 REPORT 區塊之外。有關詳細資訊，請參閱第163頁的『DTW\_ASSIGN』。

**範例 3：**在 HTML 表格內用來定義直欄名稱的  $Nn$

```
%REPORT{
<H2>產品目錄</H2>
<TABLE BORDER=1 CELLPADDING=3>
<TR><TD>$(N1)</TD><TD>$(N2)</TD><TD>$(N3)</TD>
%ROW{
<TR><TD>$(V1)</TD><TD>$(V2)</TD><TD>$(V3)</TD>
%}
</TABLE>

%}
```



## NLIST

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

包含所有來自函數呼叫或查詢結果的直欄名稱。其預設的區隔符號是一個空格。

您可參照在 **REPORT** 及 **ROW** 區塊中的 **NLIST**。

### 範例

**範例 1：**有 **ALIGN** 的一串直欄名稱

```
%DEFINE ALIGN="YES"
...
%FUNCTION (DTW_SQL) myfunc() {
select * from MyTable
  %report{
您的查詢是在這些直欄上：$(NLIST)。
%row {
...
%}
%}
%}
```

在直欄名稱的列示中，直欄名稱之間是以空格加以分隔 (將 **ALIGN** 設為 **YES**)。

**範例 2：**使用 **%LIST** 變數將區隔符號改為 " | "

```
%DEFINE %LIST " | " NLIST
...
%FUNCTION (DTW_SQL) myfunc() {
select * from MyTable
  %report{
您的查詢是在這些直欄上：$(NLIST)。
%row {
...
%}
%}
%}
```

## NUM\_COLUMNS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

Net.Data 在報告區塊中處理的表格直欄數目；這些直欄數是由函數呼叫或查詢傳回。

您可參照在 REPORT 及 ROW 區塊中的的 NUM\_COLUMNS。

### 範例

**範例 1：** 將 NUM\_COLUMNS 作為有 NLIST 的變數參照使用

```
%REPORT{
您的查詢結果有 $(NUM_COLUMNS) 個直欄：$(NLIST)。
...
%}
```

## NUM\_ROWS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

Net.Data 在 REPORT 區塊中處理的表格列數。列數會受 *upper limit* 參數的值影響，該值是定義來供保留資料的 Net.Data 表格使用。例如，如果上限設定為 30，但是 SELECT 陳述式傳回 1000 列，則 NUM\_ROWS 的值為 30。此外，如果上限設定為 30，而 SELECT 陳述式傳回 20 列，則 NUM\_ROWS 等於 20。有關 TABLE 陳述式及上限參數的詳細資訊，請參閱第53頁的『TABLE 陳述式』。

只要 START\_ROW\_NUM 不被傳到語言環境，NUM\_ROWS 就不受 START\_ROW\_NUM 值的影響。例如，如果 START\_ROW\_NUM 設定為 5（設定顯示在 Web 首頁上的表格應由第 5 列開始移項），而 SELECT 陳述式傳回 25 列，則 NUM\_ROWS 將被設成 25，而非 21。前四列將從表格中被捨棄，但仍內含於 NUM\_ROWS 的值中。然而，如果 START\_ROW\_NUM 被傳到語言環境，則 NUM\_ROWS 只會包含由 START\_ROW\_NUM 指定開始的列數。在以上範例中，NUM\_ROWS 會被設定為 21。

您可參照在 REPORT 及 ROW 區塊中的 NUM\_ROWS。

### 範例

**範例 1：**顯示在 REPORT 區塊被處理的名稱數目。

```
%DEFINE DTW_SET_TOTAL_ROWS="YES"

%REPORT{
<H2>電子郵件目錄</H2>
<UL>
%ROW{
<LI>名稱： <a href="將電子郵件傳到：$(V1)">$(V2)</a><BR>
位置：$(V3)
%}
</UL>
所顯示的名稱：$(NUM_ROWS)<BR>
所找到的名稱：$(TOTAL_ROWS)
%}
```

## ROW\_NUM

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

一個表格變數，每當在 Net.Data 表格中的一列處理完後，Net.Data 會增加它的值。這個變數有如一箇計數器，它的值就是目前正被處理的列數。

RPT\_MAX\_ROWS 會影響 ROW\_NUM 的值。例如，如果表格中有 100 列，而您已設定 RPT\_MAX\_ROWS 為20，則 ROW\_NUM 最後的值為20，因為最後被處理是第 20 列。

您只能參照 ROW 區塊中的 ROW\_NUM。

### 範例

**範例 1：**在 HTML 輸出中移入一個直欄，並用 ROW\_NUM 替表格中的每一個橫列製作標籤

```
%REPORT{
<TABLE BORDER=1>
<TR><TD> 列號 </TD> <TD> 客戶 </TD>
%ROW{
<TR><TD> $(ROW_NUM) </TD> <TD> $(V_custname) </TD>
%}
</TABLE>
%}
```

REPORT 區塊產生的表格如下所示：

列號	客戶
1	Jane Smith
2	Jon Chiu
3	Frank Nguyen
4	Mary Nichols

## TOTAL\_ROWS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

查詢所傳回的總列數，無論 TABLE 語言結構 *upper\_limit* 的值為何。比方說，如果 RPT\_MAX\_ROWS 被設定的最大值為 20 列，但是查詢卻傳回 100 列，則在 ROW 處理之後，此變數將設為 100。

### 作業系統的差異性：

- 在 OS/400 作業系統上，這個變數可以在 REPORT 或 ROW 區塊的任何地方被參考。
- 在 OS/390、OS/2、Windows NT 和 UNIX 作業系統上，這個變數只能在 REPORT 標底下被參考。

**語言環境限制：**此變數僅用於下列資料庫語言環境：

- SQL
- ODBC
- Oracle
- Sybase

**必要條件：**您必需將 DTW\_SET\_TOTAL\_ROWS 設定為 YES，才能使用這個變數。有關詳細資訊，請參閱第92頁的『DTW\_SET\_TOTAL\_ROWS』。

### 範例

**範例 1：**顯示已尋找到的名稱號碼

```
%DEFINE DTW_SET_TOTAL_ROWS="YES"

%REPORT{
<H2>電子郵件目錄</H2>
<UL>
%ROW{
<LI>名稱： <a href="將電子郵件傳到：$(V1)">$(V2)</a><BR>
位置：$(V3)
%}
</UL>
所找到的名稱：$(TOTAL_ROWS)
%}
```

## V\_columnName

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

現行列之指定欄名的值。未定義的欄名不會有變數存在。含有兩個同名直欄的查詢，會產生不可預期的結果。請考慮在您的 SQL 中使用一個 AS 子句，將重複的欄名加以更改。

您可參照只在 ROW 區塊中的 *V\_columnName*。

### 值

*V\_columnName*

表 1. *V\_columnName* 值

值	說明
<i>columnName</i>	資料庫表格現行列的直欄名稱。

### 範例

**範例 1：** 使用 *V\_columnName* 作為變數參照

```
%FUNCTION(DTW_SQL) myQuery() {  
  SELECT NAME, ADDRESS from $(qtable)  
%REPORT{  
  
%ROW{  
  
  在列 $(ROW_NUM) 中之「名稱」直欄的值是$(V_NAME)。<BR>  
%}  
%}  
%}
```

## VLIST

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

一列目前在 ROW 區塊中所處理的現行列之所有欄位值。

您可參照只在 ROW 區塊中的 VLIST。其預設的區隔符號是一個空格。

### 範例

**範例 1：**使用列示標籤來顯示查詢結果

```
%DEFINE ALIGN="YES"

%REPORT{
下面是您的查詢結果：
<OL>
%ROW{
<LI>$(VLIST)
%}
</OL>
%}
```

**範例 2：**使用列示變數將區隔符號變更為<P>

```
%DEFINE %LIST "<P>" VLIST

%REPORT{
下面是您的查詢結果：
%ROW{
<HR>$(VLIST)
%}
%}
```

## Vn

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

現行橫列之指定直欄號碼  $n$  的值。

您可參照只在 ROW 區塊中的  $Vn$ 。

Net.Data 為表格的每一個欄位指定變數；在變數參照中使用此變數，指定您要參考的欄位號碼。若要使用這個變數在區塊之外，可指定  $Vn$  的值給一個先前已定義的廣域變數或一個 OUT 或 INOUT 函數參數變數。

### 範例

**範例 1：**顯示 HTML 表格的報告

```
%REPORT{
<H2>電子郵件目錄</H2>
<TABLE BORDER=1 CELLPADDING=3>
<TR><TD>名稱</TD><TD>電子郵件位址</TD><TD>位置</TD>
%ROW{
<TR><TD>$(V1)</TD>
<TD><a href="mailto:$(V2)">$(V2)</a></TD>
<TD>$(V3)</TD>
%}
</TABLE>
%}
```

第二欄則顯示電子郵件位址。您可以按一下該鏈結，傳送一則訊息給某人。



---

## Net.Data 報告變數

這些變數可以幫助您自行設定報告。每個變數都有一個預設值。您可以指定新的值給變數，置換預設值。

- 第76頁的『ALIGN』
- 第77頁的『DTW\_DEFAULT\_REPORT』
- 第78頁的『DTW\_HTML\_TABLE』
- 第79頁的『RPT\_MAX\_ROWS』
- 第80頁的『START\_ROW\_NUM』

## ALIGN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

負責控制與報告處理變數 `NLIST` 和 `VLIST` 同用的前後空格。

**效能要訣：**僅在需要 `Net.Data` 決定表格中所有直欄的最大直欄長度，來計算填補需求時，才使用 `ALIGN`。這個處理會影響執行效能。

當它被設定為 `YES` 時，`ALIGN` 會提供填補，對齊表格處理變數，以便顯示。如果要將查詢結果內含在 `HTML` 鏈結或套表動作中，可使用預設值 `NO`，以防止 `Net.Data` 在報告變數前後加上空白。

使用 `DEFINE` 陳述式或以 `@DTW_ASSIGN()` 函數來設定這個變數值。

### 值

`ALIGN="YES" | "NO"`

表 2. *ALIGN* 值

值	說明
YES	<code>Net.Data</code> 會新增前端及尾端的空格來對齊含有空格的報告變數，以利顯示。
NO	<code>Net.Data</code> 不會新增前端或尾端的空格。 <code>NO</code> 是預設值。

### 範例

**範例 1：**使用 `ALIGN` 變數用空格來分隔每一個直欄

```
%DEFINE ALIGN="YES"
<P>您的查詢在這些直欄上：$(NLIST)
```

## DTW\_DEFAULT\_REPORT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

決定 Net.Data 是否為不具 REPORT 區塊的函數產生預設報告。此變數設定為 YES，Net.Data 會產生預設報告；設定為 NO，Net.Data 則抑制預設報告的產生。抑制不產生預設的報告是有用的，例如，如果您在表格變數中收到函數呼叫的結果，而希望將這些結果傳給另一個函數加以處理時，即可捨棄預設的報告。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

DTW\_DEFAULT\_REPORT="YES"|"NO"|"MULTIPLE"

表 3. DTW\_DEFAULT\_REPORT 值

值	說明
YES	Net.Data 會替不具 REPORT 區塊的函數產生預設報告，並將結果顯示在瀏覽器上。YES 是預設值。
NO	Net.Data 會捨棄替不具 REPORT 區塊的函數所預設的報告。
MULTIPLE	Net.Data 在具有許多 REPORT 區塊的函數中，對於未指定到 REPORT 區塊的結果組或輸出表格，產生預設的報告。

### 範例

**範例 1：**置換 Net.Data 所產生的預設報告

```
%DEFINE DTW_DEFAULT_REPORT="NO"
```

## DTW\_HTML\_TABLE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將結果顯示在 HTML 表格中，而非將表格以文字類型格式顯示（也就是，使用 TABLE 標籤，而非 PRE 標籤）。

所產生的 TABLE 標籤中，包括邊框和資料格填補規格：

```
<TABLE BORDER CELLPADDING=2>
```

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

```
DTW_HTML_TABLE="YES"|"NO"
```

表 4. DTW\_HTML\_TABLE 值

值	說明
YES	使用 HTML 表格標籤顯示表格資料。
NO	使用 PRE 標籤，顯示文字格式的表格資料。 NO 是預設值。

### 範例

**範例 1：**用 HTML 標籤顯示 SQL 函數的結果

```
%DEFINE DTW_HTML_TABLE="YES"
```

```
%FUNCTION(DTW_SQL){  
SELECT NAME, ADDRESS FROM $(qTable)  
%}
```

## RPT\_MAX\_ROWS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定在函數 **REPORT** 區塊內或產生預設報告時（若未指定 **REPORT** 區塊）要進行處理的表格列數。

資料庫語言環境使用此變數，限制傳回的列數，如此可大幅改進大型結果組的執行效能。此變數與 **START\_ROW\_NUM** 可將大型結果組的查詢分割成較小的表格，每一個表格有其自己的 **HTML** 頁。

**OS/400、Windows NT、OS/2 及 UNIX 使用者：** 若要將此變數傳遞至語言環境，請在 **Net.Data** 起始設定檔的資料庫語言環境 **ENVIRONMENT** 陳述式中，將此變數設定為 **IN** 參數。關於資料庫語言環境陳述式的詳細資訊，請參閱您作業系統的 *Net.Data* 管理和程式設計指南中架構這一章。

**OS/390 使用者：** 定義於巨集中的 **RPT\_MAX\_ROWS**，會暗中傳遞至資料庫語言環境。

使用 **DEFINE** 陳述式或 **@DTW\_ASSIGN()** 函數，指定這個變數的值。

### 值

`RPT_MAX_ROWS="ALL"|"0"|"number"`

表 5. *RPT\_MAX\_ROWS* 值

值	說明
ALL	顯示在沒有限制由函數呼叫所產生的表格中的列數。所有列數都會顯示。
0	指定表格中所有的列數都會被顯示。這個值同於指定 <b>ALL</b> 。
<i>number</i>	一個正整數，用於顯示在由函數 <b>call</b> 區塊產生的表格中的列數。 如果 <b>FUNCTION</b> 區塊包含 <b>REPORT</b> 及 <b>ROW</b> 區塊，則這個數字所指定的是執行 <b>ROW</b> 的次數。

### 範例

**範例 1：**在 **DEFINE** 陳述式中定義 **RPT\_MAX\_ROWS**

```
%DEFINE RPT_MAX_ROWS="20"
```

以上的方法可將任何函數所傳回的列數限制為 20。

**範例 2：**使用 **HTML** 輸入，以針對 **HTML** 套表定義變數

要傳回的列數上限（0 表示沒有限制）：

```
<INPUT TYPE="text" NAME="RPT_MAX_ROWS" SIZE=3>
```

以上幾行可以置於 **FORM** 標籤中，讓應用程式使用者設定他們希望讓查詢所傳回的列數。

## START\_ROW\_NUM

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定表格中的起始列號，此列將於函數 **REPORT** 區塊內或產生預設報告時（若未指定 **REPORT** 區塊）進行處理。

資料庫語言環境使用此變數，決定結果組內要開始處理的起始列。爲了顯著改善大型結果組的執行效能，請使用此變數與 **RPT\_MAX\_ROWS**，將大型結果組分割成較小的表格。

**OS/400、Windows NT、OS/2 及 UNIX 使用者：** 若要將此變數傳遞至語言環境，請在 **Net.Data** 起始設定檔的資料庫語言環境 **ENVIRONMENT** 陳述式中，將此變數設定爲 **IN** 參數。關於資料庫語言環境陳述式的詳細資訊，請參閱您作業系統的 *Net.Data* 管理和程式設計指南中架構這一章。

**OS/390 使用者：** 定義於巨集中的 **START\_ROW\_NUM**，會暗中傳遞至資料庫語言環境

使用 **DEFINE** 陳述式或以 **@DTW\_ASSIGN()** 函數來設定這個變數值。

### 值

**START\_ROW\_NUM**="number"

表 6. **START\_ROW\_NUM** 值

值	說明
<i>number</i>	<p>一個正整數，代表開始顯示報告的列號。預設值是 1。</p> <p>如果 <b>START_ROW_NUM</b> 在起始設定檔案中的資料庫語言環境之 <b>ENVIRONMENT</b> 陳述式中被指定，則這個號碼所代表的是資料庫語言環境已經處理的結果組之列號。</p> <p>如果 <b>START_ROW_NUM</b> 沒有被傳送到語言環境，則這個號碼所指定的是 <b>Net.Data</b> 表格的列號，用於顯示報告。</p>

### 範例

**範例 1：** 使用 **HTML** 套表的“下一步”及“前一步”按鈕捲動

```
%define {
  DTW_HTML_TABLE      = "YES"
  START_ROW_NUM        = "1"
  RPT_MAX_ROWS         = "10"
  totalSize            = ""
  includeNext          = "YES"
  includePrev          = "YES"
  includeLast          = "YES"
  includeFirst         = "YES"
}%

%function(DTW_SQL) myQuery(){
  select * from NETDATADEV.CUSTOMER
}%
```

```

%function(DTW_SQL) count(OUT size){
    select count(*) from NETDATADEV.CUSTOMER
    %report{
        %row{
            @DTW_ASSIGN(size,V1)
        }
    }
}

%html(report) {
    %{ 如果尚未完成，先取得記錄總數 %}
    %if (totalSize == "")
        @count(totalSize)
    %endif

    %{ 根據使用者按下的按鈕設定 START_ROW_NUM %}
    %if (totalSize <= RPT_MAX_ROWS)
        %{ 只有一頁的資料 %}
        @DTW_ASSIGN(START_ROW_NUM, "1")
        @DTW_ASSIGN(includeFirst, "NO")
        @DTW_ASSIGN(includeLast, "NO")
        @DTW_ASSIGN(includeNext, "NO")
        @DTW_ASSIGN(includePrev, "NO")
    %elif (submit == "First Page" || submit == "")
        %{ 首次使用的，或由使用者選取的“首頁”按鈕 %}
        @DTW_ASSIGN(START_ROW_NUM, "1")
        @DTW_ASSIGN(includePrev, "NO")
        @DTW_ASSIGN(includeFirst, "NO")
    %elif (submit == "Last Page")
        %{ 使用者選取的“末頁”按鈕 %}
        @DTW_SUBTRACT(totalSize, RPT_MAX_ROWS, START_ROW_NUM)
        @DTW_ADD(START_ROW_NUM, "1", START_ROW_NUM)
        @DTW_ASSIGN(includeLast, "NO")
        @DTW_ASSIGN(includeNext, "NO")
    %elif (submit == "Next")
        %{ 使用者選取的“下一步”按鈕 %}
        @DTW_ADD(START_ROW_NUM, RPT_MAX_ROWS, START_ROW_NUM)
        %if (@DTW_rADD(START_ROW_NUM, RPT_MAX_ROWS) > totalSize)
            @DTW_ASSIGN(includeNext, "NO")
            @DTW_ASSIGN(includeLast, "NO")
        %endif
    %elif (submit == "Previous")
        %{ 使用者選取的“前一步”按鈕 %}
        @DTW_SUBTRACT(START_ROW_NUM, RPT_MAX_ROWS, START_ROW_NUM)
        %if (START_ROW_NUM <= "1" )
            @DTW_ASSIGN(START_ROW_NUM, "1")
            @DTW_ASSIGN(includePrev, "NO")
            @DTW_ASSIGN(includeFirst, "NO")
        %endif
    %endif
}

%{ 執行查詢以取得資料 %}
@myQuery()

%{ 在報告底端輸出正確的按鈕 %}
<center>
<form method="POST" action="report">
<input name="START_ROW_NUM" type="hidden" value="${START_ROW_NUM}">
<input name="totalSize" type="hidden" value="${totalSize}">
%if (includeFirst == "YES" )
    <input name="submit" type="submit" value="First Page">
%endif
%if (includePrev == "YES" )
    <input name="submit" type="submit" value="Previous">
%endif
%if (includeNext == "YES" )
    <input name="submit" type="submit" value="Next">

```

```
%endif
    %if (includeLast == "YES" )
        <input name="submit" type="submit" value="Last Page">
    %endif
</form>
</center>
%}
```



使用這些變數可幫助您使用語言環境自行設定 FUNCTION 區塊的處理方式。每個變數都有一個預設值。您可以將新的值指定給變數，置換預設值。

- 第84頁的『DATABASE』
- 第85頁的『DB\_CASE』
- 第86頁的『DB2PLAN』
- 第87頁的『DB2SSID』
- 第88頁的『DTW\_APPLET\_ALTTEXT』
- 第89頁的『DTW\_EDIT\_CODES』
- 第90頁的『DTW\_PAD\_PGM\_PARMS』
- 第91頁的『DTW\_SAVE\_TABLE\_IN』
- 第92頁的『DTW\_SET\_TOTAL\_ROWS』
- 第93頁的『LOCATION』
- 第94頁的『LOGIN』
- 第95頁的『NULL\_RPT\_FIELD』
- 第96頁的『PASSWORD』
- 第97頁的『SHOWSQL』
- 第98頁的『SQL\_STATE』
- 第99頁的『TRANSACTION\_SCOPE』

## DATABASE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X		X	X	X	X

### 目的

呼叫資料庫函數時，指定存取的資料庫或 ODBC 資料來源。在巨集中可以多次變更這個變數，以存取多重的資料庫或 ODBC 資料來源。

在 **OS/400** 作業系統上：這個變數是可選用的。根據預設值，Net.Data 會指定 DATABASE="\*LOCAL"；DTW\_SQL 語言環境是使用區域關聯式資料庫目錄登錄。

在 **Windows NT、OS/2 及 UNIX** 作業系統上：除了當使用 DTW\_ORA (Oracle) 語言環境以外，請在呼叫任何資料庫函數前，定義這個變數。此外，當自同一個 HTML 區塊或經由相同的語言環境存取多重資料庫時，您必需使用「現場連線」。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

DATABASE="dbname"

表 7. DATABASE 值

值	說明
dbname	Net.Data 所連接的資料庫名稱。

### 範例

**範例 1：**指定任何 SQL 作業均連接 CELDIAL 資料庫

```
%DEFINE DATABASE="CELDIAL"

%FUNCTION (DTW_SQL) getRpt() {
SELECT * FROM customer
%}

%HTML(report){
%INCLUDE "rpthead.htm"
@getRpt()
%INCLUDE "rptfoot.htm"
%}
```

當函數 getRpt 被呼叫時，即可存取資料庫 CELDIAL。

**範例 2：**使用 DTW\_ASSIGN 置換先前的 DATABASE 定義

```
%DEFINE DATABASE="DB2C1"
...
%HTML(monthRpt){
@DTW_ASSIGN(DATABASE, "DB2D1")
%INCLUDE "rpthead.htm"
@getRpt()
%INCLUDE "rptfoot.htm"
%}
```

不管 DATABASE 之前的值是什麼，HTML 區塊都會查詢資料庫 DB2D1。

## DB\_CASE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定 SQL 指令所用的字體，並將所有字元轉換成大寫或小寫字體。如果沒有定義這個變數，則預設動作是不轉換 SQL 指令字元。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

DB\_CASE="UPPER"|"LOWER"

表 8. DB\_CASE 值

值	說明
UPPER	將所有 SQL 指令字元轉換成大寫字體。
LOWER	將所有 SQL 指令字元轉換成小寫字體。

### 範例

**範例 1：**指定所有 SQL 指令皆使用大寫字體

```
%DEFINE DB_CASE="UPPER"
```

## DB2PLAN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				X				

### 目的

配置一個連接區域 DB2 子系統的規劃。這個變數會在 Net.Data 將存取的區域 DB2 子系統上為 Net.Data SQL 語言環境指定一個規劃名稱。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

**需求：** 為了讓巨集中此變數的值生效，變數必須列示在 SQL 語言環境的 ENVIRONMENT 陳述式中。

### 值

DB2PLAN="*plan\_name*"

表 9. DB2PLAN 值

值	說明
<i>plan_name</i>	DB2 規劃的名稱。這個名稱可以是 8 個以下的字元。

### 範例

**範例 1：**在 DEFINE 陳述式中指定規劃

手%DEFINE DB2PLAN="DTWGAV22"

## DB2SSID

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				X				

### 目的

建立一個到區域 DB2 子系統的連線。此變數會指定 Net.Data 所要存取的區域 DB2 子系統的子系統 ID。每一個巨集只能有一個區域資料庫連線。

**需求：** 爲了讓巨集中此變數的值生效，變數必須列示在 SQL 語言環境的 ENVIRONMENT 陳述式中。

### 值

DB2PLAN="*subsytem\_id*"

表 10. DB2SSID 值

值	說明
<i>subsystem_id</i>	DB2 子系統的名稱。 這個名稱可以是 8 個以下的字元。

### 範例

**範例 1：** 在 DEFINE 陳述式中指定一個子系統 ID

```
%DEFINE DB2SSID="DBNC"
```

## DTW\_APPLET\_ALTTEXT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將 HTML 標籤及文字顯示給不認得 APPLET 標籤的瀏覽器，且與 Applet 語言環境共同使用。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

DTW\_APPLET\_ALTTEXT="HTML\_text\_and\_tags"

表 11. DTW\_APPLET\_ALTTEXT 值

值	說明
HTML_text_and_tags	HTML 標籤及文字，讓不認得 APPLET 標籤的瀏覽器使用。

### 範例

**範例 1：**指示 Web 瀏覽器限制的替代文字

```
%DEFINE DTW_APPLET_ALTTEXT="<P>對不起，您的瀏覽器不支援 java。"
```

## DTW\_EDIT\_CODES

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

將 SQL 作業結果所傳回的 NUMERIC、DECIMAL、INTEGER 及 SMALLINT 資料類型轉換成適用於 DTW\_SQL 語言環境。變數 DTW\_EDIT\_CODES 是一串字元，對應 DTW\_SQL LE 所建置的表格直欄結果；例如，DTW\_EDIT\_CODES 中的第五個字元將被引用在結果組的第五個欄位，但該欄位必需是受支援型類之一。這個單一字元可以是定義在 *Data 資料說明規格參考手冊* 中任何一個已支援系統所提供的編輯碼。

例如，DECIMAL(6,0) 欄位在正常狀況下會顯示出字串 '112698'。若指定變數 DTW\_EDIT\_CODES 中的直欄為一個編輯碼 'Y' 時，該欄位會顯示出結果表格中之相對應直欄的字串，代表日期 '11/26/98'。

**要訣：** 引用一個使用者自定的編輯碼給一個直欄，而造成字串含有非數值字元 (如逗點或貨幣符號) 時，如果該字串被送回伺服器在 Net.Data 中進行後續處理，則會導致語法錯誤。例如，非數值的欄位值可能被用於後續 DTW\_SQL 函數呼叫中的數字比較，因而造成語法錯誤。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

DTW\_EDIT\_CODES="edit\_code"

表 12. DTW\_EDIT\_CODES 值

值	說明
edit_code	指定一個字串，對應 SQL 語言環境所建置表格的結果直欄。

### 範例

#### 範例 1：

```
@DTW_ASSIGN(DTW_EDIT_CODES "JJLJJ*****Y")
```

## DTW\_PAD\_PGM\_PARMS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

指示語言環境將字元參數（CHAR 或 CHARACTER 的資料類型）傳遞至程式或儲存程序時，是否要填補空格。

就 IN 或 INOUT 參數而言，若參數值長度小於指定的精確度，空格將插入參數值的右側，直到參數值與精確度兩者的長度相同為止。

就 OUT 參數而言，參數值設為精確度的空格。

呼叫程式或儲存程序之後，從 OUT 和 INOUT 參數值中除去所有的尾端空格。

在起始設定檔案中設定這個變數，指定一個值給您所有的巨集。您可以在巨集中定義這個值，將它置換。若巨集中未定義 DTW\_PAD\_PGM\_PARMS，則會使用 Net.Data 起始設定檔的值。

Direct Call 及 SQL 語言環境支援 DTW\_PAD\_PGM\_PARMS。

### 值

DTW\_PAD\_PGM\_PARMS="YES"|"NO"

表 13. DTW\_PAD\_PGM\_PARMS 值

值	說明
YES	所有 IN 和 INOUT 字元參數值皆向左對齊，在傳遞至程式或儲存程序之前，會填補參數精確度的空格數。呼叫程式或儲存程序之後，除去尾端空格。
NO	參數傳遞至程式或儲存程序時，不填補字元參數值（以 NULL 結尾的值）。呼叫程式或儲存程序之後，不除去尾端空格。

### 範例

#### 範例 1：參數填補空格

DTW\_PAD\_PGM\_PARMS="YES"



## DTW\_SAVE\_TABLE\_IN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

識別 SQL 語言環境用來存放來自查詢之表格資料的表格變數。此表可以稍後再用，例如，用於分析表格資料的 REXX 程式。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

DTW\_SAVE\_TABLE\_IN="table\_name\_var"

表 14. DTW\_SAVE\_TABLE\_IN 值

值	說明
table_name_var	SQL 語言環境用來存放來自查詢之表格資料的表格名稱。

### 範例

**範例 1：** 用於 REXX 呼叫且先前已定義的表格變數

```
%DEFINE theTable = %TABLE(2)
%DEFINE DTW_SAVE_TABLE_IN = "theTable"

%FUNCTION(DTW_SQL) doQuery() {
  SELECT MODNO, COST, DESCRIP FROM EQPTABLE
  WHERE TYPE='MONITOR'
}%

%FUNCTION(DTW_REXX) analyze_table(myTable) {
  %EXEC{ anzTbl.cmd %}
}%

%HTML(doTable) {
  @doQuery()
  @analyze_table(theTable)
}%
```

REXX FUNCTION 區塊會呼叫 REXX 程式 anzTbl.cmd，該程式是使用表格變數 theTable 來分析表格中的資料。變數 theTable 是由之前一個 SQL 函數呼叫所傳回。

## DTW\_SET\_TOTAL\_ROWS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定資料庫語言環境中查詢結果組的總列數必需被指定給 `TOTAL_ROWS`。

使用 `DEFINE` 陳述式或以 `@DTW_ASSIGN()` 函數來設定這個變數值。

**OS/400、OS/2、Windows NT 及 UNIX 使用者：**若要將此變數傳遞至語言環境，請在 `Net.Data` 起始設定檔的資料庫語言環境 `ENVIRONMENT` 陳述式中，將此變數設定為 `IN` 變數。請參閱 *Net.Data 管理及程式設計手冊* 中的架構章節，以取得關於資料庫語言環境陳述式的詳細資料。

**OS/390 使用者：**定義於巨集中的 `DTWT_SET_TOTAL_ROWS`，會暗中傳遞至資料庫語言環境

**執行效能要訣：**將 `DTW_SET_TOTAL_ROWS` 設定為 `YES` 會影響執行效能，因為若要決定總列數，資料庫語言環境需要所有列數都被取回。

### 值

`DTW_SET_TOTAL_ROWS="YES"|"NO"`

表 15. `DTW_SET_TOTAL_ROWS` 值

值	說明
YES	指定總列數的值給 <code>TOTAL_ROWS</code> 變數。 <b>重要事項：</b> 如果想要參考變數 <code>TOTAL_ROWS</code> 來決定查詢所傳回的列數，則必需設定此值。
NO	<code>Net.Data</code> 不會設定 <code>TOTAL_ROWS</code> 變數且 <code>TOTAL_ROWS</code> 無法在巨集中被參考。 <code>NO</code> 是預設值。

### 範例

**範例 1：**定義 `DTW_SET_TOTAL_ROWS` 以便使用 `TOTAL_ROWS`

```
%DEFINE DTW_SET_TOTAL_ROWS="YES"

...

%FUNCTION (DTW_SQL) myfunc() {
select * from MyTable
%report {
...
%row
...
%}
<P>您的查詢僅限於 $(TOTAL_ROWS) 列。此查詢傳回太多列。
%}
%}
```

## LOCATION

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				X				

### 目的

建立一個到遠端資料庫伺服器的連線。此變數會指定區域 DB2 子系統據以識別該遠端伺服器的名稱。LOCATION 的值，必須在「通信資料庫」(CDB) 的 SYSIBM.SYSLOCATIONS 表格中加以定義。如果巨集中沒有定義這個變數，則該巨集所發出的任何 SQL 要求，都會在區域 DB2 子系統上執行。

**需求：** 爲了讓巨集中此變數的值生效，變數必須列示在 SQL 語言環境的 ENVIRONMENT 陳述式中。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

LOCATION="remote\_dbase\_name"

表 16. LOCATION 值

值	說明
remote_dbase_name	定義於 CDB 的 SYSIBM.SYSLOCATIONS 表格中有效遠端資料庫伺服器的名稱。這個名稱可以是 8 個以下的字元。

### 範例

**範例 1:** 在 DEFINE 陳述式中定義遠端資料庫位置

```
%DEFINE LOCATION="QMFDJ00"
```

## LOGIN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X		X	X	X	X

### 目的

傳送使用者 ID 給資料庫語言環境，以提供對受保護的資料的存取權限。您可以將這個變數與通行碼一起使用，以納入 DB2 的安全演繹法。

**OS/400 使用者：**如果未定義 DATABASE 變數，或是如果它設定為值 `"*LOCAL"`，OS/400 將不處理 LOGIN 與 PASSWORD。會透過 Net.Data 執行時所用的使用者設定檔，來遞送資料庫存取。

**安全要訣：**雖然您可以將這個撰寫在 Net.Data 巨集內，最好讓應用程式使用者在 HTML 套表上輸入使用者 ID。另外，使用 Web 伺服器 ID 的預設值所提供的存取層次，可能無法符合您的安全需要。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

LOGIN="database\_user\_id"

表 17. LOGIN 值

值	說明
database_user_id	有效的資料庫使用者 ID。其預設值是使用啟動 Web 伺服器的使用者 ID。

### 範例

**範例 1：**將存取權限制給使用者 ID, DB2USER

```
%DEFINE LOGIN="DB2USER"
```

**範例 2：**使用 HTML 套表輸入行

```
USERID&#58; <INPUT TYPE="text" NAME="LOGIN" SIZE=6>
```

此例所示範的這一行，可以併入 HTML 套表中，讓應用程式使用者輸入其使用者 ID。

## NULL\_RPT\_FIELD

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

指定一個字串讓使用者可以提供給 DTW\_SQL 語言環境，來代表 SQL 結果組所傳回的 NULL 值。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

NULL\_RPT\_FIELD="null\_char"

表 18. NULL\_RPT\_FIELD 值

值	說明
<i>null_char</i>	指定一個字串，來代表 SQL 結果組所傳回的 NULL 值。其預設值是空字串。

### 範例

**範例 1：**在 SQL 語言環境中指定一個字串來代表 NULL 值

```
%DEFINE NULL_RPT_FIELD = "++++"
```

## PASSWORD

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X		X	X	X	X

### 目的

傳送一個密碼給資料庫語言環境，以提供對受保護的資料的存取權限。您可以將這個變數與 LOGIN 一起使用，以納入 DB2 的安全演繹法。

**OS/400 使用者：**如果未定義 DATABASE 變數，或是如果它設定為值 `"*LOCAL"`，OS/400 將不處理 LOGIN 與 PASSWORD。會透過 Net.Data 執行時所用的使用者設定檔，來遞送資料庫存取。

**安全要訣：**雖然您可以將這個撰寫在 Net.Data 巨集內，最好讓應用程式使用者在 HTML 套表上輸入密碼。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

PASSWORD="*password*"

表 19. PASSWORD 值

值	說明
<i>password</i>	指定一個有效密碼，以提供自動對資料庫語言環境的存取權限。

### 範例

**範例 1：**將存取權限限制給使用密碼 NETDATA 的應用程式使用者

```
%DEFINE PASSWORD="NETDATA"
```

**範例 2:** HTML 套表輸入行

```
PASSWORD&#58; <INPUT TYPE="password" NAME="PASSWORD" SIZE=8>
```

此例所示範的這一行，可以併入 HTML 套表中，讓應用程式使用者輸入其密碼。

## SHOWSQL

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

隱藏或顯示 Web 瀏覽器所用之查詢的 SQL。在測試時顯示 SQL，特別有助於 Net.Data 巨集的除錯作業。唯有當 Net.Data 架構檔中的 DTW\_SHOWSQL 設為 YES 時，才能使用 SHOWSQL。有關 DTW\_SHOWSQL 架構變數的詳細資訊，請參閱您作業系統的 *Net.Data 管理和程式設計指南* 中架構這一章。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

SHOWSQL="YES"|"NO"

表 20. SHOW\_SQL 值

值	說明
YES	顯示送給資料庫之查詢的 SQL。
NO	隱藏送給資料庫之查詢的 SQL。NO 是預設值。

### 範例

**範例 1：**顯示所有 SQL 查詢。

架構檔：

```
DTW_SHOWSQL YES
```

巨集：

```
%DEFINE SHOWSQL="YES"
```

**範例 2：**設定是否使用 HTML 套表輸入來顯示 SQL。

架構檔：

```
DTW_SHOWSQL YES
```

巨集：

```
SHOWSQL: <INPUT TYPE="radio" NAME="SHOWSQL" VALUE="YES"> Yes  
          <INPUT TYPE="radio" NAME="SHOWSQL" VALUE="" CHECKED> No
```

## SQL\_STATE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

存取或顯示資料庫所傳回的 SQL 狀態值。

這是一種預設變數，其數值無法修改。使用這種變數作為變數參照。

### 範例

**範例 1：**在 REPORT 區塊中顯示 SQL 狀態

```
%FUNCTION (DTW_SQL) val1() {  
  select * from customer  
%REPORT {  
  ...  
  %ROW {  
  ...  
  %}  
  SQLSTATE=$(SQL_STATE)  
  %}
```



## TRANSACTION\_SCOPE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定 SQL 指令的異動範圍，來決定 Net.Data 是在 HTML 區塊中各個 SQL 指令順利完成之後或是所有 SQL 指令順利完成之後發出 COMMIT。若您設定的是所有 SQL 指令必需在發出確定前順利完成，只要有一個 SQL 指令沒有順利完成，都會使先前對該區塊同一資料庫所執行的 SQL 全數回復。

爲了讓 TRANSACTION\_SCOPE 變數生效，請在 Net.Data 架構檔的 ENVIRONMENT 陳述式中併入此變數。然後，即可使用 DEFINE 陳述式或 @DTW\_ASSIGN() 函數，指定這個變數的值。

**一致性注意事項：** 在 OS/400 和 OS/390 以外的平台上，那麼對收到失敗回應之資料庫所作的更新，可能會被回復，然如果下列條件全部成立的話，同一 HTML 區塊中對其它資料庫所作的更新，卻可能被確定：

- 指定了 TRANSACTION\_SCOPE = "MULTIPLE"
- 在一個 HTML 區塊中存取多個資料庫 (如果您是使用「現場連線」時就有可能)
- SQL 要求傳回失敗回應

如果您是從 OS/400 上的 Net.Data 存取多個資料庫，或是使用 IBM 的 DataJoiner 軟體，那麼當您從 Net.Data 更新時，即可取得多個資料庫更新協調和一致性。

在 OS/400 和 OS/390 上，如果 TRANSACTION\_SCOPE = "MULTIPLE"，則所有從單個 HTML 區塊發出的 IBM 資料庫更新，將一起確定或回復。

在 OS/400 以外的作業系統上，REXX、Perl 和 Java 語言環境在它們自己獨立的作業系統處理作業當中執行。因此，不管 Net.Data TRANSACTION\_SCOPE 值是什麼，只要是發自這些語言環境的資料庫更新，其確定或回復都與發自某個 Net.Data 巨集的資料庫更新互不影響。

### 值

TRANSACTION\_SCOPE="SINGLE"|"MULTIPLE"

表 21. TRANSACTION\_SCOPE 值

值	說明
SINGLE	只要 HTML 區塊中的每一個 SQL 指令順利完成，Net.Data 就會發出 COMMIT。
MULTIPLE	Net.Data 只會在 HTML 區塊中所有的 SQL 指令都順利完成之後，才會發出 COMMIT。MULTIPLE 是預設值。

### 範例

**範例 1：**指定在每一個異動之後發出 COMMIT

```
%DEFINE TRANSACTION_SCOPE="SINGLE"
```

---

## Net.Data 雜項變數

這些變數由 Net.Data 定義，用來影響 Net.Data 處理程序、尋找函數呼叫的狀態、與取得關於資料庫查詢結果的資訊，也用來決定關於檔案位置和日期的資訊。您可能會覺得這些變數用在您所寫的函數當中，或者用在測試 Net.Data 巨集時，相當得心應手。

- 第101頁的『DTW\_CURRENT\_FILENAME』
- 第102頁的『DTW\_CURRENT\_LAST\_MODIFIED』
- 第103頁的『DTW\_DEFAULT\_MESSAGE』
- 第104頁的『DTW\_LOG\_LEVEL』
- 第105頁的『DTW\_MACRO\_FILENAME』
- 第106頁的『DTW\_MACRO\_LAST\_MODIFIED』
- 第107頁的『DTW\_MBMODE』
- 第108頁的『DTW\_MP\_PATH』
- 第109頁的『DTW\_MP\_VERSION』
- 第110頁的『DTW\_PRINT\_HEADER』
- 第111頁的『DTW\_REMOVE\_WS』
- 第112頁的『RETURN\_CODE』

## DTW\_CURRENT\_FILENAME

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

現行輸入檔的名稱及副檔名。此輸入檔可能是 Net.Data 巨集，也可能 INCLUDE 陳述式所指定的檔案。

這是一種預設變數，其數值無法修改。使用這種變數作為變數參照。

### 範例

<P>這個檔案是 <I>\$(DTW\_CURRENT\_FILENAME)</I>，  
它是在 \$(DTW\_CURRENT\_LAST\_MODIFIED) 時更新。

## DTW\_CURRENT\_LAST\_MODIFIED

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

前次修改現行檔案的日期和時間。現行檔案可能是一個 `Net.Data` 巨集，也可能是 `INCLUDE` 陳述式中所指定的檔案。輸出格式是根據 `Net.Data` 執行的系統而定。

這是一種預設變數，其數值無法修改。使用這種變數作為變數參照。

### 範例

<P>這個檔案是 <I>\$(DTW\_CURRENT\_FILENAME)</I>，  
它是在 \$(DTW\_CURRENT\_LAST\_MODIFIED) 時更新。

## DTW\_DEFAULT\_MESSAGE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X		X	X	X	X

### 目的

包含當錯誤發生時，由某次呼叫傳給內建函數或語言環境的訊息文字。

您可在 Net.Data 巨集中的任何部分使用 DTW\_DEFAULT\_MESSAGE 變數。

這是一個已事先定義好的變數，建議您不要修改它的值。將它作為一個變數參照使用。

### 範例

**範例 1：** 告知函數是否順利完成的訊息

```
@function1()
%IF ("$(RETURN_CODE)" == "0")
    函數已經順利完成。
%ELSE
    函數失敗，傳回的回覆碼為 $(RETURN_CODE)。傳回的錯誤訊息為
    "$(DTW_DEFAULT_MESSAGE)"。
%ENDIF
```

**範例 2：** 當函數傳回一個非零回覆碼時的預設文字

```
%MESSAGE{
default: {<h2>Net.Data 收到回覆碼: $(RETURN_CODE).
錯誤訊息是 $(DTW_DEFAULT_MESSAGE)</h2> %} : continue
%}
```

如果函數所傳回的回覆碼不是 0，則使用者會看見預設的錯誤訊息。

## DTW\_LOG\_LEVEL

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X			X	X	X

### 目的

Net.Data 寫入日誌檔的訊息層次。

您可以使用 `DEFINE` 陳述式或 `@DTW_ASSIGN()` 函數，來指定這個變數的值。

**需求：** 您必需在 Net.Data 起始設定檔中定義 `DTW_LOG_DIR` 來起始記載日誌；否則當您在巨集中設定 `DTW_LOG_LEVEL` 變數時，Net.Data 不會記錄訊息。

### 值

`DTW_LOG_LEVEL="OFF|ERROR|WARNING"`

表 22. *DTW\_LOG\_LEVEL* 值

值	說明
OFF	Net.Data 不會記錄錯誤。OFF 是預設值。
ERROR	Net.Data 會記錄錯誤訊息。
WARNING	Net.Data 記錄警告及錯誤訊息。

### 範例

```
%DEFINE DTW_LOG_LEVEL="ERROR"
```

## DTW\_MACRO\_FILENAME

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

現行 Net.Data 巨集檔的名稱與副檔名。

這是一種預設變數，其數值無法修改。使用這種變數作為變數參照。

### 範例

<P>這個 Net.Data 巨集是 <I>\$(DTW\_MACRO\_FILENAME)</I>，  
它是在 \$(DTW\_MACRO\_LAST\_MODIFIED) 時更新。

## DTW\_MACRO\_LAST\_MODIFIED

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

前次修改 Net.Data 巨集的日期和時間。輸出格式是依 Net.Data 執行的系統而定。

這是一種預設變數，其數值無法修改。使用這種變數作為變數參照。

### 範例

<P>這個 Net.Data 巨集是 <I>\$(DTW\_MACRO\_FILENAME)</I>，  
它是在 \$(DTW\_MACRO\_LAST\_MODIFIED) 時更新。



DTW\_MBMODE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X		X	X	X

目的

提供多位元字集支援給預設語言環境所使用的字串及字組函數。您可以在起始設定檔中設定這個變數，但也可以在巨集中使用它來設定或置換現行的設定。

使用 `DEFINE` 陳述式或以 `@DTW_ASSIGN()` 函數來設定這個變數值。

您可以在起始設定檔中設定這個變數，但也可以在巨集中使用它來設定或置換現行的設定。

**OS/400 的使用者請注意：** `Net.Data for OS/400` 會自動啓用 `MBCS` 支援的函數，所以不需要這個變數。`Net.Data for OS/400` 不處理被移到 `OS/400` 作業系統的巨集內所含的此變數。

這個架構變數使用 `DTW_UNICODE` 架構變數。若 `DTW_UNICODE` 使用預設值 `NO`，則使用 `DTW_MBMODE` 的值。若 `DTW_UNICODE` 設定的值不是 `NO`，則使用其值。表23 說明這兩個變數決定內建函數處理字串的方式：

表 23. `DTW_UNICODE` 與 `DTW_MBMODE` 設定之間的關係

若 <code>DTW_UNICODE</code> 設定為	若 <code>DTW_MBMODE=YES</code>	若 <code>DTW_MBMODE=NO</code>
...		
NO	支援 DBCS 混合 SBCS	僅支援 SBCS
UTF8	支援 UTF-8	支援 UTF-8

值

`DTW_MBMODE="YES"|"NO"`

表 24. `DTW_MBMODE` 值

值	說明
YES	指定 <code>MBCS</code> 支援給字串及字組函數
NO	指定字串及字組函數不要有 <code>MBCS</code> 支援。 <code>NO</code> 是預設值。

範例

範例 1：置換 `INI` 檔案中的值

`INI` 檔：

`DTW_MBMODE NO`

巨集：

`%DEFINE DTW_MBMODE = "YES"`

## DTW\_MP\_PATH

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

Net.Data 可執行檔的路徑與名稱。依照不同的系統，其輸出路徑與名稱看起來會像以下所示的樣本：

```
/usr/lpp/internet/server_root/cgi-bin/db2www
```

這是一種預設變數，其數值無法修改。使用這種變數作為變數參照。

### 範例

Net.Data 可執行檔是 `$(DTW_MP_PATH)`。

## DTW\_MP\_VERSION

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

在伺服器上執行之 Net.Data 的版本和版次號碼。

這是一種預設變數，其數值無法修改。使用這種變數作為變數參照。

### 範例

這個 Web 應用程式是使用 \$(DTW\_MP\_VERSION)。

## DTW\_PRINT\_HEADER

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定 HTTP 表頭的文字。

您必須先設定這個變數，才能讓 Net.Data 處理任何傳到 Web 瀏覽器的文字，因為 Net.Data 會在顯示文字之前，先讀取這個變數，之後便不會再察看它。一旦 Net.Data 將文字傳給瀏覽器之後，任何對 DTW\_PRINT\_HEADER 變數所做的變更，一律不予處理。

如果您使用 DTW\_PRINT\_HEADER 製作自己的表頭 (DTW\_PRINT\_HEADER="NO")，則您必需設定 DTW\_REMOVE\_WS="NO"。

使用 DEFINE 陳述式或以 @DTW\_ASSIGN() 函數來設定這個變數值。

### 值

DTW\_PRINT\_HEADER="YES"|"NO"

表 25. DTW\_PRINT\_HEADER 值

值	說明
YES	Net.Data 列出 HTTP 表頭的文字 Content-type: text/html。YES 是預設值。
NO	Net.Data 不要列出 HTTP 表頭。您可以製作自定的 HTTP 表頭資訊。

### 範例

這個變數最常見的用法，是啓用 Net.Data 巨集來傳送 cookies。要設定 cookies，必須將 DTW\_PRINT\_HEADER 變數設為 NO，而前三行必須是 Content-type 表頭、Set-Cookie 陳述式、然後是一個空行。

#### 範例 1：啓用 Net.Data 傳送 cookie

```
%DEFINE DTW_PRINT_HEADER="NO"

%HTML(cookie1) {
Content-type: text/html
Set-Cookie: UsrId=56, expires=Friday, 12-Dec-99, 12:00:00 GMT; path=/

<P>
任何文字
%}
```

## DTW\_REMOVE\_WS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

藉由壓縮定位、白色空間及換行字元引起的額外空間，減少動態產生的 Web 網頁大小。

在 DEFINE 陳述式中指定這個變數的值。

**使用<PRE></PRE>標籤：**將這個變數定義為 YES 時，會影響列印出來的白色空間數量和類型。如果將該變數設定為 YES，則使用 <PRE></PRE> 標籤的 HTML 頁面會有一部份可能無法如預期地顯示。

如果您使用 DTW\_PRINT\_HEADER 製作自己的表頭 (DTW\_PRINT\_HEADER="NO")，您必需設定 DTW\_REMOVE\_WS="NO"。

**OS/390 使用者：**在起始設定檔案中設定這個變數，來指定一個值給您所有的巨集。您可以在巨集中定義這個值，將它置換。如果巨集中沒有定義 DTW\_REMOVE\_WS，則使用起始設定檔案中的值。

### 值

DTW\_REMOVE\_WS="YES"|"NO"

表 26. DTW\_REMOVE\_WS 值

值	說明
YES	Net.Data 壓縮兩格以上連續的白色空間，成為一個換行字元，以產生較少的 HTML 結果網頁。
NO	Net.Data 不會壓縮白色空間。NO 是預設值。

### 範例

**範例 1：**壓縮白色空間

DTW\_REMOVE\_WS="YES"

## RETURN\_CODE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

由一呼叫傳回給內建函數或語言環境的回覆碼。Net.Data 是使用此值來處理 MESSAGE 區塊。您可以使用這個變數，來判斷函數呼叫的結果是成功還是失敗。如果其值為零，表示函數呼叫已經順利完成。

您可在 Net.Data 巨集中的任何部分參照 RETURN\_CODE 變數。

這個值是一個預設值；建議您不要修改它。將它作為一個變數參照使用。

### 範例

**範例 1：**告知函數是否順利完成的訊息

```
@function1()
%IF ("$(RETURN_CODE)" == "0")
    函數已經順利完成。
%ELSE
    函數失敗，其回覆碼為 $(RETURN_CODE)。
%ENDIF
```

**範例 2：**當回覆碼為 0 時的預設訊息

```
%MESSAGE{
default: "<h2>Net.Data 收到回覆碼： $(RETURN_CODE)</h2>" : continue
%}
```

如果函數所傳回的回覆碼不是 0，則畫面會出現預設的訊息。

---

## 第3章 Net.Data 內建函數

Net.Data 提供您在沒有建立您自己的 FUNCTION 區塊狀況下，即可立即使用廣泛的函數。Net.Data 內建函數分為下列種類：

- **一般目的函數**可協助您使用 Net.Data 來開發網頁，且不適用於其它的種類。請參閱第114頁的『一般函數』。
- **算術函數**執行算術作業，請參閱第144頁的『算術函數』。
- **字串操作函數**修改字串與字元。請參閱第162頁的『字串函數』。
- **字組操作函數**修改字組或字組集。請參閱第188頁的『字組函數』。
- **表格操作函數**幫助您從您的表格資料產生套表與報告。請參閱第199頁的『表格函數』。
- **純本文檔介面函數**執行檔案輸入與輸出。請參閱第238頁的『純本文檔介面函數』。
- **Web 登記函數**在 Web 登記上執行作業。請參閱第262頁的『Web 登記函數』。
- **持續巨集函數**支援 Net.Data 中處理的異動。請參閱第280頁的『持續巨集函數』。

雖然某些函數參數被描述為具有整數或浮點數類型，但這些詞彙是分別用來表示整數或浮點數值的字串。

---

### 函數名稱

Net.Data 內建函數均以 DTW 開始，這是保留的字首。使用者定義函數不可使用這個字首。

對不是 Net.Data 內建函數的函數使用 DTW 字首可能會造成無法預期的行為。

內建函數名稱並不區分大小寫。

---

### 輸入與輸出參數

函數可以有參數傳遞規格，決定 Net.Data 使用參數作為輸入、輸出或輸入與輸出兩者。這些參數的傳遞規格是以下列關鍵字指定：

**IN** 指定參數從 Net.Data 傳遞輸入資料到語言環境。

**OUT** 指定參數從語言環境傳回輸出資料到 Net.Data。

**INOUT**

指定參數傳遞輸入資料到語言環境並從語言環境傳回輸出資料到 Net.Data。

---

### 函數結果格式

許多函數具有一種或多種下列的格式：

- 以 DTW\_r、DTWF\_r、和 DTWR\_r 作為開頭的函數，將它們的結果傳回至函數呼叫，所以它們並不會擁有輸出參數。這個範例會顯現伺服器的時間：  
現行區域時間是 @DTW\_rTIME()。

- 以 DTW\_m 作為開頭的函數是以多重參數執行的函數。每個參數都是輸入參數也都是輸出參數。函數針對參數執行，而其結果也透過參數傳回。這個範例會將三個輸入參數轉換為大寫字母，以使其看起來整齊劃一：

```
@DTW_mUPPERCASE(model, style, shipNo)
第 $(shipNo) 批出貨包含 $(quantity) 個機型 $(model) $(style)。
```

- 其它以 DTW\_、DTWF\_、和 DTWR\_ 作為開頭的函數，會將它們的結果以一個輸出參數來傳回。您必須指定輸出參數。這個範例會顯現伺服器的時間：

```
@DTW_TIME(nowTime)
現行區域時間是 $(nowTime)。
```

---

## 函數參數規則

將函數參數以正確的次序放置。在設定最後一個輸入參數之前，您必須指定所有輸入參數，或指定為空值（『』）表示接受預設值。例如，您可以仿照下列範例來呼叫 DTW\_TB\_INPUT\_TEXT：

```
@DTW_TB_INPUT_TEXT(myTable, "1", "2", "", "", "32")
```

在上面的範例中，第四個和第五個參數使用預設值。將它們併入為空字元以表示在所產生的 HTML 中，『32』是 MAXLENGTH 的值。最終的參數未指定，則所使用的是預設值。如果您選擇接受使用預設值來當作 MAXLENGTH 以及之前的兩個參數的話，就請將它們省略，如下面所示：

```
@DTW_TB_INPUT_TEXT(myTable, "1", "2")
```

當後續有非空的輸入參數存在時，您必須在參數列示中指定中間的空值給輸入參數。在您指定您最終輸出參數之前，您無須指定任何輸入參數為中間的空值。

---

## 一般函數

一般函數可協助您使用 Net.Data 來開發網頁，且不適用於其它的種類。下列函數都是一般目的函數：

- 第116頁的『DTW\_ADDQUOTE』
- 第118頁的『DTW\_CACHE\_PAGE』
- 第121頁的『DTW\_DATE』
- 第123頁的『DTW\_EXIT』
- 第124頁的『DTW\_GETCOOKIE』
- 第126頁的『DTW\_GETENV』
- 第128頁的『DTW\_GETINIDATA』
- 第129頁的『DTW\_HTMLENCODE』
- 第131頁的『DTW\_QHTMLENCODE』
- 第132頁的『DTW\_SENDMAIL』
- 第136頁的『DTW\_SETCOOKIE』
- 第139頁的『DTW\_SETENV』



- 第140頁的『DTW\_TIME』
- 第142頁的『DTW\_URLESCSEQ』

# DTW\_ADDQUOTE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

## 目的

將在輸入字串中的單引號置換為兩個單引號。

## 語法

```
@DTW_ADDQUOTE(stringIn, stringOut)
@DTW_rADDQUOTE(stringIn)
@DTW_mADDQUOTE(stringMult, stringMult2, ..., stringMultn)
```

## 參數

表 27. DTW\_ADDQUOTE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。 DTW_mADDQUOTE 可以有多個輸入字串。
字串	<i>stringOut</i>	OUT	包含 <i>stringIn</i> 已修改格式的變數。
字串	<i>stringMult</i>	INOUT	<ul style="list-style-type: none"><li>在輸入上：包含字串的變數。</li><li>在輸出上：包含輸入字串的一個變數，該字串的每一個單引號 (') 字元都被兩個單引號字元所置換。</li></ul>

## 回覆碼

表 28. DTW\_ADDQUOTE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

## 用法注意事項

對於從 Web 瀏覽器取得輸入的所有 SQL INPUT 陳述式使用這個函數。例如，如果您如下列範例一般，將 O'Brien 輸入為姓氏，則單引號將會造成錯誤：

```
INSERT INTO USER1.CUSTABLE (LNAME, FNAME)
VALUES ('O'Brien', 'Patrick')
```

使用 DTW\_ADDQUOTE 函數會變更 SQL 陳述式並防止錯誤發生：

```
INSERT INTO USER1.CUSTABLE (LNAME, FNAME)
VALUES ('O''Brien', 'Patrick')
```

## 範例

範例 1：在 OUT 參數新增一個額外的單引號

```
@DTW_ADDQUOTE(string1,string2)
```

- 輸入：string1="John's Web page"
- 傳回：string2="John''s Web page"

**範例 2：**在函數呼叫的傳回值上新增額外的單引號

```
@DTW_rADDQUOTE("The title of the article is 'Once upon a time'")
```

- 傳回："The title of the article is ''Once upon a time''"

**範例 3：**在函數呼叫的每一個 INOUT 參數上新增額外的單引號

```
@DTW_mADDQUOTE(string1,string2)
```

- 輸入：string1="Joe's bag", string2="''to be or not to be'"
- 傳回：string1="Joe''s bag", string2="''to be or not to be''"

**範例 4：**在插入 DB2 表格的資料中插入額外的單引號

```
%FUNCTION(DTW_SQL) insertName(){
INSERT INTO USER1.CUSTABLE (LNAME,FNAME)
VALUES ('@DTW_rADDQUOTE(lastname)', '@DTW_rADDQUOTE(firstname)')
%}
```

- 輸入：lastname="O'Brien", firstname="Patrick"
- 傳回："O''Brien", "Patrick"

## DTW\_CACHE\_PAGE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X								

### 目的

快取因巨集處理結果而產生的部份或全部網頁。

### 語法

@DTW\_CACHE\_PAGE(cacheid, url, age, status)

### 參數

表 29. DTW\_CACHE\_PAGE 參數

參數	使用	說明
<i>cache_id</i>	IN	一個字串變數用來識別頁面將被放置的快取記憶體。
<i>cached_page_ID</i>	IN	一個包含識別字的字串變數，在後續的 DTW_CACHE_PAGE 快取要求中，用來指定快取頁面的位置。字串可以是一個 URL。
<i>age</i>	IN	<p>一個字串變數，包含了以秒為單位的時間長度。這個參數決定一個頁面是否過期。如果頁面已經比 <i>age</i> 還要舊時，此頁面就不會被傳送到瀏覽器。</p> <p>如果 <i>age</i> 被設定為 -1，且頁面存在快取記憶體中，則不論其經歷時間，Net.Data 會直接從快取記憶體中將它傳送到 Web 瀏覽器。Net.Data 不會置換快取記憶體中的頁面。</p>
<i>status</i>	OUT	<p>顯示快取頁面狀態的字串變數。可能的值是以小寫顯示：</p> <ul style="list-style-type: none"><li>• <b>ok</b>：當巨集執行終止時，輸出頁面會被送到快取記憶體。</li><li>• <b>new</b>：頁面不在快取記憶體中。</li><li>• <b>renew</b>：頁面在快取記憶體中，但已過期。</li><li>• <b>no_cache</b>：所指定的快取記憶體識別字不存在。必須在快取記憶體架構檔中定義。您的巨集可以繼續執行，無須快取頁面。</li><li>• <b>inactive</b>：您指定的快取記憶體已被標示為非作用中。您的巨集可以繼續執行，無須快取頁面。</li><li>• <b>busy</b>：在這次執行中，您的巨集已經曾發出 DTW_CACHE_PAGE 內建函數。您的巨集可以繼續執行。</li><li>• <b>error</b>：嘗試與快取記憶體通信時發生錯誤。</li></ul>

### 回覆碼

表 30. DTW\_CACHE\_PAGE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。

表 30. DTW\_CACHE\_PAGE 回覆碼 (繼續)

回覆碼	說明
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

## 用法注意事項

1. 當 DTW\_CACHE\_PAGE() 被呼叫時，它會嘗試從快取記憶體取回指定的頁面，並將該頁面傳送到 Web 瀏覽器，就如同它是由巨集所產生的輸出頁面一樣。如果找到頁面且該頁面未過期，Net.Data 會停止巨集的處理、跳出巨集、並將快取的頁面傳送到 Web 瀏覽器。

如果所要求的頁面不在快取記憶體中，或現有的快取頁面已經大於 *age* 值，Net.Data 會產生新的輸出頁面。當巨集順利的完成後，Net.Data 會將新頁面傳送到瀏覽器，並同時快取頁面。

2. 對於大部份快取應用程式而言，在巨集頂端設定 DTW\_CACHE\_PAGE()，以便快取執行巨集時產生的所有網頁。這個方法可在更新巨集時更容易維護巨集。例如，當函數位於巨集的中間，一個 HTML 報告區段在較早時加入巨集中，可能會被忽略。Net.Data 將不會快取新報告輸出。此外，這個方法可以增進執行效能，因為當 Net.Data 決定頁面已快取時，它會停止所有進一步的處理程序。

對於進階快取應用程式而言，當您需要在處理過程中決定快取的特定位置時，您可以將函數放置在巨集的特定位置，而不是巨集開頭。例如，您可能需要基於在查詢或函數呼叫後要傳回多少列數而來作快取的決定。

## 範例

**範例 1：**將 DTW\_CACHE\_PAGE() 函數放置在巨集的開頭，以擷取所有 HTML 輸出

```
%IF (customer_status == "Classic")
  @DTW_CACHE_PAGE("mymacro.mac", "http://www.mypage.org", "-1", status)
%ENDIF
% DEFINE { ...%}

...

%HTML (OUTPUT) {
  <title>這是頁面標題
</head>
  <body>
    <center>
      這是主標題
    <p>現在時刻 $(time)。祝您有愉快的一天！
    </body>
  </html>

%}
```

**範例 2：**將函數放置在 HTML 區塊中，因為快取的決定取決於 HTML 輸出的預期大小

```
%DEFINE { ...%}

...
```

```

%FUNCTION(DTW_SQL) count_rows(){
    從客戶選取 count(*)
%REPORT{
%ROW{
    @DTW_ASSIGN(ALL_ROWS, V1)
%}
%}
%}

%FUNCTION(DTW_SQL) all_customers(){
    select * from customer
%}

%HTML (OUTPUT) {
    <html>
    <head>
    <title>這是客戶列表
    </head>
    <body>

@count_rows()

    %IF ($(ALL_ROWS) > "100")
    @DTW_CACHE_PAGE("mymacro.mac", "http://www.mypage.org", "-1", status)
%ENDIF

@all_customers()

    </body>
    </html>
%}

```

在這個範例中，頁面的快取或取回是基於 HTML 輸出的預期大小。只有當資料庫表格內容超過 100 列時，HTML 輸出頁才值得快取。執行巨集後，Net.Data 總是將 OUTPUT 區塊中的文字，這是客戶列表，傳送到瀏覽器；文字從不快取。當 IF 區塊的條件滿足時，@count\_rows() 函數呼叫的下幾行，會被快取或取回。兩者一同形成完整的 Net.Data 輸出頁面。

### 範例 3：動態取回快取記憶體 ID 與快取頁面 ID

```

%HTML (OUTPUT) {
    %IF (customer == "Joe Smith")

@DTW_CACHE_PAGE(@DTW_rGETENV("DTW_MACRO_FILENAME"), @DTW_rGETENV("URL"), "-1", status)

%ENDIF

...

    <html>
    <head>
    <title>這是頁面標題</title>
    </head>
    <body>
    <center>
    <h3>這是主標題</h3>
    <p>今天是 @DTW_rDATE()。祝您有愉快的一天！
    </body>
    </html>

%}

```

## DTW\_DATE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

在所指定的格式中傳回現行系統日期。

### 語法

@DTW\_DATE(format, stringOut)

@DTW\_DATE(stringOut)

@DTW\_rDATE(format)

@DTW\_rDATE()

### 參數

表 31. DTW\_DATE 參數

資料類型	參數	使用	說明
字串	<i>format</i>	IN	一個指定資料格式的變數或文字字串。有效格式包括： D - 一年中的第幾天 (001-366) E - 歐式日期格式 (dd/mm/yy) N - 一般日期格式 (dd mon yyyy) O - 照順序的日期格式 (yy/mm/dd) S - 標準日期格式 (yyyymmdd) U - 美式日期格式 (mm/dd/yy) 預設值是 N。
字串	<i>stringOut</i>	OUT	一個包含具有所指定格式之日期的變數。

### 回覆碼

表 32. DTW\_DATE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：一般日期格式

```
@DTW_DATE(results)
```

- 傳回：results = "25 Apr 1997"

**範例 2：**歐洲日期格式

```
@DTW_DATE("E", results)
```

- 傳回：results="25/04/97"

**範例 3：**美式日期格式

```
%HTML(report){  
<P>這個報告建立於 @DTW_rDATE("U")。
```

- 傳回：04/25/97



## DTW\_EXIT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定立即離開巨集。Net.Data 會傳送 DTW\_EXIT() 被呼叫之前所產生的任何網頁到 Web 瀏覽器。

### 語法

@DTW\_EXIT()

### 回覆碼

表 33. DTW\_EXIT 回覆碼

回覆碼	說明
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。

### 用法注意事項

1. 使用 DTW\_EXIT() 立即停止巨集處理。這個方法可節省 Net.Data 處理整個檔案所花的時間。
2. 確定在新增 DTW\_EXIT() 函數之前，整個巨集的語法是正確的。使用 DTW\_EXIT() 會使 Net.Data 在遇到此函數的呼叫時停止處理巨集，這可避免您發覺在處理完 DTW\_EXIT() 函數之後發生的錯誤。

### 範例

#### 範例 1：跳出巨集

```
%HTML(cache_example) {  
  
    <html>  
    <head>  
    <title>這是頁面標題</title>  
    </head>  
    <body>  
    <center>  
    <h3>這是主標題</h3>  
    <!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>  
    < ! Joe Smith 看到一篇非常短的頁面 ! >  
    <!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>  
    %IF (customer == "Joe Smith")  
  
@DTW_EXIT()  
  
%ENDIF  
  
...  
  
    </body>  
    </html>  
    %}
```

## DTW\_GETCOOKIE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回所指定 cookie 的值。

### 語法

@DTW\_GETCOOKIE(IN cookie\_name, OUT cookie\_value)

@DTW\_rGETCOOKIE(IN cookie\_name)

### 參數

表 34. DTW\_GETCOOKIE 參數

資料類型	參數	使用	說明
字串	cookie_name	IN	指定 cookie 名稱的變數或文字字串。
字串	cookie_value	OUT	包含經由函數取回之 cookie 值的變數，例如使用者狀態資訊。  <b>OS/400 與 OS/390 使用者：</b> 如果 cookie 值具有 URL 樣式編碼 (例如 "%20")，則 cookie 值會在傳回之前被解碼。  <b>工作站使用者：</b> 如果 cookie 值具有 URL 樣式編碼 (例如 "%20")，則 cookie 值不會在傳回之前被解碼。

### 回覆碼

表 35. DTW\_GETCOOKIE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
8000	找不到 cookie。

### 用法注意事項

在兩個不同的 HTTP 要求中定義並取回 cookie。因為 cookie 只能在被傳送到從屬站以後才可看見，如果巨集嘗試去獲得定義於相同 HTTP 要求的 cookie，您可能會接收到不預期的結果。

### 範例

**範例 1：**取回包含使用者 ID 和密碼資訊的 cookie

```
@DTW_GETCOOKIE("mycookie_name_for_userID", userID)
@DTW_GETCOOKIE("mycookie_name_for_password", password)
```

**範例 2：**在收集使用者資訊之前，決定使用者 cookie 是否存在

```
%MESSAGE {
    8000 : "" : continue
%}

%HTML(welcome) {
    <html>
    <body>
    <h1>Net.Data 俱樂部</h1>
    @DTW_GETCOOKIE("NDC_name", name)
    %IF ($(RETURN_CODE) == "8000") %{ 找不到 cookie。%}
    <form method="post" action="remember">
    <p>歡迎蒞臨本會。請輸入您的姓名。<br>
    <input name="name">
    <input type="submit" value="submit"><br>
    </form>
    %ELSE
    <p>$(name) 您好，歡迎再度光臨。
    %ENDIF
    </body>
    </html>
    %}
```

HTML 歡迎使用區段會檢查 cookie NDC\_name 是否存在。如果 cookie 存在，瀏覽器會顯示一個個人化的歡迎。如果 cookie 不存在，套表會要求輸入使用者姓名，並將它置於 HTML remember 區段，如此將使用者名稱設定成如下的 cookie NDC\_name：

```
%HTML(remember) {
    <html>
    <body>
    <H1>Net.Data 俱樂部</H1>
    @DTW_SETCOOKIE("NDC_name", name, "expires=Wednesday, 01-Dec-2010 00:00:00;path=/")
    <p>感謝您。
    <p><a href="welcome">Come back</a>
    </body>
    </html>
    %}
```

## DTW\_GETENV

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將所指定之環境變數的值傳回。

### 語法

@DTW\_GETENV(envVarName, envVarValue)

@DTW\_rGETENV(envVarName)

### 參數

表 36. DTW\_GETENV 參數

資料類型	參數	使用	說明
字串	<i>envVarName</i>	IN	一個變數或文字字串。
字串	<i>envVarValue</i>	OUT	在 <i>envVarName</i> 上所指定的環境變數值。 若該值找不到的話將會傳回一個空字串。

### 回覆碼

表 37. DTW\_GETENV 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

您亦可使用 ENVVAR 陳述式來參考環境變數的值。有關詳細資訊，請參閱第13頁的『ENVVAR 陳述式』。

### 範例

**範例 1：**透過 OUT 參數傳回 PATH 陳述式的值

```
@DTW_GETENV(myEnvVarName, myEnvVarValue)
```

- 輸入：myEnvVarName = "PATH"
- 傳回：myEnvVarValue = "/usr/bin"

**範例 2：**傳回 PATH 陳述式的值

```
@DTW_rGETENV(myPath)
```

- 輸入：myPath = "PATH"
- 傳回："/usr/bin"

**範例 3：**傳回伺服器通信協定的值

伺服器是 @DTW\_rGETENV("SERVER\_PROTOCOL")。

- 傳回："HTTP/1.0"

## DTW\_GETINIDATA

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回所指定之架構變數的值。

### 語法

@DTW\_GETINIDATA(iniVarName, iniVarValue)

@DTW\_rGETINIDATA(iniVarName)

### 參數

表 38. DTW\_GETINIDATA 參數

資料類型	參數	使用	說明
字串	<i>iniVarName</i>	IN	一個變數或文字字串。
字串	<i>iniVarValue</i>	OUT	在 <i>iniVarName</i> 中所指定的架構變數值。

### 回覆碼

表 39. DTW\_GETINIDATA 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

1. 如果指定的架構變數不是架構檔，則 Net.Data 會傳回空字串。
2. 對於 **OS/390**、**OS/2**、**Windows NT** 及 **UNIX** 使用者： 架構路徑變數 (MACRO\_PATH、EXEC\_PATH 和 INCLUDE\_PATH)，以及 ENVIRONMENT 陳述式，無法由這個呼叫取回。
3. 對於 **OS/400** 使用者： ENVIRONMENT 陳述式無法由這個呼叫取回。

### 範例

範例 1：傳回 Net.Data 路徑變數值

@DTW\_GETINIDATA(myEnvVarName, myEnvVarValue)

- 輸入：myEnvVarName = "FFI\_PATH"
- 傳回：myEnvVarValue = "D:\FFI"

## DTW\_HTMLENCODE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

利用 HTML 字元跳出碼來編寫選取的字元。

### 語法

@DTW\_HTMLENCODE(stringIn, stringOut)

@DTW\_rHTMLENCODE(stringIn)

### 參數

表 40. DTW\_HTMLENCODE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含已修改輸入字串的變數，該輸入字串中的某些字元已被 HTML 字元跳出碼所置換。

### 回覆碼

表 41. DTW\_HTMLENCODE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

1. 您可以使用這個函數，來編寫您不希望 Web 瀏覽器解譯為 HTML 的字元資料。例如，藉由使用適當的跳出碼，您可以在網頁內顯示一些字元，例如小於 (<) 及大於 (>) 符號，否則這些字元會被瀏覽器解譯為 HTML 標籤的元件。
2. 表42 顯示由 DTW\_HTMLENCODE 函數所編碼的字元。

表 42. HTML 的字元跳出碼

字元	名稱	程式碼
SPACE	空格	&#32;
"	雙引號	&#34;
#	號碼記號	&#35;
%	百分比	&#37;
&	& 記號	&#38;
[	左方括弧	&#40;
]	右方括弧	&#41;
+	加	&#43;

表 42. *HTML* 的字元跳出碼 (繼續)

\	斜線	&#47;
:	冒號	&#58;
;	分號	&#59;
<	小於	&#60;
=	等於	&#61;
>	大於	&#62;
?	問號	&#63;
@	At 記號	&#64;
/	反斜線	&#92;
^	Carat	&#94;
{	左大括弧	&#123;
	直線	&#124;
}	右大括弧	&#125;
~	波型符號	&#126;

## 範例

### 範例 1：對空格字元編碼

```
@DTW_HTML_ENCODE(string1,string2)
```

- 輸入：string1 = "Jim's dog"
- 傳回：string2 = "Jim's&#32;dog"

### 範例 2：對空格、小於符號與等號編碼

```
@DTW_rHTML_ENCODE("X <= 10")
```

- 傳回："X&#32;&#60;&#61;&#32;10"



## DTW\_QHTMLENCODE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

執行與 @DTW\_HTMLENCODE 相同的功能，但亦將單引號字元 (') 編寫為 &#39;。第 129 頁的表 42 顯示 DTW\_QHTMLENCODE 所使用的 HTML 字元跳出碼。

### 語法

@DTW\_QHTMLENCODE(stringIn, stringOut)

@DTW\_rQHTMLENCODE(stringIn)

### 參數

表 43. DTW\_QHTMLENCODE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含已修改格式的 <i>stringIn</i> 的變數，其中有某些字元已被 HTML 字元跳出碼所置換。

### 回覆碼

表 44. DTW\_QHTMLENCODE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

**範例 1：**對單引號與空格編碼

```
@DTW_QHTMLENCODE(string1,string2)
```

- 輸入：string1 = "Jim's dog"
- 傳回：string2 = "Jim&#39;s&#32;dog"

**範例 2：**對單引號、空格與 & 符號編碼

```
@DTW_rQHTMLENCODE("John's & Jane's")
```

- 傳回："John&#39;s&#32;&#38;&#32;Jane&#39;s"

## DTW\_SENDMAIL

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

動態建立與傳輸電子郵件（e-mail）訊息。

### 語法

@DTW\_SENDMAIL(IN Sender, IN Recipient, IN Message, IN Subject, IN CarbonCopy, IN BlindCarbonCopy, IN ReplyTo, IN Organization)

@DTW\_SENDMAIL(IN Sender, IN Recipient, IN Message, IN Subject, IN CarbonCopy, IN BlindCarbonCopy, IN ReplyTo)

@DTW\_SENDMAIL(IN Sender, IN Recipient, IN Message, IN Subject, IN CarbonCopy, IN BlindCarbonCopy)

@DTW\_SENDMAIL(IN Sender, IN Recipient, IN Message, IN Subject, IN CarbonCopy)

@DTW\_SENDMAIL(IN Sender, IN Recipient, IN Message, IN Subject)

@DTW\_SENDMAIL(IN Sender, IN Recipient, IN Message)

### 參數

表 45. DTW\_SENDMAIL 參數

資料類型	參數	使用	說明
字串	<i>sender</i>	IN	指定作者位址的變數或文字字串。這個參數是必要的。有效的格式為： <ul style="list-style-type: none"><li>• 姓名 &lt;使用者 @ 領域&gt;</li><li>• &lt;使用者 @ 領域&gt;</li><li>• 使用者 @ 領域</li></ul>
字串	<i>recipient</i>	IN	指定收件者的電子郵件位址的變數或文字字串。藉由逗點（,）的分隔，這個值可以包含多重收件者。這個參數是必要的。有效的收件者格式為： <ul style="list-style-type: none"><li>• 姓名 &lt;使用者 @ 領域&gt;</li><li>• &lt;使用者 @ 領域&gt;</li><li>• 使用者 @ 領域</li></ul>
字串	<i>message</i>	IN	包含電子郵件訊息本文的變數或文字字串。這個參數是必要的。
字串	<i>subject</i>	IN	包含主旨行文字的變數或文字字串。這是一個可選用的參數。您必須指定一個空字串（""）才能指定額外的參數。
字串	<i>CarbonCopy</i>	IN	包含額外收件者之電子郵件位址、或姓名與電子郵件位址的變數或文字字串。藉由逗點（,）的分隔，這個值可以包含額外收件者。請參閱 <i>Recipient</i> 參數以取得有效的收件者格式。這是一個可選用的參數。您必須指定一個空字串（""）才能指定額外的參數。

表 45. DTW\_SENDMAIL 參數 (繼續)

資料類型	參數	使用	說明
字串	<i>BlindCarbonCopy</i>	IN	包含額外收件者之電子郵件位址、或姓名與電子郵件位址的變數或文字字串，但收件者不會出現在電子郵件表頭。藉由逗點 (,) 的分隔，這個值可以包含額外收件者。請參閱 <i>Recipient</i> 參數以取得有效的收件者格式。這是一個可選用的參數。您必須指定一個空字串 ("" ) 才能指定額外的參數。
字串	<i>ReplyTo</i>	IN	包含回覆此訊息要送往寄件者電子郵件的變數或文字字串。這是一個可選用的參數。您必須指定一個空字串 ("" ) 才能指定額外的參數。有效的回覆格式為： <ul style="list-style-type: none"> <li>• 姓名 &lt;使用者 @ 領域&gt;</li> <li>• &lt;使用者 @ 領域&gt;</li> <li>• 使用者 @ 領域</li> </ul>
字串	<i>Organization</i>	IN	包含寄件者組織名稱的變數或文字字串。這是一個可選用的參數。

## 回覆碼

表 46. DTW\_SENDMAIL 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
7000	Net.Data 無法連接指定的 SMTP 伺服器。
7001	Net.Data 嘗試將電子郵件訊息轉遞至指定的 SMTP 伺服器時，發生 SMTP 錯誤。
7002	指定的 SMTP 伺服器不支援 ESMTP (Extended Simple Mail Transfer Protocol) 。

## 用法注意事項

1. 您可以使用可選用的架構變數 DTW\_SMTP\_SERVER，指定 SMTP 伺服器來傳輸電子郵件訊息。這個參數值可以是主電腦名稱、IP 位址或節點與名稱。當這個變數沒有被定義，Net.Data 會使用區域主電腦以作為 SMTP 伺服器。若想進一步了解此變數，請參閱您作業系統的 Net.Data 管理和程式設計指南內的架構這一章。
2. **OS/400、OS/2、Windows NT 及 UNIX 使用者：**標準 SMTP (Simple Mail Transfer Protocol) 伺服器只接受 7 位元資料，如 U.S. ASCII 字元。如果您的訊息有 8 位元字元，建議您指定 ESMTP 伺服器，ESMTP 伺服器接受 8 位元字元。Net.Data 不會將 8 位元資料編碼為 7 位元資料。如果您無法存取 ESMTP 伺服器，從電子郵件訊息中除去所有的 8 位元字元。

Net.Data for OS/390 使用者不必修改電子郵件訊息的 SMTP 伺服器。

3. 字集支援：

- **OS/400 使用者：**您可以使用可選用的架構變數 DTW\_SMTP\_CHARSET，指定將訊息從 EBCDIC 轉換為 ASCII 時，要使用哪一個 ASCII 字集。如果未指定 DTW\_SMTP\_CHARSET，則預設字集為 iso-8859-1。有關這個變數及支援的字集之詳細資訊，請參閱 OS/400 版 *Net.Data* 管理及程式設計指南 內的架構這一章。
- **OS/2、Windows NT 及 UNIX 使用者：**表47說明所支援的字集：

表 47. *Net.Data* 支援的字集

語言環境	字集	OS/2 或 UNIX 字碼頁	Windows NT 字碼頁
美國, 西歐	"iso-8859-1"	819	1252
日文	"x-sjis"	943	932
中文 (簡體)	"gb2312"	1381	936
韓文	"euc-kr"	970	949
中文 (繁體)	"big5"	950	950

4. 下列列示說明 *Net.Data* 不傳送電子郵件訊息的條件：

- 無法聯繫指定的 SMTP 伺服器。
- 指定的 SMTP 伺服器不支援 ESMTP (Extended Simple Mail Transfer Protocol)，但指定的電子郵件卻包含非美式 ASCII 字集。

## 範例

**範例 1：**建立並傳送一個簡單電子郵件訊息的函數呼叫

```
@DTW_SENDMAIL("<ibmuser1@ibm.com>", "<ibmuser2@ibm.com>", "在 9:30 有一個會議。", "狀態會議")
```

DTW\_SENDMAIL 函數傳送電子郵件訊息時包含下列資訊：

```
日期：Mon, 3 Apr 1998 09:54:33 PST
收件者：<ibmuser2@ibm.com>
寄件者：<ibmuser1@ibm.com>
主旨：狀態會議
```

在 9:30 有一個會議。

日期的資訊是使用系統日期和時間函數建構而成，並以 SMTP 特定資料格式呈現。

**範例 2：**建立並傳送多重收件者、副本與密件副本收件者與公司名稱的電子郵件訊息的函數呼叫

```
@DTW_SENDMAIL("IBM User 1 <ibmuser1@ibm.com>", "IBM User 2 <ibmuser2@ibm.com>,"
"IBM User 3 <ibmuser3@ibm.com>," "IBM User 4 <ibmuser4@ibm.com>","There is a meeting"
"at 9:30.", "Status meeting", "IBM User 5 <ibmuser5@ibm.com>",""
"IBM User 6 <ibmuser6@ibm.com>","meeting@ibm.com", "IBM")
```

DTW\_SENDMAIL 函數傳送電子郵件訊息時包含下列資訊：

```
日期：Mon, 3 Apr 1998 09:54:33 PST
收件者：IBM 使用者 2 <ibmuser2@ibm.com>," IBM 使用者 3 <ibmuser3@ibm.com>,"
IBM 使用者 4 <ibmuser4@ibm.com>
副本：IBM 使用者 5 <ibmuser5@ibm.com>
密件副本：IBM User 6 <ibmuser6@ibm.com>
收件者：IBM 使用者 1 <ibmuser1@ibm.com>
回覆：meeting@ibm.com
```

組織：IBM  
主旨：狀態會議

在 9:30 有一個會議。

### 範例 3：經由 Web 套表介面建立並傳送電子郵件的巨集

```
%HTML(start) {  
  <html>  
  <body>  
  <h1>Net.Data 電子郵件範例</h1>  
  <form method="post" action="sendemail">  
  <p>給：<br><input name="recipient"><p>  
  主旨：<br><input name="subject"><p>  
  訊息：<br><textarea name="message" rows=20 cols=40>  
  </textarea><p>  
  <input type="submit" value="Send E-mail"><br>  
</form>  
</body>  
</html>  
%}  
  
%HTML(sendemail) {  
  <html>  
  <body>  
  <h1>Net.Data 電子郵件範例</h1>  
  @DTW_SENDMAIL("Net.Data E-mail Service <netdata@us.ibm.com>",  
    recipient, message, subject)  
  <p>電子郵件已送出。  
  </body>  
  </html>  
%}
```

這個巨集是經由 Web 套表介面傳送電子郵件。HTML 的啟動區段中顯示一個套表，收件者的電子郵件位址、主旨與訊息都可以鍵入。當使用者按一下**傳送電子郵件**按鈕，訊息就被傳送到 HTML(sendemail) 區段中指定的收件者。這個區段呼叫 DTW\_SENDMAIL，並使用從 Web 套表中獲得的參數來決定電子郵件訊息的內容、與寄件者與收件者。一旦電子郵件訊息傳送出去，便會顯示確認通知。

### 範例 4：使用 SQL 查詢來決定收件者列示的巨集

```
%Function(DTW_SQL) mailing_list(IN message) {  
  SELECT EMAIL_ADDRESS FROM CUSTOMERS WHERE ZIPCODE='CA'  
%REPORT {  
  傳送產品資訊給所有住在加州的客戶...<P>  
  %ROW {  
    @DTW_SENDMAIL("John Doe Corp. <John.Doe@doe.com>", V1, message,  
      "New Product Release")  
    E-mail 已送出給客戶 $(V1)。<BR>  
  %}  
  %}  
%}
```

經由客戶資料庫的 SQL 查詢結果所決定，這個巨集會自動傳送電子郵件訊息到指定的客戶群組。SQL 查詢也取回客戶的電子郵件位址。電子郵件內容是由 *message* 值所決定，而且可以是靜態或動態的（例如您可以使用另一個 SQL 查詢來動態指定產品的版本號碼，或各種報價的價格）。

## DTW\_SETCOOKIE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

產生可在從屬站系統上設定 cookie 的 JavaScript 程式碼。

### 語法

```
@DTW_SETCOOKIE(IN cookie_name, IN cookie_value, IN advanced_options)
```

```
@DTW_SETCOOKIE(IN cookie_name, IN cookie_value)
```

### 參數

表 48. DTW\_SETCOOKIE 參數

資料類型	參數	使用	說明
字串	<i>cookie_name</i>	IN	指定 cookie 名稱的變數或文字字串。
字串	<i>cookie_value</i>	IN	指定 cookie 值的變數或文字字串。  避免在 <i>cookie_value</i> 中使用分號 (;)、逗點 (,) 及空格。 若需要這些特殊字元時，請先使用 Net.Data 函數 DTW_rURLESCSEQ 來處理包含這些特殊字元的字串，再將字串遞送到 DTW_SETCOOKIE。例如，  @DTW_SETCOOKIE("my_cookie_name", @DTW_rURLESCSEQ("my cookie value"))

表 48. DTW\_SETCOOKIE 參數 (繼續)

資料類型	參數	使用	說明
字串	<i>advanced_options</i>	IN	<p>包含可選用屬性的字串，由分號所隔開，可用來定義 cookie。這些屬性可以是：</p> <p><b>到期 = 日期</b> 指定一個定義 cookie 有效生命週期的日期字串。在到期日以後，便不再儲存或取回 cookie。語法： <i>weekday, DD-month-YYYY HH:MM:SS GMT</i></p> <p>其中：</p> <p><i>weekday</i> 指定工作日的完整名稱。</p> <p><i>DD</i> 指定某月中的日期。</p> <p><i>month</i> 指定月份的三字元縮寫。</p> <p><i>YYYY</i> 指定年份的四字元號碼。</p> <p><i>HH:MM:SS</i> 以小時、分鐘與秒鐘來指定時間戳記。</p> <p><b>領域 = 領域名稱</b> 指定 cookie 的領域屬性，使用在領域屬性匹配。</p> <p><b>路徑 = 路徑</b> 指定 cookie 是有效的領域中 URL 的子集。</p> <p><b>安全</b> 指定 cookie 只能透過安全的通道傳輸到 HTTPS 伺服器。</p> <p>未指定 <i>secure</i> 選項時，cookie 可以透過非安全的通道傳送。安全選項不需要瀏覽器對 cookie 加密，也不需要它確保包含 DTW_SETCOOKIE 陳述式的 HTML 頁面要透過 SSL 傳輸。</p> <p>有關進階選項的其餘資訊，請參閱網站 <a href="http://home.netscape.com">http://home.netscape.com</a> 上的 Netscape cookie 規格。</p>

## 回覆碼

表 49. DTW\_SETCOOKIE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。

## 用法注意事項

1. 如果從屬站 Web 瀏覽器不支援 Java Script，瀏覽器便不設定 cookie。

2. 因為 DTW\_SETCOOKIE 會產生 Java Script，請勿在 <SCRIPT> 或 <NOSCRIPT> HTML 元素中呼叫 DTW\_SETCOOKIE。
3. 要取回 cookie，請使用 DTW\_GETCOOKIE() 函數。請參閱第124頁的『DTW\_GETCOOKIE』以學習如何定義 cookie。
4. 在兩個不同的 HTTP 要求中定義並取回 cookie。因為 cookie 只能在被傳送到從屬站以後才可看見，如果巨集嘗試去獲得定義於相同 HTTP 要求的 cookie，您可能會接收到不預期的結果。

## 範例

**範例 1：**以「安全」進階選項來定義包含使用者 ID 和密碼資訊

```
@DTW_SETCOOKIE("mycookie_name_for_userID", "User1")
@DTW_SETCOOKIE("mycookie_name_for_password", "sd3dT", "secure")
```

**範例 2：**定義包含到期日進階選項的 cookie

```
@DTW_SETCOOKIE("mycookie_name_for_userID", "User1",
"expires=Wednesday 01-Dec-2010 00:00:00")
@DTW_SETCOOKIE("mycookie_name_for_password",
"sd3dT", "expires=Wednesday, 01-Dec-2010 00:00:00; secure")
```

函數呼叫應在同一行；本範例中分成許多行是受限於版面。

**範例 3：**在收集使用者資訊之前，決定使用者 cookie 是否存在

```
%HTML(welcome) {
  <html>
  <body>
  <h1>Net.Data 俱樂部</h1>
  @DTW_GETCOOKIE("NDC_name", name)
  %IF ($(RETURN_CODE) == "8000") %{ 找不到 cookie。%}
  <form method="post" action="remember">
  <p>歡迎蒞臨本會。請輸入您的姓名。<br>
  <input name="name">
  <input type="submit" value="submit"><br>
  </form>
  %ELSE
  <p>$(name) 您好，歡迎再度光臨。
  %ENDIF
  </body>
  </html>
  %}
```

HTML(welcome) 區段會檢查 cookie NDC\_name 是否存在。如果 cookie 存在，瀏覽器會顯示一個個人化的歡迎。如果 cookie 不存在，瀏覽器會要求使用者輸入名稱，並將它置於 HTML(remember) 區段。這個區段將使用者名稱記錄到 cookie NDC\_name，如下面所顯示：

```
%HTML(remember) {
  <html>
  <body>
  <H1>Net.Data 俱樂部>
  @DTW_SETCOOKIE("NDC_name", name, "expires=Wednesday, 01-Dec-2010 00:00:00;path=/")
  <p>感謝您。
  <p><a href="welcome">Come back</a>
  </body>
  </html>
  %}
```



## DTW\_SETENV

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定一個指定值給一環境變數，並傳回上一個值。

### 語法

```
@DTW_SETENV(envVarName, envVarValue, prevValue)
```

```
@DTW_rSETENV(envVarName, envVarValue)
```

### 參數

表 50. DTW\_SETENV 參數

資料類型	參數	使用	說明
字串	<i>envVarName</i>	IN	一個代表環境變數的變數或文字字串。
字串	<i>envVarValue</i>	IN	一個有指定為環境變數數值的變數或文字字串。
字串	<i>prevValue</i>	OUT	一個包含環境變數之上一個值的變數。

### 回覆碼

表 51. DTW\_SETENV 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

如果找不到先前的環境變數值，則傳回空字串。

### 範例

**範例 1：**傳回前路徑的值

```
@DTW_SETENV("PATH", "myPath", prevValue)
```

- 輸入：envVarName = "PATH", envVarValue = "myPath"
- 傳回：prevValue = "myPreviousPath"

**範例 2：**傳回前路徑的值，並指定 PATH 的值

```
@DTW_rSETENV("PATH", "myPath")
```

- 輸入：envVarName = "PATH", envVarValue = "myPath"
- 傳回："myPreviousPath", PATH = "myPath"

## DTW\_TIME

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

以所指定的格式將現行系統時間傳回。

### 語法

@DTW\_TIME(stringIn, stringOut)

@DTW\_TIME(stringOut)

@DTW\_rTIME(stringIn)

@DTW\_rTIME()

### 參數

表 52. DTW\_TIME 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	指定時間格式的變數或文字字串。有效的格式為： C - 常用時間 (hh:mmAM/PM 以 12 小時時間顯示) L - 區域時間 (hh:mm:ss) N - 正常時間 (hh:mm:ss 以 24 小時時間顯示)；預設值 X - 擴充時間 (hh:mm:ss.ccc 以 24 小時格式顯示，其中 ccc 是指毫秒數) H - 從午夜 12 點起算的小時數 M - 從午夜 12 點起算的分鐘數 S - 從午夜 12 點起算的秒數
字串	<i>stringOut</i>	OUT	一個包含以所指定格式顯示之時間的變數。

### 回覆碼

表 53. DTW\_TIME 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

範例 1：24 小時時間格式

@DTW\_TIME(results)

- 傳回：results = "10:30:53"

**範例 2：**常用時間格式

```
@DTW_TIME("C", results)
```

- 傳回：results = "10:30AM"

**範例 3：**以函數呼叫傳回從午夜起算的分鐘數

```
@DTW_rTIME("M")
```

- 傳回："630"

**範例 4：**以函數呼叫傳回時間預設值與資料格式

```
%REPORT{
```

```
<P>這個報告於 @DTW_rTIME()，@DTW_rDATE() 建立的。
```

```
%}
```

- 傳回：這個報告於 15:04:39，01 May 1997 建立的。

## DTW\_URLESCSEQ

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

URL 中不容許的已選取字元以其跳出值來置換，即所謂的 URL 編碼。

### 語法

@DTW\_URLESCSEQ(stringIn, stringOut)

@DTW\_rURLESCSEQ(stringIn)

### 參數

表 54. DTW\_URLESCSEQ 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含輸入字串的變數，該輸入字串具有 URL 中不允許的字元，會由它們的十六進位跳出值來置換。

### 回覆碼

表 55. DTW\_URLESCSEQ 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

您可以使用這個函數，將表56中列示的任何字元傳送到另一個巨集或 HTML 區塊。

表 56. URL 的字元跳出值

字元	名稱	程式碼
SPACE	空格	%20
"	雙引號	%22
#	號碼記號	%23
%	百分比	%25
&	& 記號	%26
+	加	%2B
\	反斜線	%2F
:	冒號	%3A
;	分號	%3B
<	小於	%3C

表 56. URL 的字元跳出值 (繼續)

=	等於	%3D
>	大於	%3E
?	問號	%3F
@	At 記號	%40
[	左方括弧	%5B
/	斜線	%5C
]	右方括弧	%5D
^	Carat	%5E
{	左大括弧	%7B
	直線	%7C
}	右大括弧	%7D
~	波型符號	%7E

## 範例

**範例 1：**以跳出值置換 *string1* 中的 & 符號與空格，並將結果指定到 *string2*

```
@DTW_URLESCSEQ(string1,string2)
```

- 輸入：string1 = "Guys & Dolls"
- 傳回：string2 = "Guys%20%26%20Dolls"

**範例 2：**以跳出碼置換空格與 & 符號。

```
@DTW_rURLESCSEQ("Guys & Dolls")
```

- 傳回："Guys%20%26%20Dolls"

**範例 3：**在 ROW 區塊中使用 DTW\_rURLESCSEQ，並以跳出碼置換空格和 'at' 字元。

```
%ROW{
<P><a href="fullrpt.mac/input?name=@DTW_rURLESCSEQ(V1)&email=@DTW_rURLESCSEQ(V2)">
$(V1)</a>
%}
```

- 輸入：V1="Patrick O'Brien", V2="obrien@ibm.com"
- 傳回：

```
<P><a href="fullrpt.mac/input?name=Patrick%20O'Brien&email="obrien%40ibm.com">
Patrick O'Brien</a>
```

當應用程式使用者按一下名稱 "Patrick O'Brien" 時，指定給名稱和電子郵件位址的值會在 URL 查詢字串中矩陣式排列，這會導致 Net.Data 執行 fullrpt.mac 巨集的輸入區段。

---

## 算術函數

這些函數可讓您作算術的計算。

**算術函數的 NLS 考量：** Net.Data 會根據執行 Net.Data 的 Web 伺服器上所指定的區域設定值，來顯示數值中的小數點。例如，Web 伺服器上小數點若是指定為逗點 (.)，Net.Data 便使用逗點，格式化含小數的資料。Net.Data 使用下列設定值，以決定哪個字元被指定為小數點。

**針對 OS/390、Windows NT、OS/2 及 UNIX 作業系統：**

Web 伺服器執行時的語言環境 (LOCALE)

**針對 OS/400 作業系統：**

- V4R2 或後續版次：由執行此處理的使用者設定檔所指定。
- V4R1 或先前版次：從 QDECFMT 系統值取回。

下列函數可讓您作算術的計算：

- 第145頁的『DTW\_ADD』
- 第147頁的『DTW\_DIVIDE』
- 第149頁的『DTW\_DIVREM』
- 第151頁的『DTW\_FORMAT』
- 第154頁的『DTW\_INTDIV』
- 第156頁的『DTW\_MULTIPLY』
- 第158頁的『DTW\_POWER』
- 第160頁的『DTW\_SUBTRACT』

## DTW\_ADD

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

兩數相加。

### 語法

@DTW\_ADD(number1, number2, precision, result)

@DTW\_ADD(number1, number2, result)

@DTW\_rADD(number1, number2, precision)

@DTW\_rADD(number1, number2)

### 參數

表 57. DTW\_ADD 參數

資料類型	參數	使用	說明
浮點數	<i>number1</i>	IN	一個代表某個數字的變數或文字字串。
浮點數	<i>number2</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	包含 <i>number1</i> 與 <i>number2</i> 總和的變數。

### 回覆碼

表 58. DTW\_ADD 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。
4002	運算作業產生的指數超過支援範圍 -999,999,999 至 +999,999,999。

### 範例

#### 範例 1：

```
@DTW_ADD(NUM1, NUM2, "2", result)
```

- 輸入：NUM1 = "105", NUM2 = "3"
- 傳回：result = "1.1E+2"

#### 範例 2：

```
@DTW_rADD("12", NUM2, "5")
```

- 輸入：NUM2 = "7.00"
- 傳回："19.00"



## DTW\_DIVIDE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

一個數字除以另一個數字。

### 語法

@DTW\_DIVIDE(number1, number2, precision, result)

@DTW\_DIVIDE(number1, number2, result)

@DTW\_rDIVIDE(number1, number2, precision)

@DTW\_rDIVIDE(number1, number2)

### 參數

表 59. DTW\_DIVIDE 參數

資料類型	參數	使用	說明
浮點數	<i>number1</i>	IN	一個代表被除數的變數或文字字串。
浮點數	<i>number2</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	包含將 <i>number1</i> 除以 <i>number2</i> 之結果的變數。

### 回覆碼

表 60. DTW\_DIVIDE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。
4002	運算作業產生的指數超過支援範圍 -999,999,999 至 +999,999,999。

### 範例

#### 範例 1：

```
@DTW_DIVIDE("8.0", NUM2, result)
```

- 輸入：NUM2 = "2"
- 傳回：result = "4"

#### 範例 2：

@DTW\_rDIVIDE("1", NUM2, "5")

- 輸入："1", NUM2 = "3"
- 傳回："0.33333"

**範例 3：**

@DTW\_rDIVIDE(NUM1, "2", "5")

- 輸入：NUM1 = "5"
- 傳回："2.5"

## DTW\_DIVREM

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將一個數字除以另一個數字，並傳回餘數。

### 語法

@DTW\_DIVREM(number1, number2, precision, result)

@DTW\_DIVREM(number1, number2, result)

@DTW\_rDIVREM(number1, number2, precision)

@DTW\_rDIVREM(number1, number2)

### 參數

表 61. DTW\_DIVREM 參數

資料類型	參數	使用	說明
浮點數	<i>number1</i>	IN	一個代表被除數的變數或文字字串。
浮點數	<i>number2</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	包含將 <i>number1</i> 除以 <i>number2</i> 之餘數的變數。

### 回覆碼

表 62. DTW\_DIVIDEREM 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。
4002	運算作業產生的指數超過支援範圍 -999,999,999 至 +999,999,999。

### 用法注意事項

如果餘數非零，則其正負號與第一個參數的正負號是相同的。

### 範例

範例 1：

```
@DTW_DIVREM(NUM1, NUM2, result)
```

- 輸入：NUM1 = "2.1", NUM2 = "3"
- 傳回：result = "2.1"

**範例 2：**

@DTW\_rDIVREM("10", NUM2)

- 輸入：NUM2 = "0.3"
- 傳回："0.1"

**範例 3：**

@DTW\_rDIVREM("3.6", "1.3")

- 傳回："1.0"

**範例 4：**

@DTW\_rDIVREM("-10", "3")

- 傳回："-1"

## DTW\_FORMAT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

自行設定數字的格式。

### 語法

@DTW\_FORMAT(number, before, after, exp, expt, precision, result)

@DTW\_FORMAT(number, before, after, exp, expt, result)

@DTW\_FORMAT(number, before, after, exp, result)

@DTW\_FORMAT(number, before, after, result)

@DTW\_FORMAT(number, before, result)

@DTW\_FORMAT(number, result)

@DTW\_rFORMAT(number, before, after, exp, expt, precision)

@DTW\_rFORMAT(number, before, after, exp, expt)

@DTW\_rFORMAT(number, before, after, exp)

@DTW\_rFORMAT(number, before, after)

@DTW\_rFORMAT(number, before)

@DTW\_rFORMAT(number)

### 參數

表 63. DTW\_FORMAT 參數

資料類型	參數	使用	說明
浮點數	<i>number</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>before</i>	IN	一個代表正完整數字的變數或文字字串。這是一個可選用的參數。您必須輸入一個空字串（""）以擁有額外的參數。
整數	<i>after</i>	IN	一個代表正完整數字的變數或文字字串。這是一個可選用的參數。您必須輸入一個空字串（""）以設定額外的參數。
整數	<i>exp</i>	IN	一個代表正完整數字的變數或文字字串。您必須設定一個空字串（""）以設定額外的參數。
整數	<i>expt</i>	IN	一個代表正完整數字的變數或文字字串。您必須輸入一個空字串（""）以設定額外的參數。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	一個含所指定之整數化和格式化之數字的值。

## 回覆碼

表 64. DTW\_FORMAT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。

## 用法注意事項

1. 如果只有指定 *number* 參數，則結果將會格式化成為 @DTW\_rADD(number, 『0』) 執行後的結果。
2. *before* 和 *after* 參數將說明，有多少字元會分別被用來作 *result* 參數中整數與小數部份。如果您省略了這些參數中的一項或兩項，則作為那部份的字元數與所需的一樣多。
3. 如果 *before* 參數不足以包含數字的整數部份（加上負數之前的負號），則會產生錯誤的結果。如果 *before* 參數大於該部分之所需，則 *number* 參數值將會以空白填補在左邊。如果 *after* 參數的大小與 *number* 參數的小數部份不同時，則數字將被四捨五入（或以零填補）以適合其大小。指定 0 會導致數字被四捨五入為整數。
4. *expp* 和 *expt* 參數可控制結果的指數部份。*expp* 參數設定指數部份位置的數量；預設值是使用與所需的一樣多（可能是零）。*expt* 參數設定使用指數表示法的觸發點。預設值是 *precision* 參數的預設值。
5. 如果 *expp* 是 0，則不提供指數，且數字將以簡單格式表示，如有必要的話會再加上零。如果 *expp* 沒有大到足夠包含指數，則會產生錯誤。
6. 如果整數或小數部份所需位置的數量分別超出 *expt* 或二倍的 *expt*，請使用指數表示法。如果 *expt* 是 0，則一定會使用指數表示法，除非指數是 0。（如果 *expp* 是 0，則會置換 *expt* 的一個 0 值。）如果當 *expp* 被指定一個非零的值而指數是 0，則 *expp+2* 的空白提供作為結果的指數部份。如果指數是 0 且未指定 *expp*，則將使用簡單格式。

## 範例

### 範例 1：

```
@DTW_FORMAT(NUM, BEFORE, result)
• 輸入：NUM = "3", BEFORE = "4"
• 傳回：result= " 3"
```

### 範例 2：

```
@DTW_FORMAT("1.73", "4", "0", result)
• 傳回：result = " 2"
```

### 範例 3：

@DTW\_FORMAT("1.73", "4", "3", result)

- 傳回：result = " 1.730"

**範例 4：**

@DTW\_FORMAT(" - 12.73", "", "4", result)

- 傳回：result = "-12.7300"

**範例 5：**

@DTW\_FORMAT("12345.73", "", "", "2", "2", result)

- 傳回：result = "1.234573E+04"

**範例 6：**

@DTW\_FORMAT("1.234573", "", "3", "", "0", result)

- 傳回：result = "1.235"

**範例 7：**

@DTW\_rFORMAT(" - 12.73")

- 傳回：" - 12.73"

**範例 8：**

@DTW\_rFORMAT("0.000")

- 傳回："0"

**範例 9：**

@DTW\_rFORMAT("12345.73", "", "", "3", "6")

- 傳回："12345.73"

**範例 10：**

@DTW\_rFORMAT("1234567e5", "", "3", "0")

- 傳回："123456700000.000"

**範例 11：**

@DTW\_rFORMAT("12345.73", "", "3", "", "0")

- 傳回："1.235E+4"

## DTW\_INTDIV

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將一個數字除以另一個數字，並傳回結果的整數部份。

### 語法

@DTW\_INTDIV(number1, number2, precision, result)

@DTW\_INTDIV(number1, number2, result)

@DTW\_rINTDIV(number1, number2, precision)

@DTW\_rINTDIV(number1, number2)

### 參數

表 65. DTW\_INTDIV 參數

資料類型	參數	使用	說明
浮點數	<i>number1</i>	IN	一個代表被除數的變數或文字字串。
浮點數	<i>number2</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	包含 <i>number1</i> 除以 <i>number2</i> 所得商的整數部份的變數。

### 回覆碼

表 66. DTW\_INTDIV 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。
4002	運算作業產生的指數超過支援範圍 -999,999,999 至 +999,999,999。

### 範例

#### 範例 1：

@DTW\_INTDIV(NUM1, NUM2, result)

- 輸入：NUM1 = "10", NUM2 = "3"
- 傳回：result = "3"

#### 範例 2：

@DTW\_rINTDIV("2", NUM2)



- 輸入：NUM2 = "3"
- 傳回："0"

## DTW\_MULTIPLY

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

兩個數字相乘。

### 語法

@DTW\_MULTIPLY(number1, number2, precision, result)

@DTW\_MULTIPLY(number1, number2, result)

@DTW\_rMULTIPLY(number1, number2, precision)

@DTW\_rMULTIPLY(number1, number2)

### 參數

表 67. DTW\_MULTIPLY 參數

資料類型	參數	使用	說明
浮點數	<i>number1</i>	IN	一個代表某個數字的變數或文字字串。
浮點數	<i>number2</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	包含 <i>number1</i> 與 <i>number2</i> 乘積的變數。

### 回覆碼

表 68. DTW\_MULTIPLY 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。
4002	運算作業產生的指數超過支援範圍 -999,999,999 至 +999,999,999。

### 範例

#### 範例 1：

@DTW\_MULTIPLY(NUM1, NUM2, result)

- 輸入：NUM1 = "4", NUM2 = "5"
- 傳回：result = "20"

#### 範例 2：

@DTW\_rMULTIPLY("0.9", NUM2)

- 輸入：NUM2 = "0.8"
- 傳回："0.72"

## DTW\_POWER

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將整數升幂成整數次方。

### 語法

@DTW\_POWER(number1, number2, precision, result)

@DTW\_POWER(number1, number2, result)

@DTW\_rPOWER(number1, number2, precision)

@DTW\_rPOWER(number1, number2)

### 參數

表 69. DTW\_POWER 參數

資料類型	參數	使用	說明
浮點數	<i>number1</i>	IN	一個代表要升幂成某次方的數字之變數或文字字串。
浮點數	<i>number2</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	包含 <i>number1</i> 的 <i>number2</i> 次方結果的變數。

### 回覆碼

表 70. DTW\_POWER 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。
4002	運算作業產生的指數超過支援範圍 -999,999,999 至 +999,999,999。

### 範例

#### 範例 1：

@DTW\_POWER(NUM1, NUM2, result)

- 輸入：NUM1 = "2", NUM2 = "-3"
- 傳回：result = "0.125"

#### 範例 2：

```
@DTW_rPOWER("1.7", NUM2, precision)
```

- 輸入：NUM2 = "8", precision = "5"
- 傳回："69.758"

## DTW\_SUBTRACT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

一個數字減另一個數字。

### 語法

@DTW\_SUBTRACT(number1, number2, precision, result)

@DTW\_SUBTRACT(number1, number2, result)

@DTW\_rSUBTRACT(number1, number2, precision)

@DTW\_rSUBTRACT(number1, number2)

### 參數

表 71. DTW\_SUBTRACT 參數

資料類型	參數	使用	說明
浮點數	<i>number1</i>	IN	一個代表要減去另一個數字的數字之變數或文字字串。
浮點數	<i>number2</i>	IN	一個代表某個數字的變數或文字字串。
整數	<i>precision</i>	IN	一個代表指定結果之精確度之正整數的變數或文字字串。預設值是 9。
浮點數	<i>result</i>	OUT	包含 <i>number1</i> 與 <i>number2</i> 差的變數。

### 回覆碼

表 72. DTW\_SUBTRACT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
4000	參數包含無效的整數值。
4001	參數包含無效的數值。
4002	運算作業產生的指數超過支援範圍 -999,999,999 至 +999,999,999。

### 範例

#### 範例 1：

```
@DTW_SUBTRACT(NUM1, NUM2, comp)
%IF(comp > "0")
<P>$(NUM1) 大於 $(NUM2)。
%ENDIF
```

- 輸入：NUM2 = "2.07"

- 傳回："-0.77"

這個範例顯現一個比較數值的方式，其為 `Net.Data` 中的字串。

**範例 2：**

```
@DTW_SUBTRACT(NUM1, NUM2, result)
```

- 輸入：NUM1 = "1.3"，NUM2 = "1.07"
- 傳回：result = "0.23"

**範例 3：**

```
@DTW_rSUBTRACT("1.3", NUM2)
```

- 輸入：NUM2 = "2.07"
- 傳回："-0.77"

---

## 字串函數

下列的函數是由 `Net.Data` 所支援的標準字串函數組：

- 第163頁的『DTW\_ASSIGN』
- 第164頁的『DTW\_CHARTOHEX』
- 第165頁的『DTW\_CONCAT』
- 第166頁的『DTW\_DELSTR』
- 第167頁的『DTW\_HEXTOCHAR』
- 第168頁的『DTW\_INSERT』
- 第170頁的『DTW\_LASTPOS』
- 第172頁的『DTW\_LENGTH』
- 第173頁的『DTW\_LOWERCASE』
- 第175頁的『DTW\_POS』
- 第177頁的『DTW\_REPLACE』
- 第179頁的『DTW\_REVERSE』
- 第180頁的『DTW\_STRIP』
- 第182頁的『DTW\_SUBSTR』
- 第184頁的『DTW\_TRANSLATE』
- 第186頁的『DTW\_UPPERCASE』

**OS/390、OS/2、Windows NT 與 UNIX 的 MBCS 支援：** 您可以使用 `DTW_MBMODE` 架構值來指定字組與字串函數的多位元組字集（MBCS）支援。請在 `Net.Data` 起始設定檔中設定這個值；預設值是不支援。您可以在 `Net.Data` 巨集中設定 `DTW_MBMODE` 變數，來置換起始設定檔中的值。請參閱 *Net.Data 管理及程式設計手冊* 中的架構變數部分，與第107頁的『DTW\_MBMODE』以取得有關詳細資訊。

**OS/400 的 MBCS 支援：** 自動提供 DBCS 支援，無須這個變數。



## DTW\_ASSIGN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定一個值給變數。

### 語法

```
@DTW_ASSIGN(stringOut, stringIn)
```

### 參數

表 73. DTW\_ASSIGN 參數

資料類型	參數	使用	說明
字串	<i>stringOut</i>	OUT	包含與 <i>stringIn</i> 相同文字字串的變數。
字串	<i>stringIn</i>	IN	一個變數或文字字串。

### 回覆碼

表 74. DTW\_ASSIGN 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_ASSIGN(RC, "0")
```

- 將 RC 設定為 "0"。

#### 範例 2：

```
@DTW_ASSIGN(string1, string2)
```

- 將 *string1* 設定為 *string2* 的值。

## DTW\_CHARTOHEX

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

將字串中每一個字元轉換成兩個十六進位字元。

### 語法

@DTW\_CHARTOHEX(stringIn, stringOut)

@DTW\_rCHARTOHEX(stringIn)

### 參數

表 75. DTW\_CHARTOHEX 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	要被轉換的變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含以十六進位格式表示的 <i>stringIn</i> 之變數。

### 回覆碼

表 76. DTW\_CHARTOHEX 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

每一個十六進位字元代表輸入字元的 4 位元 (一個字元以 8 位元代表)。

### 範例

範例 1：EBCDIC 作業系統

@DTW\_rCHARTOHEX("12345")

- 傳回："F1F2F3F4F5"

範例 2：ASCII 作業系統

@DTW\_rCHARTOHEX("12345")

- 傳回："C1C2C3C4C5"

## DTW\_CONCAT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

連接兩個字串。

### 語法

@DTW\_CONCAT(stringIn1, stringIn2, stringOut)

@DTW\_rCONCAT(stringIn1, stringIn2)

### 參數

表 77. DTW\_CONCAT 參數

資料類型	參數	使用	說明
字串	<i>stringIn1</i>	IN	一個變數或文字字串。
字串	<i>stringIn2</i>	IN	一個變數或文字字串。
字串	<i>stringOut</i>	OUT	包含字串 ' <i>stringIn1stringIn2</i> ' 的變數，其中 <i>string1</i> 與 <i>string2</i> 相連。

### 回覆碼

表 78. DTW\_CONCAT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_CONCAT("This", " is a test.", result)
```

- 傳回：result = "This is a test."

#### 範例 2：

```
@DTW_CONCAT(string1, "1-2-3", result)
```

- 輸入：string1 = "Testing "
- 傳回：result = "Testing 1-2-3"

#### 範例 3：

```
@DTW_rCONCAT("This", " is a test.")
```

- 傳回："This is a test."

## DTW\_DELSTR

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

刪除某字串的子字串，從第 *n* 個字元開始刪除 *length* 個字元。

### 語法

@DTW\_DELSTR(stringIn, n, length, stringOut)

@DTW\_DELSTR(stringIn, n, stringOut)

@DTW\_rDELSTR(stringIn, n, length)

@DTW\_rDELSTR(stringIn, n)

### 參數

表 79. DTW\_DELSTR 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
整數	<i>n</i>	IN	要開始刪除子字串的字元位置。如果 <i>n</i> 大於 <i>stringIn</i> 的長度，則 <i>stringOut</i> 會被設定為 <i>stringIn</i> 的值。
整數	<i>length</i>	IN	所要刪除之子字串的長度。預設值是刪除所有字元，直到 <i>stringIn</i> 的尾端。
字串	<i>stringOut</i>	OUT	包含 <i>stringIn</i> 已修改格式的變數。

### 回覆碼

表 80. DTW\_DELSTR 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_DELSTR("abcde", "3", "2", result)
```

- 傳回：result = "abe"

#### 範例 2：

```
@DTW_rDELSTR("abcde", "4", "1")
```

- 傳回："abce"

## DTW\_HEXTOCHAR

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

將字串中每一個十六進位字元轉換成一個字元值。

### 語法

@DTW\_HEXTOCHAR(stringIn, stringOut)

@DTW\_rHEXTOCHAR(stringIn)

### 參數

表 81. DTW\_HEXTOCHAR 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	被轉換的變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含以字元格式表示的 <i>stringIn</i> 之變數。

### 回覆碼

表 82. DTW\_HEXTOCHAR 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 用法注意事項

輸入字串中每一個十六進位字元代表結果字串中的 4 位元 (一個字元以 8 位元代表)。  
輸入字串必須包含偶數個十六進位字元，並可包含下列字元：0-9、A-F 及 a-f。

### 範例

範例 1：EBCDIC 作業系統

```
@DTW_rHEXTOCHAR("F0F1F2")
```

- 傳回："123"

範例 2：ASCII 作業系統

```
@DTW_rHEXTOCHAR("303132")
```

- 傳回："123"

## DTW\_INSERT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將一個字串插入至另一個字串中第  $n$  個字元之後開始的位置。

### 語法

@DTW\_INSERT(stringIn1, stringIn2, n, length, pad, stringOut)

@DTW\_INSERT(stringIn1, stringIn2, n, length, stringOut)

@DTW\_INSERT(stringIn1, stringIn2, n, stringOut)

@DTW\_INSERT(stringIn1, stringIn2, stringOut)

@DTW\_rINSERT(stringIn1, stringIn2, n, length, pad)

@DTW\_rINSERT(stringIn1, stringIn2, n, length)

@DTW\_rINSERT(stringIn1, stringIn2, n)

@DTW\_rINSERT(stringIn1, stringIn2)

### 參數

表 83. DTW\_INSERT 參數

資料類型	參數	使用	說明
字串	<i>stringIn1</i>	IN	被插入 <i>stringIn2</i> 的變數或文字字串。
字串	<i>stringIn2</i>	IN	一個變數或文字字串。
整數	<i>n</i>	IN	字元位置是在 <i>stringIn2</i> 中，其後插入 <i>stringIn1</i> 。如果 $n$ 大於 <i>stringIn2</i> 的長度，則將會使用填補字元， <i>pad</i> 來填補不足的字元。預設值是在 <i>stringIn2</i> 的開頭插入。
整數	<i>length</i>	IN	所要插入之 <i>stringIn1</i> 字元數目。如果這個參數大於 <i>stringIn1</i> 的長度，則將會以填補字元， <i>pad</i> 來填滿字串。預設值是 <i>stringIn1</i> 的長度。
整數	<i>pad</i>	IN	如同 $n$ 與 <i>length</i> 的說明，填補字元。預設 <i>pad</i> 字元是空白。
字串	<i>stringOut</i>	OUT	一個包含有插入部分或所有 <i>stringIn1</i> 的修改後之 <i>stringIn2</i> 。

### 回覆碼

表 84. DTW\_INSERT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

表 84. DTW\_INSERT 回覆碼 (繼續)

回覆碼	說明
1007	參數包含的值無效。

## 範例

### 範例 1：

```
@DTW_INSERT("123", "abc", result)
```

- 傳回：result = "123abc"

### 範例 2：

```
@DTW_INSERT("123", "abc", "5", result)
```

- 傳回：result = "abc 123"

### 範例 3：

```
@DTW_INSERT("123", "abc", "5", "6", result)
```

- 傳回：result = "abc 123"

### 範例 4：

```
@DTW_INSERT("123", "abc", "5", "6", "/", result)
```

- 傳回：result = "abc//123///"

### 範例 5：

```
@DTW_rINSERT("123", "abc", "5", "6", "+")
```

- 傳回："abc++123+++"

## DTW\_LASTPOS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回在另一個字串中，某一個字串最後一次出現的位置，從第 *n* 個字元開始往回算（從右到左）

### 語法

@DTW\_LASTPOS(stringIn1, stringIn2, n, position)

@DTW\_LASTPOS(stringIn1, stringIn2, position)

@DTW\_rLASTPOS(stringIn1, stringIn2, n)

@DTW\_rLASTPOS(stringIn1, stringIn2)

### 參數

表 85. DTW\_LASTPOS 參數

資料類型	參數	使用	說明
字串	<i>stringIn1</i>	IN	一個在 <i>stringIn2</i> 中被搜尋的變數或文字字串。
字串	<i>stringIn2</i>	IN	一個變數或文字字串。
整數	<i>n</i>	IN	位於 <i>stringIn2</i> 中開始搜尋 <i>stringIn1</i> 的字元位置。預設值是從最後一個字元開始搜尋，並往前掃描（從右至左）。
整數	<i>position</i>	OUT	在 <i>stringIn2</i> 中出現最後一個 <i>stringIn1</i> 的位置。如果找不到的話，將會傳回 0。

### 回覆碼

表 86. DTW\_LASTPOS 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_LASTPOS(" ", "abc def ghi", result)
```

- 傳回：result = "8"

#### 範例 2：

```
@DTW_LASTPOS(" ", "abc def ghi", "10", result)
```



- 傳回：result = "8"

**範例 3：**

```
@DTW_rLASTPOS(" ", "abc def ghi", "7")
```

- 傳回："4"

## DTW\_LENGTH

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回字串的長度。

### 語法

@DTW\_LENGTH(stringIn, length)

@DTW\_rLENGTH(stringIn)

### 參數

表 87. DTW\_LENGTH 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
整數	<i>length</i>	OUT	包含 <i>stringIn</i> 中字元數目的符號。

### 回覆碼

表 88. DTW\_LENGTH 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_LENGTH("abcdefgh", result)
```

- 傳回：result = "8"

#### 範例 2：

```
@DTW_rLENGTH("")
```

- 傳回："0"

## DTW\_LOWERCASE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回一個全部都是小寫的字串。

### 語法

@DTW\_LOWERCASE(stringIn, stringOut)

@DTW\_rLOWERCASE(stringIn)

@DTW\_mLOWERCASE(stringMult1, stringMult2, ..., stringMultn)

### 參數

表 89. DTW\_LOWERCASE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個有大小寫字元之變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含所有都是小寫字元之 <i>stringIn</i> 的變數。
字串	<i>stringMult</i>	INOUT	<ul style="list-style-type: none"><li>在輸入上：包含字串的變數。</li><li>在輸出上：一個包含被轉換為小寫之輸入字串的變數。</li></ul>

### 回覆碼

表 90. DTW\_LOWERCASE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_LOWERCASE("This", stringOut)
```

- 傳回：stringOut = "this"

#### 範例 2：

```
@DTW_rLOWERCASE(string1)
```

- 輸入：string1 = "Hello"
- 傳回："hello"

#### 範例 3：

```
@DTW_mLOWERCASE(string1, string2, string3)
```

- 輸入：string1 = "THIS", string2 = "IS", string3 = "LOWERCASE"

- 傳回: `string1 = "this", string2 = "is", string3 = "lowercase"`

## DTW\_POS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回在另一個字串中，某一字串第一次出現的位置，使用向後搜尋形式。

### 語法

@DTW\_POS(stringIn1, stringIn2, n, nOut)

@DTW\_POS(stringIn1, stringIn2, nOut)

@DTW\_rPOS(stringIn1, stringIn2, n)

@DTW\_rPOS(stringIn1, stringIn2)

### 參數

表 91. DTW\_POS 參數

資料類型	參數	使用	說明
字串	<i>stringIn1</i>	IN	一個所要搜尋的變數或文字字串。
字串	<i>stringIn2</i>	IN	一個所要搜尋的變數或文字字串。
整數	<i>n</i>	IN	在 <i>stringIn2</i> 中開始進行搜尋的字元位置。 預設值是開始在 <i>stringIn2</i> 的第一個字元進行搜尋。
整數	<i>nOut</i>	OUT	一個包含在 <i>stringIn2</i> 中第一個出現 <i>stringIn1</i> 之位置的變數。如果找不到的話， 將會傳回 0。

### 回覆碼

表 92. DTW\_POS 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_POS("day", "Saturday", result)
```

- 傳回：result = "6"

#### 範例 2：

```
@DTW_POS("a", "Saturday", "3", result)
```

- 傳回：result = "7"

**範例 3：**

```
@DTW_rPOS(" ", "abc def ghi", "5")
```

- 傳回："8"

## DTW\_REPLACE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

置換字串中的字元。

### 語法

@DTW\_REPLACE(stringIn, stringFrom, stringTo, n, option, stringOut)

@DTW\_REPLACE(stringIn, stringFrom, stringTo, n, stringOut)

@DTW\_REPLACE(stringIn, stringFrom, stringTo, stringOut)

@DTW\_rREPLACE(stringIn, stringFrom, stringTo, n, option)

@DTW\_rREPLACE(stringIn, stringFrom, stringTo, n)

@DTW\_rREPLACE(stringIn, stringFrom, stringTo)

### 參數

表 93. DTW\_REPLACE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	被搜尋的變數或文字字串。
字串	<i>stringFrom</i>	IN	被置換的變數或文字字串。
字串	<i>stringTo</i>	IN	一個置換 <i>stringFrom</i> 出現項目的變數或文字字串。
整數	<i>n</i>	IN	要開始搜尋的字元位置。
字串	<i>option</i>	IN	指定要置換所有出現項目或僅置換第一個出現項目，可以有下列其中一值：  <b>A 或 a</b> 置換所有出現項目。預設值是 A。 <b>F 或 f</b> 僅置換第一個出現項目。
字串	<i>stringOut</i>	OUT	包含 <i>stringIn</i> 的變數，其中 <i>stringFrom</i> 已被 <i>stringTo</i> 置換。

### 回覆碼

表 94. DTW\_REPLACE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_rREPLACE("ABCABCABC", "AB", "1234")
```

- 傳回："1234C1234C1234C"



## DTW\_REVERSE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將字串反轉，讓最後一個字元變成第一個字元，第二個字元變倒數第二個字元，直到整個字串反轉過來為止。

### 語法

@DTW\_REVERSE(stringIn, stringOut)

@DTW\_rREVERSE(stringIn)

### 參數

表 95. DTW\_REVERSE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個要反轉的變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含 <i>stringIn</i> 之反轉格式的變數。

### 回覆碼

表 96. DTW\_REVERSE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_REVERSE("This is it.", result)
```

- 傳回：result = ".ti si sihT"

#### 範例 2：

```
@DTW_rREVERSE(string1)
```

- 輸入：string1 = "reversed"
- 傳回："desrever"

## DTW\_STRIP

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

除去字串中的前導空白、尾端空白，或兩者。

### 語法

@DTW\_STRIP(stringIn, option, stringOut)

@DTW\_STRIP(stringIn, stringOut)

@DTW\_rSTRIP(stringIn, option)

@DTW\_rSTRIP(stringIn)

### 參數

表 97. DTW\_STRIP 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
字串	<i>option</i>	IN	指定要從 <i>stringIn</i> 中所除去的空白。預設值是 B。 B 或 b - 將前導與尾隨空白均除去 L 或 l - 只除去前導空白 T 或 t - 只除去尾隨空白
字串	<i>stringOut</i>	OUT	一個已依選項所指定除去空白之 <i>stringIn</i> 的變數。

### 回覆碼

表 98. DTW\_STRIP 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_STRIP(" day ", result)
```

- 傳回：result = "day"

#### 範例 2：

```
@DTW_STRIP(" day ", "T", result)
```

- 傳回：result = "day"

**範例 3：**

```
@DTW_rSTRIP(" a day ", "L")
```

- 傳回："a day "

## DTW\_SUBSTR

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將字串的子字串與可選用的填補字元一起傳回。

### 語法

@DTW\_SUBSTR(stringIn, n, length, pad, stringOut)

@DTW\_SUBSTR(stringIn, n, length, stringOut)

@DTW\_SUBSTR(stringIn, n, stringOut)

@DTW\_rSUBSTR(stringIn, n, length, pad)

@DTW\_rSUBSTR(stringIn, n, length)

@DTW\_rSUBSTR(stringIn, n)

### 參數

表 99. DTW\_SUBSTR 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個所要搜尋的變數或文字字串。
整數	<i>n</i>	IN	子字串的第一個字元位置。預設值是從 <i>stringIn</i> 的開頭開始
整數	<i>length</i>	IN	子字串的字元數目。預設值是其餘的字串。
字串	<i>pad</i>	IN	如果 <i>n</i> 大於 <i>stringIn</i> 的長度，或如果 <i>length</i> 長於 <i>stringIn</i> 時，將會使用的填補字元。預設值是空白。
字串	<i>stringOut</i>	OUT	一個包含 <i>stringIn</i> 之子字串的變數。

### 回覆碼

表 100. DTW\_SUBSTR 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_SUBSTR("abc", "2", result)
```

- 傳回：result = "bc"

#### 範例 2：

@DTW\_SUBSTR("abc", "2", "4", result)

- 傳回：result = "bc"

**範例 3：**

@DTW\_SUBSTR("abc", "2", "4", ".", result )

- 傳回：result = "bc.."

**範例 4：**

@DTW\_rSUBSTR("abc", "2", "6", ".")

- 傳回："bc...."

## DTW\_TRANSLATE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回字串，其中每一個字元都轉換成另一個字元或維持不變。

### 語法

@DTW\_TRANSLATE(stringIn, tableO, tableI, default, stringOut)

@DTW\_TRANSLATE(stringIn, tableO, tableI, stringOut)

@DTW\_TRANSLATE(stringIn, tableO, stringOut)

@DTW\_TRANSLATE(stringIn, stringOut)

@DTW\_rTRANSLATE(stringIn, tableO, tableI, default)

@DTW\_rTRANSLATE(stringIn, tableO, tableI)

@DTW\_rTRANSLATE(stringIn, tableO)

@DTW\_rTRANSLATE(stringIn)

### 參數

表 101. DTW\_TRANSLATE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
字串	<i>tableO</i>	IN	使用作為轉換表格的變數或文字字串。請使用空值 ("") 來指定 <i>tableI</i> 或 <i>default</i> ；否則這個參數將是可選用的。
字串	<i>tableI</i>	IN	一個將在 <i>stringIn</i> 中被搜尋的變數或文字字串。請使用空值 ("") 來指定 <i>default</i> ；否則這個參數將是可選用的。
字串	<i>default</i>	IN	所使用的預設字元。預設值是空白。
字串	<i>stringOut</i>	OUT	一個包含 <i>stringIn</i> 轉換結果的變數。

### 回覆碼

表 102. DTW\_TRANSLATE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 用法注意事項

1. 如果 *tableI*、*tableO* 和 *default* 字元不在參數列示中，則將會把 *stringIn* 轉換為大寫。

2. 如果 *tableI* 和 *tableO* 都在列示中，則將會搜尋 *tableI* 中輸入字串的每一個字元，並將其轉換為在 *tableO* 中的相符字元。如果有一個 *tableI* 中的字元，在 *tableO* 沒有對應的字元，則會使用 *default* 字元來代替。

## 範例

### 範例 1：

```
@DTW_TRANSLATE("abbc", result)
```

- 傳回：result = "ABBC"

### 範例 2：

```
@DTW_TRANSLATE("abbc", "R", "bc", result)
```

- 傳回：result = "aRR "

### 範例 3：

```
@DTW_rTRANSLATE("abcdef", "12", "abcd", ".")
```

- 傳回： "12..ef"

### 範例 4：

```
@DTW_rTRANSLATE("abbc", "", "", "")
```

- 傳回："abbc"

## DTW\_UPPERCASE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回大寫字體的字串。

### 語法

@DTW\_UPPERCASE(stringIn, stringOut)

@DTW\_rUPPERCASE(stringIn)

@DTW\_mUPPERCASE(stringMult1, stringMult2, ..., stringMultn)

### 參數

表 103. DTW\_UPPERCASE 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個有大小寫字元之變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含所有都是大寫字元之 <i>stringIn</i> 的變數。
字串	<i>stringMult</i>	INOUT	<ul style="list-style-type: none"><li>在輸入上：包含字串的變數。</li><li>在輸出上：一個包含被轉換為大寫之輸入字串的變數。</li></ul>

### 回覆碼

表 104. DTW\_UPPERCASE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_UPPERCASE("Test", result)
```

- 傳回：result = "TEST"

#### 範例 2：

```
@DTW_rUPPERCASE(string1)
```

- 輸入：string1 = "Web pages"
- 傳回："WEB PAGES"

#### 範例 3：

```
@DTW_mUPPERCASE(string1, string2, string3)
```

- 輸入：string1 = "This", string2 = "is", string3 = "uppercase"



- 傳回：string1 = "THIS", string2 = "IS", string3 = "UPPERCASE"

---

## 字組函數

這些函數藉由修改字組或字組集來支援字串函數。Net.Data 將字組解譯為一個以空白做分隔的字串，或者一個在兩端均具有空白的字串。這裡是幾個範例：

字串值	字組數
one two three	3
one , two , three	5
Part 2: Internet Sales Grow	5

**OS/390、OS/2、Windows NT 與 UNIX 的 MBCS 支援：** 您可以使用 DTW\_MBMODE 架構值來指定字組與字串函數的多位元組字集（MBCS）支援。請在 Net.Data 起始設定檔中設定這個值；預設值是不支援。您可以在 Net.Data 巨集中設定 DTW\_MBMODE 變數，來置換起始設定案中的值。請參閱 *Net.Data 管理及程式設計手冊* 中的架構變數部分，與第107頁的『DTW\_MBMODE』以取得有關詳細資訊。

**OS/400 的 MBCS 支援：** 自動提供 DBCS 支援，無須這個變數。

下列函數是 Net.Data 支援的字組函數：

- 第189頁的『DTW\_DELWORD』
- 第191頁的『DTW\_SUBWORD』
- 第193頁的『DTW\_WORD』
- 第194頁的『DTW\_WORDINDEX』
- 第195頁的『DTW\_WORDLENGTH』
- 第196頁的『DTW\_WORDPOS』
- 第198頁的『DTW\_WORDS』

## DTW\_DELWORD

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

刪除字串中的字組，從第  $n$  個字組開始刪除  $length$  所指定的字組數。

### 語法

@DTW\_DELWORD(stringIn, n, length, stringOut)

@DTW\_DELWORD(stringIn, n, stringOut)

@DTW\_rDELWORD(stringIn, n, length)

@DTW\_rDELWORD(stringIn, n)

### 參數

表 105. DTW\_DELWORD 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
整數	$n$	IN	要被刪除的第一個字組的位置。
整數	<i>length</i>	IN	要刪除的字組數目。預設值是刪除從 $n$ 到 <i>stringIn</i> 尾端的所有字組。選用性的參數。
字串	<i>stringOut</i>	OUT	包含 <i>stringIn</i> 已修改格式的變數。

### 回覆碼

表 106. DTW\_DELWORD 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1

```
@DTW_DELWORD("Now is the time", "5", result)
```

• 傳回：result = "Now is the time"

#### 範例 2：

```
@DTW_DELWORD("Now is the time", "2", result)
```

• 傳回：result = "Now"

#### 範例 3：

```
@DTW_DELWORD("Now is the time", "2", "2", result)
```

- 傳回：result = "Now time"

**範例 4：**

```
@DTW_rDELWORD("Now is the time.", "3")
```

- 傳回："Now is"

## DTW\_SUBWORD

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回字串的子字串，從第  $n$  個字組開始傳回  $length$  所指定的字組數。

### 語法

@DTW\_SUBWORD(stringIn, n, length, stringOut)

@DTW\_SUBWORD(stringIn, n, stringOut)

@DTW\_rSUBWORD(stringIn, n, length)

@DTW\_rSUBWORD(stringIn, n)

### 參數

表 107. DTW\_SUBWORD 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
整數	$n$	IN	子字串之第一個字組的字組位置。如果這個值大於在 <i>stringIn</i> 中的字組數目，則將會傳回一個空值。
整數	<i>length</i>	IN	在子字串中的字組數目。如果這個值大於從 $n$ 到 <i>stringIn</i> 尾端的字組數目，則將會傳回一直到 <i>stringIn</i> 尾端的所有字組。預設值是傳回所有從 $n$ 到 <i>stringIn</i> 尾端的字組。
字串	<i>stringOut</i>	OUT	一個包含由 $n$ 和 <i>length</i> 所指定之 <i>stringIn</i> 子字串的變數。

### 回覆碼

表 108. DTW\_SUBWORD 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_SUBWORD("Now is the time", "5", result)
```

• 傳回：result = ""

#### 範例 2：

```
@DTW_SUBWORD("Now is the time", "2", result)
```

- 傳回：result = "is the time"

**範例 3：**

```
@DTW_SUBWORD(Now is the time", "2", "2", result)
```

- 傳回：result = "is the"

**範例 4：**

```
@DTW_rSUBWORD("Now is the time", "3")
```

- 傳回："the time"

## DTW\_WORD

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回字串中的第  $n$  個字組。

### 語法

@DTW\_WORD(stringIn, n, stringOut)

@DTW\_rWORD(stringIn, n)

### 參數

表 109. DTW\_WORD 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
整數	$n$	IN	所要傳回字組的位置。如果這個值大於在 <i>stringIn</i> 中的字組數目，則將會傳回一個空值。
字串	<i>stringOut</i>	OUT	包含位於字組位置 $n$ 之字組的變數。

### 回覆碼

表 110. DTW\_WORD 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_WORD("Now is the time", "3", result)
```

- 傳回：result = "the"

#### 範例 2：

```
@DTW_WORD("Now is the time", "5", result)
```

- 傳回：result = ""

#### 範例 3：

```
@DTW_rWORD("Now is the time", "4")
```

- 傳回："time"

## DTW\_WORDINDEX

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回字串第  $n$  個字組中第一個字元的字元位置。

### 語法

@DTW\_WORDINDEX(stringIn, n, stringOut)

@DTW\_rWORDINDEX(stringIn, n)

### 參數

表 111. DTW\_WORDINDEX 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
整數	$n$	IN	所要索引之字組的字組位置。如果這個值大於在輸入字串中的字組數目，則將會傳回 0。
字串	<i>stringOut</i>	OUT	一個包含 <i>stringIn</i> 之第 $n$ 個字組之字元位置的變數。

### 回覆碼

表 112. DTW\_WORDINDEX 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_WORDINDEX("Now is the time", "3", result)
```

- 傳回：result = "8"

#### 範例 2：

```
@DTW_WORDINDEX("Now is the time", "6", result)
```

- 傳回：result = "0"

#### 範例 3：

```
@DTW_rWORDINDEX("Now is the time", "2")
```

- 傳回："5"



## DTW\_WORDLENGTH

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回字串第  $n$  個字組的長度。

### 語法

@DTW\_WORDLENGTH(stringIn, n, stringOut)

@DTW\_rWORDLENGTH(stringIn, n)

### 參數

表 113. DTW\_WORDLENGTH 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
整數	<i>n</i>	IN	您想要知道的字組長度之字組位置。如果這個值大於在輸入字串中的字組數目，則將會傳回 0。
字串	<i>stringOut</i>	OUT	一個包含在 <i>stringIn</i> 中之第 $n$ 個字組長度的變數。

### 回覆碼

表 114. DTW\_WORDLENGTH 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 範例

#### 範例 1：

```
@DTW_WORDLENGTH("Now is the time", "1", result)
```

- 傳回：result = "3"

#### 範例 2：

```
@DTW_rWORDLENGTH("Now is the time", "6")
```

- 傳回："0"

## DTW\_WORDPOS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回在另一個字串中，某一個字串第一次出現的字組號碼。

### 語法

@DTW\_WORDPOS(stringIn1, stringIn2, n, stringOut)

@DTW\_WORDPOS(stringIn1, stringIn2, stringOut)

@DTW\_rWORDPOS(stringIn1, stringIn2, n)

@DTW\_rWORDPOS(stringIn1, stringIn2)

### 參數

表 115. DTW\_WORDPOS 參數

資料類型	參數	使用	說明
字串	<i>stringIn1</i>	IN	一個變數或文字字串。
字串	<i>stringIn2</i>	IN	一個所要搜尋的變數或文字字串。
整數	<i>n</i>	IN	在 <i>stringIn2</i> 中要開始進行搜尋的字組位置。如果這個值大於在 <i>stringIn2</i> 中的字組數目，則將會傳回 0。預設值是從 <i>stringIn2</i> 的開頭進行搜尋。
字串	<i>stringOut</i>	OUT	<i>stringIn1</i> 在 <i>stringIn2</i> 中之的字組位置。

### 回覆碼

表 116. DTW\_WORDPOS 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。

### 用法注意事項

多重空白將以一個單一空白來看待，以供比較。

### 範例

範例 1：

```
@DTW_WORDPOS("the", "Now is the time", result)
```

- 傳回：result = "3"

範例 2：

@DTW\_WORDPOS("The", "Now is the time", result)

- 傳回：result = "0"

**範例 3：**

@DTW\_WORDPOS("The", "Now is the time", "5", result)

- 傳回：result = "0"

**範例 4：**

@DTW\_WORDPOS("is the", "Now is the time", result)

- 傳回：result = " 2"

**範例 5：**

@DTW\_rWORDPOS("be", "To be or not to be", "3")

- 傳回："6"

## DTW\_WORDS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回在字串中的字組數目。

### 語法

@DTW\_WORDS(stringIn, stringOut)

@DTW\_rWORDS(stringIn)

### 參數

表 117. DTW\_WORDS 參數

資料類型	參數	使用	說明
字串	<i>stringIn</i>	IN	一個變數或文字字串。
字串	<i>stringOut</i>	OUT	一個包含在 <i>stringIn</i> 中之字組數目的變數。

### 回覆碼

表 118. DTW\_WORDS 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

#### 範例 1：

```
@DTW_WORDS("Now is the time", result)
```

- 傳回：

```
result = "4"
```

#### 範例 2：

```
@DTW_rWORDS(" ")
```

- 傳回："0"

這些函數可簡化 Net.Data 表格的使用化，且比使用 REXX、C 或 Perl 撰寫您自己的函數更有效率。

- 第200頁的『DTW\_TB\_APPENDROW』
- 第201頁的『DTW\_TB\_COLS』
- 第203頁的『DTW\_TB\_DELETEROW』
- 第202頁的『DTW\_TB\_DELETECOL』
- 第204頁的『DTW\_TB\_DLIST』
- 第206頁的『DTW\_TB\_DUMP』
- 第207頁的『DTW\_TB\_DUMPV』
- 第209頁的『DTW\_TB\_GETN』
- 第211頁的『DTW\_TB\_GETV』
- 第213頁的『DTW\_TB\_HTMLENCODE』
- 第214頁的『DTW\_TB\_INPUT\_CHECKBOX』
- 第216頁的『DTW\_TB\_INPUT\_RADIO』
- 第218頁的『DTW\_TB\_INPUT\_TEXT』
- 第220頁的『DTW\_TB\_INSERTCOL』
- 第221頁的『DTW\_TB\_INSERTROW』
- 第222頁的『DTW\_TB\_LIST』
- 第224頁的『DTW\_TB\_QUERYCOLNONJ』
- 第226頁的『DTW\_TB\_ROWS』
- 第227頁的『DTW\_TB\_SELECT』
- 第229頁的『DTW\_TB\_SETCOLS』
- 第230頁的『DTW\_TB\_SETN』
- 第232頁的『DTW\_TB\_SETV』
- 第234頁的『DTW\_TB\_TABLE』
- 第236頁的『DTW\_TB\_TEXTAREA』

## DTW\_TB\_APPENDROW

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將一列或數列新增到 Net.Data 表格尾端。

### 語法

@DTW\_TB\_APPENDROW(table, rows)

### 參數

表 119. DTW\_TB\_APPENDROW 參數

資料類型	參數	使用	說明
表格	table	INOUT	附加橫列的巨集表格變數。
整數	rows	IN	附加至 table 中的橫列數。

### 回覆碼

表 120. DTW\_TB\_APPENDROW 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。
1010	資料會被寫入表格中，直到表格沒有空位為止，剩餘資料則捨棄。

### 用法注意事項

1. 在呼叫 DTW\_TB\_APPENDROW() 之前，須先設定表格中的直欄數目。您可以使用 DTW\_TB\_SETCOLS() 或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。
2. 在表格中附加列之後，您可以使用 DTW\_TB\_SETV() 函數指定新列的值，或將表格傳送到語言環境去處理。
3. 如果表格中有橫列總數的限制，且將附加的列數使得超過這個限制，會有一個錯誤傳回給呼叫者。

### 範例

範例 1：附加十列橫列至表格

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_APPENDROW(myTable, "10")
```

## DTW\_TB\_COLS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回 Net.Data 表格中的直欄數目。

### 語法

@DTW\_TB\_COLS(table, cols)

@DTW\_TB\_rCOLS(table)

### 參數

表 121. DTW\_TB\_COLS 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	傳回直欄數目的巨集表格變數。
整數	<i>cols</i>	OUT	一個包含在 <i>table</i> 中之直欄數目的變數。

### 回覆碼

表 122. DTW\_TB\_COLS 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

**範例 1：**取回直欄的數目，並將值指定到 *cols*

```
%DEFINE myTable = %TABLE
%DEFINE cols = ""
...
@FillTable(myTable)
...
@DTW_TB_COLS(myTable, cols)
```

**範例 2：**取回並顯示表格中現行直欄數目的值

```
%DEFINE myTable = %TABLE
...
@FillTable(myTable)
...
<P>我的表格包含 @DTW_TB_rCOLS(myTable) 個直欄。
```

## DTW\_TB\_DELETECOL

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

刪除 Net.data 表格中一個或多個直欄。

### 語法

```
@DTW_TB_DELETECOL(table, after_col, cols)
```

### 參數

表 123. DTW\_TB\_DELETECOL 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	將刪除其直欄的巨集表格變數。
整數	<i>after_col</i>	IN	將在該直欄後刪除後續直欄的直欄號碼。若要刪除第一個直欄，請指定 0。
整數	<i>cols</i>	IN	從 <i>table</i> 刪除的直欄數目。

### 回覆碼

表 124. DTW\_TB\_DELETECOL 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 範例

**範例 1：**刪除表格中第三及第四個直欄

```
%DEFINE myTable = %TABLE  
  
@DTW_TB_DELETECOL(myTable, "3", "2")
```

**範例 2：**刪除表格中第一個直欄

```
%DEFINE myTable = %TABLE  
  
@DTW_TB_DELETECOL(myTable, "0", "1")
```



## DTW\_TB\_DELETEROW

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

刪除 Net.Data 表格中一系列或數列。

### 語法

@DTW\_TB\_DELETEROW(table, start\_row, rows)

### 參數

表 125. DTW\_TB\_DELETEROW 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	其中橫列被刪除的巨集表格變數。
整數	<i>start_row</i>	IN	<i>table</i> 中將刪除的第一列的橫列號碼。
整數	<i>rows</i>	IN	從 <i>table</i> 所刪除之橫列數。

### 回覆碼

表 126. DTW\_TB\_DELETEROW 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 用法注意事項

呼叫 DTW\_TB\_DELETEROW() 之前，您必須設定表格的直欄數目。您可以使用 DTW\_TB\_SETCOLS() 或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。

### 範例

**範例 1：**刪除表格中第 10 列開始的 5 列。

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_DELETEROW(myTable, "10", "5")
```

**範例 2：**刪除表格中所有的橫列

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_DELETEROW(myTable, "1", @DTW_TB_rROWS(myTable))
```

## DTW\_TB\_DLIST

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格中產生 HTML 定義列示。

### 語法

@DTW\_TB\_DLIST(table, term, def, termstyle, defstyle, link, link\_u, image, image\_u)

@DTW\_TB\_DLIST(table, term, def, termstyle, defstyle, link, link\_u, image)

@DTW\_TB\_DLIST(table, term, def, termstyle, defstyle, link, link\_u)

@DTW\_TB\_DLIST(table, term, def, termstyle, defstyle, link)

@DTW\_TB\_DLIST(table, term, def, termstyle, defstyle)

@DTW\_TB\_DLIST(table, term, def, termstyle)

@DTW\_TB\_DLIST(table, term, def)

@DTW\_TB\_DLIST(table, term)

@DTW\_TB\_DLIST(table)

### 參數

表 127. DTW\_TB\_DLIST 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	指定巨集表格變數來顯示為 HTML 列示的符號。
整數	<i>term</i>	IN	在 <i>table</i> 中的直欄號碼，其包含 <i>term</i> 名稱值（在 <DT> 標籤之後的本文）。預設值是使用第一欄。
整數	<i>def</i>	IN	在 <i>table</i> 中的直欄號碼，其包含 <i>term</i> 定義值（在 <DD> 標籤之後的本文）。預設值是使用第二直欄。
字串	<i>termstyle</i>	IN	一個包含 <i>term</i> 名稱值之 HTML 元素列示的變數或文字字串。預設值是使用沒有樣式的標籤。
字串	<i>defstyle</i>	IN	一個包含 <i>term</i> 定義值之 HTML 元素列示的變數或文字字串。預設值是使用沒有樣式的標籤。
字串	<i>link</i>	IN	指定哪一個 HTML 元素是由 HTML 鏈結所產生的。有效的值有 DT 和 DD。預設值是不產生 HTML 鏈結。
整數	<i>link_u</i>	IN	在包含 HTML 參照的 URL 之 <i>table</i> 中的欄號碼。預設值是不產生 HTML 鏈結。
字串	<i>image</i>	IN	指定列入壓縮檔所產生的是哪一個 HTML 元素。有效的值有 DT 和 DD。預設值是不產生列入壓縮檔（DT）。

表 127. DTW\_TB\_DLIST 參數 (繼續)

資料類型	參數	使用	說明
整數	<i>image_u</i>	IN	在包含列入壓縮檔之 URL 之 <i>table</i> 中的欄號碼。預設值是不產生列入壓縮檔。

## 回覆碼

表 128. DTW\_TB\_DLIST 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

**範例 1：**依據表格資料，建立一個產生下列 HTML 的定義列表

```
@DTW_TB_DLIST(Mytable,"3","4","b i","strong","DD","2","DT","1")
```

結果：

```
<DL>
<DT>
<IMG SRC="http://www.mycompany.com/images/image1.gif" ALT=""><b><i>image1text</i></b>
<DD>
<A HREF="http://www.mycompany.com/link1.html"><strong>link1text</strong></A>
<DT>
<IMG SRC="http://www.mycompany.com/images/image2.gif" ALT=""><b><i>image2text</i></b>
<DD>
<A HREF="http://www.mycompany.com/link2.html"><strong>link2text</strong></A>
<DT>
<IMG SRC="http://www.mycompany.com/images/image3.gif" ALT=""><b><i>image3text</i></b>
<DD>
<A HREF="http://www.mycompany.com/link3.html"><strong>link3text</strong></A>
<DT>
<IMG SRC="http://www.mycompany.com/images/image4.gif" ALT=""><b><i>image4text</i></b>
<DD>
<A HREF="http://www.mycompany.com/link4.html"><strong>link4text</strong></A>
</DT>
</DL>
```

## DTW\_TB\_DUMPH

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

使用 HTML <PRE> 標籤印出 Net.Data 表格的內容，使表格每一列顯示成一行。

### 語法

@DTW\_TB\_DUMPH(table)

### 參數

表 129. DTW\_TB\_DUMPH 參數

資料類型	參數	使用	說明
表格	table	IN	指定要顯示之巨集表格變數的符號。

### 回覆碼

表 130. DTW\_DB\_DUMPH 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1007	參數包含的值無效。

### 用法注意事項

如果 Net.Data 表格是空的，則傳回錯誤。

### 範例

#### 範例 1：

@DTW\_TB\_DUMPH(Mytable)

由這個範例所產生的 HTML 看起來類似這個：

```
<PRE>
名稱          部門          職位
Jack Smith    Internet 技術    軟體工程師
Helen Williams 資料庫          開發部經理
Alex Jones    製造            工業工程師
Tom Baker     採購            業務代表
</PRE>
```

## DTW\_TB\_DUMPV

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

使用 HTML <PRE> 標籤印出 Net.Data 表格的內容，使表格每一個欄位各在一行。

### 語法

@DTW\_TB\_DUMPV(table)

### 參數

表 131. DTW\_TB\_DUMPV 參數

資料類型	參數	使用	說明
表格	table	IN	指定要顯示之巨集表格變數的符號。

### 回覆碼

表 132. DTW\_TB\_DUMPV 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1007	參數包含的值無效。

### 用法注意事項

如果 Net.Data 表格是空的，則傳回錯誤。

### 範例

#### 範例 1：

@DTW\_TB\_DUMPV(Mytable)

由這個範例所產生的 HTML 看起來類似這個：

```
<PRE>
http://www.mycompany.com/images/image1.gif
http://www.mycompany.com/link1.html
image1text
link1text
http://www.mycompany.com/images/image2.gif
http://www.mycompany.com/link2.html
image2text
link2text
http://www.mycompany.com/images/image3.gif
http://www.mycompany.com/link3.html
image3text
link3text
http://www.mycompany.com/images/image4.gif
```

```
http://www.mycompany.com/link4.html  
image4text  
link4text  
</PRE>
```

## DTW\_TB\_GETN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格傳回欄位標題。

### 語法

@DTW\_TB\_GETN(table, col, name)

@DTW\_TB\_rGETN(table, col)

### 參數

表 133. DTW\_TB\_GETN 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	傳回一個直欄名稱的巨集表格變數。
整數	<i>col</i>	IN	將被傳回名稱的直欄的直欄號碼。
字串	<i>name</i>	OUT	包含在 <i>col</i> 中被指定之直欄名稱的變數。

### 回覆碼

表 134. DTW\_TB\_GETN 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 用法注意事項

呼叫 DTW\_TB\_GETN() 之前，請設定表格的直欄數目。您可以使用 DTW\_TB\_SETCOLS()或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。

### 範例

範例 1：取回直欄 4 的直欄名稱

```
%DEFINE myTable = %TABLE
%DEFINE name = ""
...
@FillTable(myTable)
...
@DTW_TB_GETN(myTable, "4", name)
```

**範例 2：**取回表格中最後一欄的直欄名稱

```
%DEFINE myTable = %TABLE
...
@FillTable(myTable)
...
<P>最後一欄的直欄名稱是 @DTW_TB_rGETN(myTable, @DTW_TB_rCOLS(myTable))
```



## DTW\_TB\_GETV

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回 Net.Data 表格中給定的橫列與直欄之值。

### 語法

@DTW\_TB\_GETV(table, row, col, value)

@DTW\_TB\_rGETV(table, row, col)

### 參數

表 135. DTW\_TB\_GETV 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	傳回表格值的巨集表格變數。
整數	<i>row</i>	IN	傳回值的列號。
整數	<i>col</i>	IN	傳回值的欄號。
字串	<i>value</i>	OUT	包含 <i>row</i> 和 <i>col</i> 所指定的橫列與直欄值的變數。

### 回覆碼

表 136. DTW\_TB\_GETV 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 用法注意事項

呼叫 DTW\_TB\_GETV() 之前，請設定表格的直欄數目。您可以使用 DTW\_TB\_SETCOLS()或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。

### 範例

範例 1：取回第 6 列、第 3 欄的表格值

```

%DEFINE myTable = %TABLE
%DEFINE value = ""
...
@FillTable(myTable)
...
@DTW_TB_GETV(myTable, "6", "3", value)

```

**範例 2：**取回橫列 1，直欄 1 的值

```

%DEFINE myTable = %TABLE
...
@FillTable(myTable)
...

```

<P>橫列 1，直欄 1 的表格值是 @DTW\_TB\_rGETV(myTable, "1", "1")。

## DTW\_TB\_HTMLENCODE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

以對應的 HTML 字元跳出碼置換 Net.Data 表格中資料的某些字元。

### 語法

@DTW\_TB\_HTMLENCODE(table, collist)

@DTW\_TB\_HTMLENCODE(table)

### 參數

表 137. DTW\_TB\_HTMLENCODE 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	所要修改的巨集表格變數。
字串	<i>collist</i>	IN	在 <i>table</i> 中所要編碼的直欄號碼。預設值是對所有的直欄進行編碼。

### 回覆碼

表 138. DTW\_TB\_HTMLENCODE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 用法注意事項

下表列出被置換的字元。

名稱	字元	程式碼
& 記號	&	&#38;
雙引號	"	&#34;
大於	>	&#62;
小於	<	&#60;

### 範例

#### 範例 1：

```
@DTW_TB_HTMLENCODE(Mytable, "3 4")
```

在指定的表格之直欄 3 和直欄 4 中的特殊字元被它們的編碼格式所置換。

## DTW\_TB\_INPUT\_CHECKBOX

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格產生一或數個 HTML 核對框輸入標籤。

### 語法

@DTW\_TB\_INPUT\_CHECKBOX(table, prompt, namecol, valuecol, rows, checkedrows)

@DTW\_TB\_INPUT\_CHECKBOX(table, prompt, namecol, valuecol, rows)

@DTW\_TB\_INPUT\_CHECKBOX(table, prompt, namecol, valuecol)

@DTW\_TB\_INPUT\_CHECKBOX(table, prompt, namecol)

### 參數

表 139. DTW\_TB\_INPUT\_CHECKBOX 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	顯示為核對框輸入標籤的巨集表格變數。
字串	<i>prompt</i>	IN	在 <i>table</i> 中的欄號碼，或包含顯示在核對框旁之文字的字串。這個是必要的參數，但可以具有一個空 ("" ) 值。當 <i>prompt</i> 是空值的時候，所使用的值是 <i>namecol</i> 所定義的值。
字串	<i>namecol</i>	IN	在 <i>table</i> 中的直欄號碼或包含輸入欄位名稱的字串。
整數	<i>valuecol</i>	IN	<i>table</i> 中包含輸入欄位值的直欄號碼。預設值是 1。
整數	<i>rows</i>	IN	在產生輸入欄位之 <i>table</i> 之列的列示。預設值是使用所有的列。
整數	<i>checkedrows</i>	IN	指定所要檢查之 <i>table</i> 之列的列示。預設值是不檢查欄位。

### 回覆碼

表 140. DTW\_TB\_INPUT\_CHECKBOX 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

範例 1：產生三個核對框輸入標籤的 HTML

```
@DTW_TB_INPUT_CHECKBOX(Mytable,"3","4","","2 3 4","1 3 4")
```

結果：

```
<INPUT TYPE="CHECKBOX" NAME="link2text" VALUE="1">image2text<BR>
<INPUT TYPE="CHECKBOX" NAME="link3text" VALUE="1" CHECKED>image3text<BR>
<INPUT TYPE="CHECKBOX" NAME="link4text" VALUE="1" CHECKED>image4text<BR>
```

## DTW\_TB\_INPUT\_RADIO

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格產生 HTML 圓鈕輸入標籤。

### 語法

@DTW\_TB\_INPUT\_RADIO(table, prompt, namecol, valuecol, rows, checkedrows)

@DTW\_TB\_INPUT\_RADIO(table, prompt, namecol, valuecol, rows)

@DTW\_TB\_INPUT\_RADIO(table, prompt, namecol, valuecol)

### 參數

表 141. DTW\_TB\_INPUT\_RADIO 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	顯示為圓鈕輸入標籤的巨集表格變數。
字串	<i>prompt</i>	IN	在 <i>table</i> 中或包含顯示於圓鈕旁邊之文字字串的直欄號碼。必要的參數，但是可以包含一個空 ("" ) 值。當 <i>prompt</i> 是空值時，請使用 <i>valuecol</i> 的值。
字串	<i>namecol</i>	IN	在 <i>table</i> 中的直欄號碼或包含輸入欄位名稱的字串。
整數	<i>valuecol</i>	IN	<i>table</i> 中包含輸入欄位值的直欄號碼。
字串	<i>rows</i>	IN	在產生輸入欄位之 <i>table</i> 之列的列示。預設值是使用所有的列。
整數	<i>checkedrows</i>	IN	在 <i>table</i> 中的列號碼顯示所勾選之相對應的圓鈕。只有一個值是被允許的。

### 回覆碼

表 142. DTW\_TB\_INPUT\_RADIO 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

### 範例 1：產生三個圓鈕輸入標籤的 HTML

```
@DTW_TB_INPUT_RADIO(Mytable,"3","Radio4","4","2 3 4","4")
```

結果：

```
<INPUT TYPE="RADIO" NAME="Radio4" VALUE="link2text">image2text<BR>
<INPUT TYPE="RADIO" NAME="Radio4" VALUE="link3text">image3text<BR>
<INPUT TYPE="RADIO" NAME="Radio4" VALUE="link4text" CHECKED>image4text<BR>
```

## DTW\_TB\_INPUT\_TEXT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

產生 Net.Data 表格中指定列的 HTML <INPUT> 標籤。

### 語法

@DTW\_TB\_INPUT\_TEXT(table, prompt, namecol, valuecol, size, maxlen, rows)

@DTW\_TB\_INPUT\_TEXT(table, prompt, namecol, valuecol, size, maxlen)

@DTW\_TB\_INPUT\_TEXT(table, prompt, namecol, valuecol, size)

@DTW\_TB\_INPUT\_TEXT(table, prompt, namecol, valuecol)

@DTW\_TB\_INPUT\_TEXT(table, prompt, namecol)

### 參數

表 143. DTW\_TB\_INPUT\_TEXT 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	顯示為文字輸入標籤的巨集表格變數。
字串	<i>prompt</i>	IN	在 <i>table</i> 中的欄號碼或包含顯示在輸入欄旁之文字的字串。如果提示是空值，則不會顯示文字。
字串	<i>namecol</i>	IN	<i>table</i> 中包含輸入欄位名稱的直欄號碼。
整數	<i>valuecol</i>	IN	<i>table</i> 中包含預設輸入欄位值的直欄號碼，它是指定給 INPUT 標籤上的 VALUE 屬性。預設值是不產生 VALUE 屬性值。
整數	<i>size</i>	IN	輸入欄位的字元號碼，其為 INPUT 標籤上 SIZE 屬性所指定的。預設值是最長預設輸入值的長度，或者如果沒有預設輸入時則是 10。
整數	<i>maxlen</i>	IN	輸入字串的最大長度，是為 INPUT 標籤之 MAXLENGTH 屬性所指定的。預設值是不產生 MAXLENGTH 屬性值。
整數	<i>rows</i>	IN	在產生輸入欄位之 <i>table</i> 之列的列示。預設值是使用所有的列。

### 回覆碼

表 144. DTW\_TB\_INPUT\_TEXT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。



表 144. DTW\_TB\_INPUT\_TEXT 回覆碼 (繼續)

回覆碼	說明
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

**範例 1：**傳回三個 HTML <INPUT> 標籤

```
@DTW_TB_INPUT_TEXT(Mytable,"3","3","4","35","40","1 2 3")
```

結果：

```
<P>image1text
<INPUT TYPE="TEXT" NAME="image1text" VALUE="link1text" SIZE="35" MAXLENGTH="40">
<P>image2text
<INPUT TYPE="TEXT" NAME="image2text" VALUE="link2text" SIZE="35" MAXLENGTH="40">
<P>image3text
<INPUT TYPE="TEXT" NAME="image3text" VALUE="link3text" SIZE="35" MAXLENGTH="40">
```

## DTW\_TB\_INSERTCOL

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將一或數個直欄插入 Net.Data 表格。

### 語法

```
@DTW_TB_INSERTCOL(table, after_col, cols)
```

### 參數

表 145. DTW\_TB\_INSERTCOL 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	其中將插入直欄的巨集表格變數。
整數	<i>after_col</i>	IN	直欄號碼，將在該欄之後插入新直欄。若要在表格起始處插入直欄，請指定 0。
整數	<i>cols</i>	IN	將插入 <i>table</i> 中的直欄數目。

### 回覆碼

表 146. DTW\_TB\_INSERTCOL 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 範例

**範例 1：**在表格最後插入 5 欄直欄

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_INSERTCOL(myTable, @DTW_TB_rCOLS(myTable), "5")
```

**範例 2：**在表格起始處插入一直欄

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_INSERTCOL(myTable, "0", "1")
```

## DTW\_TB\_INSERTROW

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

將一列或數列插入 Net.Data 表格。

### 語法

@DTW\_TB\_INSERTROW(table, after\_row, rows)

### 參數

表 147. DTW\_TB\_INSERTROW 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	其中將插入橫列的巨集表格變數。
整數	<i>after_row</i>	IN	橫列號碼，在該列後將插入新的橫列。若要在表格起始處插入橫列，請指定 0。
整數	<i>rows</i>	IN	插入 <i>table</i> 中之橫列數。

### 回覆碼

表 148. DTW\_TB\_INSERTROW 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 用法注意事項

呼叫 DTW\_TB\_INSERTROW() 之前，請設定表格的直欄數目。您可以使用 DTW\_TB\_SETCOLS() 或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。

### 範例

**範例 1：**在表格中的第 5 橫列後插入一橫列

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_INSERTROW(myTable, "5", "1")
```

**範例 2：**在表格起始處插入 3 列橫列

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_INSERTROW(myTable, "0", "3")
```

## DTW\_TB\_LIST

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格產生 HTML 列示。

### 語法

@DTW\_TB\_LIST(table, listtype, listitem, itemstyle, link\_u, image\_u)

@DTW\_TB\_LIST(table, listtype, listitem, itemstyle, link\_u)

@DTW\_TB\_LIST(table, listtype, listitem, itemstyle)

@DTW\_TB\_LIST(table, listtype, listitem)

@DTW\_TB\_LIST(table, listtype)

### 參數

表 149. DTW\_TB\_LIST 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	指定巨集表格變數顯示當作 HTML 列示的符號。
字串	<i>listtype</i>	IN	所要產生的列示類型。可接受的值包括： DIR MENU OL UL
整數	<i>listitem</i>	IN	在 <i>table</i> 中的直欄號碼包含列表值（在 <LI> 標籤之後的本文）。預設值是使用第一欄。
字串	<i>itemstyle</i>	IN	一個包含詞彙名稱值之 HTML 元素列示的變數或文字字串。預設值是使用沒有樣式的標籤。
整數	<i>link_u</i>	IN	在包含 HTML 鏈結的 URL 之 <i>table</i> 中的欄號碼。如果並未指定這個值，則將不會產生 HTML 鏈結。
整數	<i>image_u</i>	IN	在包含列入壓縮檔之 URL 之 <i>table</i> 中的欄號碼。如果並未指定這個值，則將不會產生列入壓縮檔。

### 回覆碼

表 150. DTW\_TB\_LIST 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。

表 150. DTW\_TB\_LIST 回覆碼 (繼續)

回覆碼	說明
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

**範例 1：**產生一個次序列表的 HTML 標籤

```
@DTW_TB_LIST(Mytable,"OL","4","TT U","2","1")
```

結果：

```
<TT><U>
<OL>
<LI><A HREF="http://www.mycompany.com/link1.html">
<IMG SRC="http://www.mycompany.com/images/image1.gif" ALT="">link1text</A>
<LI><A HREF="http://www.mycompany.com/link2.html">
<IMG SRC="http://www.mycompany.com/images/image2.gif" ALT="">link2text</A>
<LI><A HREF="http://www.mycompany.com/link3.html">
<
IMG SRC="http://www.mycompany.com/images/image3.gif" ALT="">link3text</A>
<LI><A HREF="http://www.mycompany.com/link4.html">
<IMG SRC="http://www.mycompany.com/images/image4.gif" ALT="">link4txt</A>
</OL>
</U></TT>
```

## DTW\_TB\_QUERYCOLNONJ

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回與 Net.Data 表格欄位標題相關的直欄號碼。

### 語法

@DTW\_TB\_QUERYCOLNONJ(table, name, col)

@DTW\_TB\_rQUERYCOLNONJ(table, name)

### 參數

表 151. DTW\_TB\_QUERYCOLNONJ 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	將傳回直欄數目的巨集表格變數。
字串	<i>name</i>	IN	將傳回其直欄號碼的欄位標題的名稱。若表格中不存在欄位標題，將傳回 0。
整數	<i>col</i>	OUT	含有直欄號碼的變數，此直欄的名稱是在 <i>name</i> 中指定的。

### 回覆碼

表 152. DTW\_TB\_QUERYCOLNONJ 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

1. 呼叫 DTW\_TB\_QUERYCOLNONJ() 之前，請設定表格的直欄數目。您可以使用 DTW\_TB\_SETCOLS() 或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。
2. 若表格中不存在欄位標題，將傳回 0。

### 範例

**範例 1：**取回名稱爲 SERIAL\_NUMBER 的直欄之直欄號碼

```
%DEFINE myTable = %TABLE  
%DEFINE col = ""
```

```
@DTW_TB_QUERYCOLNONJ(myTable, "SERIAL_NUMBER", col)
```

**範例 2：**取回名稱爲 SERIAL\_NUMBER 的直欄之直欄號碼

```
%DEFINE myTable = %TABLE  
<P>The "SERIAL_NUMBER" column is column number @DTW_TB_rQUERYCOLNONJ(myTable, "SERIAL_NUMBER")
```

## DTW\_TB\_ROWS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

傳回 Net.Data 表格中的橫列數目。

### 語法

@DTW\_TB\_ROWS(table, rows)

@DTW\_TB\_rROWS(table)

### 參數

表 153. DTW\_TB\_ROWS 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	傳回現行列數的巨集表格變數。
整數	<i>rows</i>	OUT	包含在 <i>table</i> 中之現行列數的變數。

### 回覆碼

表 154. DTW\_TB\_ROWS 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 範例

**範例 1：**取回表格中的現行列數，並將值指定到 *rows*

```
%DEFINE myTable = %TABLE
%DEFINE rows = ""
...
@FillTable(myTable)
...
@DTW_TB_ROWS(myTable, rows)
```



## DTW\_TB\_SELECT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格產生 HTML 選項列示。

### 語法

@DTW\_TB\_SELECT(table, name, optioncol, size, multiple, rows, selectedrows, valuecol)

@DTW\_TB\_SELECT(table, name, optioncol, size, multiple, rows, selectedrows)

@DTW\_TB\_SELECT(table, name, optioncol, size, multiple, rows)

@DTW\_TB\_SELECT(table, name, optioncol, size, multiple)

@DTW\_TB\_SELECT(table, name, optioncol, size)

@DTW\_TB\_SELECT(table, name, optioncol)

@DTW\_TB\_SELECT(table, name)

### 參數

表 155. DTW\_TB\_SELECT 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	顯示為 SELECT 欄位的巨集表格變數。
字串	<i>name</i>	IN	SELECT 欄位之 NAME 屬性的值。
整數	<i>optioncol</i>	IN	在 <i>table</i> （該表格具有使用在 SELECT 欄位中之 OPTION 標籤中的值）中的直欄號碼。預設值是使用第一欄。
整數	<i>size</i>	IN	在使用作為在 SELECT 欄位中之 OPTION 標籤之 <i>table</i> 中的列號碼。預設值是使用所有的列。
字串	<i>multiple</i>	IN	指定是否允許所作的多重選擇。預設值是 N，表示不允許多重選擇。
字串	<i>rows</i>	IN	從 <i>table</i> 選取，要使用在 SELECT 欄位中的列。預設值是使用所有的列。
字串	<i>selectedrows</i>	IN	從已檢查過標籤之表格中選取列的列示。若要指定多於一行，您必須將多重參數設定為 Y。預設值是選取第一個項目。
字串	<i>valuecol</i>	IN	表格中用於 OPTION 標籤的 VALUE 屬性之直欄號碼。這個參數是可選用的。

### 回覆碼

表 156. DTW\_TB\_SELECT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。

表 156. DTW\_TB\_SELECT 回覆碼 (繼續)

回覆碼	說明
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

**範例 1：**產生含有多重選擇的 HTML SELECT 功能表

```
@DTW_TB_SELECT(Mytable,"URL6","3","", "y", "1 2 4", "1 4")
```

結果：

```
<SELECT NAME="URL6" SIZE="3" MULTIPLE>
<OPTION SELECTED>image1text
<OPTION>image2text
<OPTION SELECTED>image4text
</SELECT>
```

**範例 2：**使用 *valuecol* 參數產生一個 HTML SELECT 功能表，此功能表使用直欄號碼來取得一些值。

結果：

```
<SELECT NAME="URL6" SIZE="3" MULTIPLE>
<OPTION VALUE="1" SELECTED>image1text
<OPTION VALUE="2">image2text
<OPTION VALUE="3" SELECTED>image3text
</SELECT>
```

## DTW\_TB\_SETCOLS

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

設定 Net.Data 表格中的直欄數目。

### 語法

@DTW\_TB\_SETCOLS(table, cols)

### 參數

表 157. DTW\_TB\_SETCOLS 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	設定直欄數目的巨集表格變數。
整數	<i>cols</i>	IN	<i>table</i> 中配置的起始直欄數目。

### 回覆碼

表 158. DTW\_TB\_SETCOLS 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。

### 用法注意事項

1. 一個表格僅能使用一次 DTW\_TB\_SETCOLS() 函數。然後，使用 DTW\_TB\_DELETECOL() 或 DTW\_TB\_INSERTCOL() 函數，變更表格中的直欄數目。
2. 使用 DTW\_TB\_SETN() 函數指定欄位標題。

### 範例

**範例 1：**配置三欄供表格使用，並指定直欄的名稱

```
%DEFINE myTable = %TABLE

@DTW_TB_SETCOLS(myTable, "3")
@DTW_TB_SETN(myTable, "Name", "1")
@DTW_TB_SETN(myTable, "Address", "2")
@DTW_TB_SETN(myTable, "Phone", "3")
```

## DTW\_TB\_SETN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定名稱給 Net.Data 中的欄位標題。

### 語法

@DTW\_TB\_SETN(table, name, col)

### 參數

表 159. DTW\_TB\_SETN 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	將設定直欄名稱的巨集表格變數。
字串	<i>name</i>	IN	指定為 <i>col</i> 中所指定直欄之欄位標題的字串。
整數	<i>col</i>	IN	將設定標題之直欄的直欄號碼。

### 回覆碼

表 160. DTW\_TB\_SETN 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 用法注意事項

1. 呼叫 DTW\_TB\_SETN() 之前，請設定表格的直欄數目。您可以使用 DTW\_TB\_SETCOLS() 或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。
2. 若要刪除欄位標題，請將欄位標題值指定為 NULL。

### 範例

範例 1：指定欄位標題 1 到 3 的名稱

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_SETCOLS(myTable, "3")
@DTW_TB_SETN(myTable, "Name", "1")
@DTW_TB_SETN(myTable, "Address", "2")
@DTW_TB_SETN(myTable, "Phone", "3")
```

**範例 2：**刪除直欄 2 的欄位標題。經由在函數呼叫上傳遞一個未定義的變數，即可達成此目的。依據預設值，此變數將具有 NULL 值。

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_SETN(myTable, nullVar, "2")
```

## DTW\_TB\_SETV

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

指定一個值給 Net.Data 表格中特定列與直欄。

### 語法

@DTW\_TB\_SETV(table, value, row, col)

### 參數

表 161. DTW\_TB\_SETV 參數

資料類型	參數	使用	說明
表格	<i>table</i>	INOUT	將設定表格值的巨集表格變數。
字串	<i>value</i>	IN	指定為 <i>row</i> 與 <i>col</i> 中指定的橫列與直欄的表格值的字串。
整數	<i>row</i>	IN	將設定值的橫列號碼。
整數	<i>col</i>	IN	將設定值的直欄號碼。

### 回覆碼

表 162. DTW\_TB\_SETV 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

### 用法注意事項

1. 呼叫 DTW\_TB\_SETV() 之前，請設定表格的直欄數目。您可以使用 DTW\_TB\_SETCOLS() 或 DTW\_TB\_INSERTCOL() 函數，或將表格傳送到將被設定的語言環境中來設定直欄的數目。
2. 若要刪除表格值，請將該值指定為 NULL。

### 範例

**範例 1：**指定第 3 列第 3 欄的值

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_SETV(myTable, "value3.3", "3", "3")
```

**範例 2：**刪除第 4 列、第 2 欄的表格值。經由在函數呼叫上傳遞一個未定義的變數，即可完成此刪除。依據預設值，此變數將具有 NULL 值。

```
%DEFINE myTable = %TABLE
```

```
@DTW_TB_SETV(myTable, nullVar, "4", "2")
```

## DTW\_TB\_TABLE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格產生 HTML 表格。

### 語法

@DTW\_TB\_TABLE(table, options, collist, cellstyle, link\_u, image\_u, url\_text, url\_style)

@DTW\_TB\_TABLE(table, options, collist, cellstyle, link\_u, image\_u, url\_text)

@DTW\_TB\_TABLE(table, options, collist, cellstyle, link\_u, image\_u)

@DTW\_TB\_TABLE(table, options, collist, cellstyle, link\_u)

@DTW\_TB\_TABLE(table, options, collist, cellstyle)

@DTW\_TB\_TABLE(table, options, collist)

@DTW\_TB\_TABLE(table, options)

@DTW\_TB\_TABLE(table)

### 參數

表 163. DTW\_TB\_TABLE 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	顯示為 HTML 表格的巨集表格變數。
字串	<i>options</i>	IN	位於 TABLE 標籤中的表格屬性。預設值示不使用屬性。有效的值包括： <ul style="list-style-type: none"><li>• BORDER</li><li>• CELSPACING</li><li>• WIDTH</li></ul>
字串	<i>collist</i>	IN	使用在 HTML 表格中之 <i>table</i> 的欄號碼。預設值是使用所有的欄。
字串	<i>cellstyle</i>	IN	HTML 樣式元素的列示，例如 B 和 I，是圍繞在每個 TD 標籤旁的文字。預設值是不使用樣式標籤。
整數	<i>link_u</i>	IN	在包含使用來建立 HTML 鏈結之 URL 的 <i>table</i> 中的欄號碼。您必需也在 <i>collist</i> 中指定欄。預設值是不產生 HTML 鏈結。
整數	<i>image_u</i>	IN	在包含使用來建立列入壓縮檔之 URL 的 <i>table</i> 中的欄號碼。您必需也在 <i>collist</i> 中指定欄。預設值是不產生壓縮檔標籤。
整數	<i>url_text</i>	IN	在 <i>table</i> 中包含顯示 HTML 鏈結或列入壓縮檔的文字的欄號碼。預設值是使用 URL 它本身。
字串	<i>url_style</i>	IN	指定在 <i>url_text</i> 中文字之 HTML 樣式元素的列示。預設值是不產生樣式標籤。



## 回覆碼

表 164. DTW\_TB\_TABLE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

**範例 1：**使用邊框並使用 B（粗體）與 I（斜體字）來產生表格的 HTML 標籤

```
@DTW_TB_TABLE(Mytable,"BORDER","4 2 1","i","2","1","4","b")
```

結果：

```
<TABLE BORDER>
  <TR>
    <TH>TITLE
    <TH>LINKURL
    <TH>IMAGEURL
  <TR>
    <TD><i>link1text</i>
    <TD><A HREF="http://www.mycompany.com/link1.html"><b>link1text</b></A>
    <TD><IMG SRC="http://www.mycompany.com/images/image1.gif" ALT=""><b>link1text</b>
  <TR>
    <TD><i>link2text</i>
    <TD><A HREF="http://www.mycompany.com/link2.html"><b>link2text</b></A>
    <TD><IMG SRC="http://www.mycompany.com/images/image2.gif" ALT=""><b>link2text</b>
  <TR>
    <TD><i>link3text</i>
    <TD><A HREF="http://www.mycompany.com/link3.html"><b>link3text</b></A>
    <TD><IMG SRC="http://www.mycompany.com/images/image3.gif" ALT=""><b>link3text</b>
</TABLE>
```

## DTW\_TB\_TEXTAREA

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X	X

### 目的

從 Net.Data 表格產生 HTML 本文區域。

### 語法

@DTW\_TB\_TEXTAREA(table, name, numrows, numcols, valuecol, rows)

@DTW\_TB\_TEXTAREA(table, name, numrows, numcols, valuecol)

@DTW\_TB\_TEXTAREA(table, name, numrows, numcols)

@DTW\_TB\_TEXTAREA(table, name, numrows)

@DTW\_TB\_TEXTAREA(table, name)

### 參數

表 165. DTW\_TB\_TEXTAREA 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	一個顯示為 TEXTAREA 標籤的巨集表格變數。
字串	<i>name</i>	IN	文字區域的名稱。
整數	<i>numrows</i>	IN	指定在列中之本文區域的高度。預設值是在 <i>table</i> 中的列數。
整數	<i>numcols</i>	IN	指定在欄中之本文區域的寬度。預設值是在 <i>table</i> 中的最長列的長度。
整數	<i>valuecol</i>	IN	其值顯示在本文區域中的 <i>table</i> 直欄號碼。預設值是第一直欄。
字串	<i>rows</i>	IN	在 <i>table</i> 中用來產生 TEXTAREA 標籤的列。預設值是使用所有的列。

### 回覆碼

表 166. DTW\_TB\_TEXTAREA 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
1008	參數超出表格界限。

## 範例

**範例 1：**產生 HTML TEXTAREA 標籤，並指定要併入哪些列

```
@DTW_TB_TEXTAREA(Mytable,"textarea5","3","70","4","1 3 4")
```

結果：

```
<TEXTAREA NAME="textarea5" ROWS="3" COLS="70">  
link1text  
link3text  
link4text  
<TEXTAREA>
```

---

## 純本文檔介面函數

純本文檔介面（FFI）可以讓您從純本文檔來源（文字檔）來開啓、讀取、並操作，也可以用純本文檔儲存資料。下列是可用的純本文檔介面內建函數：

- 第242頁的『DTWF\_APPEND』
- 第244頁的『DTWF\_CLOSE』
- 第245頁的『DTWF\_DELETE』
- 第249頁的『DTWF\_OPEN』
- 第251頁的『DTWF\_READ』
- 第254頁的『DTWF\_REMOVE』
- 第255頁的『DTWF\_SEARCH』
- 第258頁的『DTWF\_UPDATE』
- 第260頁的『DTWF\_WRITE』

下列各節將描述如何使用 FFI 內建函數及存取純本文檔來源：

- 『存取純本文檔資料來源』
- 第240頁的『純本文檔介面區隔字元』
- 第241頁的『鎖定檔案』

### 存取純本文檔資料來源

您可以在 `Net.Data` 起始設定檔中使用 `FFI_PATH` 路徑架構陳述式，來列出當使用 FFI 函數時容許設定的目錄與子目錄，以及保護那些不在路徑陳述式所包括的目錄中的檔案。`Net.Data` 起始設定檔沒有附 `FFI_PATH`。請參閱 *Net.Data 管理與程式設計手冊*，來學習如何架構路徑。

`FFI_PATH` 會使用下列語法：

```
FFI_PATH /path1;/path2;/path3...
```

當您在巨集函數中呼叫 FFI 語言環境時，您可使用 FFI 函數的 `filename` 參數，來指定 FFI 函數將使用的純本文檔的路徑。例如：

```
%DEFINE myfile = "/macros/myfile.txt" @DTWF_READ(myfile, ...)
```

下列各節討論：

- 『`Net.Data` 如何決定純本文檔位置』
- 第239頁的『純本文檔架構規則』
- 第240頁的『安全建議』
- 第240頁的『授權需求』

### Net.Data 如何決定純本文檔位置

`Net.Data` 會使用 FFI 函數的 `filename` 參數中的資訊，在 `Net.Data` 起始設定檔中搜尋 `FFI_PATH` 陳述式，並決定是否要使用指定的目錄或現行目錄。

當在 FFI 函數上指定了一個檔名，Net.Data 便會嘗試搜尋 FFI\_PATH 中所列的每一路徑，從指定的第一個路徑開始搜尋，找出檔案的位置。Net.Data 會使用它找到的第一個拷貝。如果找不到檔案，Net.Data 會嘗試在 Net.Data 執行所在的處理或緒的現行工作目錄中，尋找檔案。

**範例：** Net.Data 使用 FFI\_PATH 架構陳述式來找出檔案位置

FFI\_PATH 含有下列目錄：

FFI\_PATH /macros;/macros/org1;/macros/org2

而且，檔案位於現行目錄與 /macros/org1 內。若函數呼叫是：

DTWF\_READ("myfile.txt")

Net.Data 將使用 /macros/org1/myfile.txt。

如果將使用 DTWF\_READ 函數來讀取舊有檔案，且指定了檔名 myfile.txt，Net.Data 將搜尋目錄 /macros、/macros/org1 與 /macros/org2 來找出檔案，假定 FFI\_PATH 含有上面指定的路徑的列示。

#### **判斷現行目錄：**

Net.Data 的現行目錄視您的 Web 伺服器的架構而定：

- 如果您使用 CGI，現行目錄即是執行 Net.Data 的目錄。
- 如果您使用 Web 伺服器 API，則現行目錄會有所不同。如果變更伺服器的預設要求遞送或資源對映，可能也會變更現行目錄。

#### **指定純本文檔存取時的建議**

請使用下列建議，以確定 Net.Data 可以存取純本文檔資料來源。

- 當使用 DTWF\_OPEN 函數建立純本文檔時，請確定您指定的目錄路徑是在 FFI\_PATH 中，或確定知道哪一個是現行目錄。如果您未指定目錄，Net.Data 會嘗試在現行工作目錄中建立檔案。
- 如果您在 *filename* 參數中包括目錄，請指定符合 FFI\_PATH 中路徑之一的完整路徑，因為 Net.Data 不會搜尋 FFI\_PATH 中指定的目錄內的次目錄。
- 請對 *filename* 參數使用絕對路徑，尤其當您使用 Web 伺服器 API 之時。

#### **純本文檔架構規則**

當新增或更新 Net.Data 起始設定檔中的 FFI\_PATH 時，請使用下列規則：

- FFI\_PATH 中的路徑陳述式須含有有效的可列印字元。FFI 不容許路徑中有問號 (?) 或雙引號 ("")。
- 與巨集中 *filename* 參數一起使用的所有目錄與次目錄均須在 FFI\_PATH 中指定。不會搜尋 *filename* 中所列的路徑的次目錄，除非在 FFI\_PATH 中明確地設定它們。
- 請對 FFI\_PATH 陳述式使用絕對路徑。

## 安全建議

您可以指定 FFI 函數可透過 Net.Data 起始設定檔中的 FFI\_PATH 陳述式存取的檔案。FFI 僅會搜尋陳述式中所列的路徑，所以其它目錄中的檔案將受到保護，因為若沒有權限，將無法存取它們。

例如，您可以指定類似於下列的 FFI\_PATH，指出公用或客人使用者 ID 的目錄。

```
FFI_PATH      C:\public;E:\WWW;E:\guest;A:
```

下列列示提供在保護純本文檔時的建議：

- 選擇適於純本文檔作業使用的目錄。這些目錄需要新增到 FFI\_PATH，以限制搜尋那些目錄。
- 不要輕易讓他人巨集中執行 DTWF\_REMOVE 或其它匯出作業，如此可避免他人除去或變更可能已在現行目錄中具有副檔名 .dll 及 .cmd 的檔案。
- 採取適當的步驟，以合理方式控制將新增到系統中的巨集，來保護系統上的檔案。
- 請勿在 FFI\_PATH 中設定一個路徑，讓匿名 FTP 使用者可寫入到該路徑中。若設定如此的路徑，某人可將 Net.Data 巨集置於系統上，來容許先前不被容許的動作。
- 請勿將 Net.Data 起始設定檔的路徑新增到 FFI\_PATH 中。

## 授權需求

確定 Net.Data 執行時所用的使用者 ID 有權存取 FFI 內建函數所使用的檔案。有關設定 Web 伺服器使用 Net.Data 檔案之存取權的詳細資料，請參閱 *Net.Data* 管理及程式設計指南中的架構章節部分。

## 純本文檔介面區隔字元

為了增進執行效能，您可以在純本文檔案中保留來自一系列 SQL 要求的 Net.Data 定位標籤輸出。您可以在後續的要求中來取回純本文檔案，以置換 SQL 要求的再次發出。

Net.Data 純本文檔案可以從 Net.Data 表格來建立，而 Net.Data 表格可以從純本文檔案來建置。為了製作表格和純本文檔案之間的轉換，您必須在表格和純本文檔案中的記錄中定義直欄之間的對映。區隔字元是一個旗號或區隔符號，FFI 會依據要求的轉換，使用它將檔案分成不同部份 (如一橫列中的直欄)。區隔字元提供如何將純本文檔案中的部分記錄分隔與對映到表格中的之欄的方式，以及如何可將在表格中的欄對映到純本文檔案中的記錄中。

有兩種區隔字元：

### 換行字元 (ASCII TEXT)

當您的表格只有一欄時，請使用這個轉換。Net.Data 會將每個在相對應純本文檔案中的記錄對映到在表格中的單一系列。在這個狀況下，將在純本文檔案中的記錄分開所使用的一般換行字元是唯一被使用的區隔字元。

### 換行字元與區隔字元字串 (DELIMITED)

當您的表格有多重欄時，請使用這個轉換。當 Net.Data 從表格中的列建立一個純本文檔案記錄時，它會將區隔字元字串以一個介於項目之間的區隔符號來置換。當 Net.Data 從純本文檔案來重新建立表格時，它會使用區隔字元字串來決

定在表格的欄中要用置放多少列。在這個範例中，一般性的換行字元會分隔在純本文檔中的記錄，該記錄將與表格中的列相對應，且區隔字元字串在一個單一記錄內會將項目分開。

對讀取作業而言，區隔字元會將檔案內容分成表格的橫列與直欄。對寫入作業而言，區隔字元指出表格橫列與直欄中值的結尾。Net.Data 會將區隔字元傳遞給 FFI，作為 Net.Data 巨集字串，且不會在字元尾端併入空值字元，除非在 DELIMITER 參數中明確列出空值字元。

若要在定界字元中使用空字元，請將 DELIMITER 參數指定為雙引號內的一條反斜線和一個零 (『\0』)，而不是使用兩個雙引號的空字串 (『""]』)。如果您指定 ASCIITEXT 轉換，Net.Data 將使用換行字元，作為區隔字元，且不處理任何要求的區隔字元。

如果您對寫入作業與讀取作業使用不同的區隔字元，檔案可能會發生不想要的變更。Net.Data 會以新的區隔字元寫入檔案。

區隔字元的最大長度為 256 個字元。

## 鎖定檔案

您可以使用 DTWF\_OPEN 與 DTWF\_CLOSE 函數，鎖定純本文檔。利用這些函數，Net.Data 可保留純本文檔，避免其它應用程式讀取或更新此檔案。

若要鎖定檔案，請使用 DTWF\_OPEN 函數。此函數確定其它應用程式無法使用檔案，如此可避免檔案在讀取與更新之間遭人變更。

若要釋放檔案，請使用 DTWF\_CLOSE 函數。此函數會釋放檔案，使其它應用程式可讀取或更新檔案。

## DTWF\_APPEND

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

將 Net.Data 表格的目次寫入文字檔尾端。

### 語法

@DTWF\_APPEND(filename, transform, delimiter, table, retry, rows)

@DTWF\_APPEND(filename, transform, delimiter, table, retry)

@DTWF\_APPEND(filename, transform, delimiter, table)

### 參數

表 167. DTWF\_APPEND 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	變數內容將新增至其中的檔案名稱。呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>transform</i>	IN	檔案的格式： <ul style="list-style-type: none"><li>• ASCIITEXT - 將表格寫入檔案，各直欄值之間放入換行字，並忽略 <i>delimiter</i> 參數。</li><li>• DELIMITED - 使用指定在 <i>delimiter</i> 參數中的區隔字元來將表格寫入檔案中。</li></ul> 檔案中的換行字元指出 ASCIITEXT 與 DELIMITED 轉換的 Net.Data 巨集表格的橫列結尾。
字串	<i>delimiter</i>	IN	一個指示值終止的字元字串。這個參數是區分大小寫的。忽略，如果 <i>transform</i> 是 ASCIITEXT。
表格	<i>table</i>	IN	一個從中讀取記錄的表格變數。  <b>對非 OS/400 使用者而言：</b> FFI 表格中橫列的最大長度為 16383 個字元。此限制包括 Net.Data 巨集表格中每一直欄的空值字元。
整數	<i>retry</i>	IN	如果檔案無法立刻附加的話，所要重試的次數。預設值是不重試。
整數	<i>rows</i>	IN	從 <i>table</i> 中所附加之最大的列數目。預設值是附加所有的列。指定 0 將附加所有的列。

### 回覆碼

表 168. DTWF\_APPEND 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。



表 168. DTWF\_APPEND 回覆碼 (繼續)

回覆碼	說明
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
2000	純本文檔介面內建函數找不到指定的檔案。
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2003	純本文檔介面內建函數無法將一系列資料讀入表格變數中，因為橫列的位元組數超出支援的最大位元組數。
2004	純本文檔介面內建函數試圖尋找檔案，但發現 FFI_PATH 架構檔變數中有一個路徑的長度超過支援的最大位元組數 4095。
2005	呼叫系統功能失敗。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 用法注意事項

檔案的現行內容會影響使用 DTWF\_APPEND 的結果，尤其是最後一橫列的最後一直欄的內容。如果換行字元後面跟著檔案的最後一橫列的最後一直欄值，附加的資料將置於新橫列中。否則，附加的資料將變成檔案的最後一橫列的一部份。

## 範例

### 範例 1：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
}%
@DTWF_APPEND(myFile, "DELIMITED", " ;", myTable)
```

### 範例 2：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
}%
@DTWF_APPEND(myFile, "ASCIITEXT", " ;", myTable)
```

### 範例 3：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
}%
@DTWF_APPEND(myFile, "ASCIITEXT", " ;", myTable, "0", "10")
```

## DTWF\_CLOSE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

關閉由 DTWF\_OPEN 所開啓的檔案。

### 語法

@DTWF\_CLOSE(filename, retry)

@DTWF\_CLOSE(filename)

### 參數

表 169. DTWF\_CLOSE 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	所要關閉之檔案的名稱。呼叫順利完成時，這個參數會傳回全部完整的檔名。
整數	<i>retry</i>	IN	如果檔案無法立刻關閉的話，所要重試的次數。預設值是不重試。

### 回覆碼

表 170. DTWF\_CLOSE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
2002	純本文檔介面內建函數無法關閉指定的檔案，因為這個巨集呼叫並未開啓此檔。

### 範例

#### 範例 1：

```
@DTWF_CLOSE(myFile, "5")
```

## DTWF\_DELETE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

從文字檔中刪除一些行。

### 語法

@DTWF\_DELETE(filename, transform, delimiter, retry, rows, startrow)

@DTWF\_DELETE(filename, transform, delimiter, retry, rows)

@DTWF\_DELETE(filename, transform, delimiter, retry)

@DTWF\_DELETE(filename, transform, delimiter)

### 參數

表 171. DTW\_DELETE 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	所要刪除之記錄的檔案名稱。呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>transform</i>	IN	檔案的格式： <ul style="list-style-type: none"><li>• ASCIITEXT - 將表格寫入檔案，各直欄值之間放入換行字，並忽略 <i>delimiter</i> 參數。</li><li>• DELIMITED - 使用指定在 <i>delimiter</i> 參數中的區隔字元來將表格寫入檔案中。</li></ul> 檔案中的換行字元指出 ASCIITEXT 與 DELIMITED 轉換的 Net.Data 巨集表格的橫列結尾。
字串	<i>delimiter</i>	IN	一個指示值終止的字元字串。這個參數是區分大小寫的。忽略，如果 <i>transform</i> 是 ASCIITEXT。
整數	<i>retry</i>	IN	如果記錄無法立刻刪除的話，所要重試的次數。預設值是不重試。
整數	<i>rows</i>	IN	所要刪除之最大的列數目。預設值是刪除所有的列。指定 0 將刪除所有的列。
整數	<i>startrow</i>	INOUT	開始刪除的列號碼。值 1 表示從第一列開始刪除。如果這個值大於在檔案中的列號碼，則該值會被變更為最後一個記錄並以錯誤傳回。預設值是從 1 開始。

### 回覆碼

表 172. DTWF\_DELETE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。

表 172. DTWF\_DELETE 回覆碼 (繼續)

回覆碼	說明
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
2000	純本文檔介面內建函數找不到指定的檔案。
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2003	純本文檔介面內建函數無法將一系列資料讀入表格變數中，因為橫列的位元組數超出支援的最大位元組數。
2004	純本文檔介面內建函數試圖尋找檔案，但發現 FFI_PATH 架構檔變數中有一個路徑的長度超過支援的最大位元組數 4095。
2005	呼叫系統功能失敗。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 範例

### 範例 1：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
    myWait = "5000"
    myRows = "2"
}%
@DTWF_DELETE(myFile, "Delimited", "|", myWait, myRows)
```

### 範例 2：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
    myStart = "1"
    myRows = "2"
}%
@DTWF_DELETE(myFile, "Asciitext", "|", "0", myRows, myStart)
```

## DTWF\_INSERT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

插入一些行到文字檔中。

### 語法

@DTWF\_INSERT(filename, transform, delimiter, table, retry, rows, startrow)

@DTWF\_INSERT(filename, transform, delimiter, table, retry, rows)

@DTWF\_INSERT(filename, transform, delimiter, table, retry)

@DTWF\_INSERT(filename, transform, delimiter, table)

### 參數

表 173. DTWF\_INSERT 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	插入記錄的檔案名稱。 呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>transform</i>	IN	檔案的格式： <ul style="list-style-type: none"><li>• ASCIITEXT - 將表格寫入檔案，各直欄值之間放入換行字，並忽略 <i>delimiter</i> 參數。</li><li>• DELIMITED - 使用指定在 <i>delimiter</i> 參數中的區隔字元來將表格寫入檔案中。</li></ul> 檔案中的換行字元指出 ASCIITEXT 與 DELIMITED 轉換的 Net.Data 巨集表格的橫列結尾。
字串	<i>delimiter</i>	IN	一個指示值終止的字元字串。這個參數是區分大小寫的。忽略，如果 <i>transform</i> 是 ASCIITEXT。
表格	<i>table</i>	IN	從該處將記錄插入表格的記錄變數。  <b>對非 OS/400 使用者而言：</b> FFI 表格中橫列的最大長度為 16383 個字元。此限制包括 Net.Data 巨集表格中每一直欄的空值字元。
整數	<i>retry</i>	IN	如果檔案無法立刻被寫入的話，所要重試的次數。預設值是不重試。
整數	<i>rows</i>	IN	從 <i>table</i> 所插入之最大的列數。預設值是插入所有的列。值 0 將會插入所有的列。
整數	<i>startrow</i>	INOUT	開始插入的行號。如果此值大於檔案的行數，則該值會被變更為最後一筆記錄並當作錯誤傳回。若指定 0，表示從檔案開頭後面插入。預設值是從 1 開始。

## 回覆碼

表 174. DTWF\_INSERT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
2000	純本文檔介面內建函數找不到指定的檔案。
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2003	純本文檔介面內建函數無法將一系列資料讀入表格變數中，因為橫列的位元組數超出支援的最大位元組數。
2004	純本文檔介面內建函數試圖尋找檔案，但發現 FFI_PATH 架構檔變數中有一個路徑的長度超過支援的最大位元組數 4095。
2005	呼叫系統功能失敗。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 範例

### 範例 1：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
    myWait = "3000"  
%}  
@DTWF_INSERT(myFile, "Delimited", "|", myTable, myWait)
```

### 範例 2：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
    myStart = "1"  
    myRows = "2"  
%}  
@DTWF_INSERT(myFile, "Asciitext", "|", myTable, "0", myRows, myStart)
```

## DTWF\_OPEN

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

開啓文字檔。

### 語法

@DTWF\_OPEN(filename, mode, retry)

@DTWF\_OPEN(filename, mode)

### 參數

表 175. DTWF\_OPEN 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	所要開啓的檔案名稱。呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>mode</i>	IN	所要求的存取類型： <ul style="list-style-type: none"><li>• r - 開啓舊有的檔案以供讀取。</li><li>• w - 建立一個檔案以供撰寫。（將舊有之相同的檔案名稱銷毀，如果存在的話。）</li><li>• a - 開啓一個檔案以供附加。如果找不到的話，Net.Data 會建立檔案。</li><li>• r+ - 開啓一個舊有的檔案以供讀取和寫入。</li><li>• w+ - 建立一個檔案以供讀取和寫入。（將舊有之相同的檔案名稱銷毀，如果存在的話。）</li><li>• a+ - 以附加模式開啓檔案以供讀取或附加。如果找不到的話，Net.Data 會建立檔案。</li></ul>
整數	<i>retry</i>	IN	如果檔案無法立刻開啓的話，所要重試的次數。預設值是不重試。

### 回覆碼

表 176. DTWF\_OPEN 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
2000	純本文檔介面內建函數找不到指定的檔案。

表 176. DTWF\_OPEN 回覆碼 (繼續)

回覆碼	說明
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 用法注意事項

1. 當檔案不存在時，應該指定檔名的絕對路徑，且將建立檔案的目錄須符合 FFI\_PATH 中指定的目錄。如果未使用絕對路徑，將開啓現行工作目錄中的檔案。
2. DTWF\_OPEN 會保持檔案開啓，否則在每一個純本文檔作業後，檔案會關閉。
3. 使用 DTWF\_OPEN 來減少檔案開啓的次數。若未使用 DTWF\_OPEN，則在每一次純本文檔作業之後，檔案會關閉。檔案會一直保持開啓，直到使用 DTWF\_CLOSE 關閉，或巨集處理程序終止。

## 範例

### 範例 1：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myMode = "r+"
}%
@DTWF_OPEN(myFile, myMode, "1000")
```



## DTWF\_READ

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

從文字檔讀取一些行到 Net.Data 表格中。

### 語法

@DTWF\_READ(filename, transform, delimiter, table, retry, rows, startrow, columns)

@DTWF\_READ(filename, transform, delimiter, table, retry, rows, startrow)

@DTWF\_READ(filename, transform, delimiter, table, retry, rows)

@DTWF\_READ(filename, transform, delimiter, table, retry)

@DTWF\_READ(filename, transform, delimiter, table)

### 參數

表 177. DTWF\_READ 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	將其記錄讀取至表格變數中的檔案名稱。 呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>transform</i>	IN	檔案的格式： <ul style="list-style-type: none"><li>• ASCIITEXT - 將表格寫入檔案，各直欄值之間放入換行字，並忽略 <i>delimiter</i> 參數。</li><li>• DELIMITED - 使用指定在 <i>delimiter</i> 參數中的區隔字元來將表格寫入檔案中。</li></ul> 檔案中的換行字元指出 ASCIITEXT 與 DELIMITED 轉換的 Net.Data 巨集表格的橫列結尾。
字串	<i>delimiter</i>	IN	一個指示值終止的字元字串。這個參數是區分大小寫的。忽略，如果 <i>transform</i> 是 ASCIITEXT。
表格	<i>table</i>	OUT	一個從中讀取檔案記錄的表格變數。  <b>對非 OS/400 使用者而言：</b> FFI 表格中橫列的最大長度為 16383 個字元。此限制包括 Net.Data 巨集表格中每一直欄的空值字元。
整數	<i>retry</i>	IN	如果檔案無法立刻讀取的話，所要重試的次數。預設值是不重試。
整數	<i>rows</i>	INOUT	所要讀取至表格中之檔案記錄的最大數目。預設值是讀取所有的記錄，或直到表格已滿。0 值表示一直讀取至檔案的尾端。返回在結果表格中的列數。
整數	<i>startrow</i>	IN	在檔案中，要開始使用的記錄。預設值是從第一個記錄開啓讀取。
整數	<i>columns</i>	OUT	傳回表格中的欄數。

## 回覆碼

表 178. DTWF\_READ 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
1010	資料會被寫入表格中，直到表格沒有空位為止，剩餘資料則捨棄。
2000	純本文檔介面內建函數找不到指定的檔案。
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2003	純本文檔介面內建函數無法將一系列資料讀入表格變數中，因為橫列的位元組數超出支援的最大位元組數。
2004	純本文檔介面內建函數試圖尋找檔案，但發現 FFI_PATH 架構檔變數中有一個路徑的長度超過支援的最大位元組數 4095。
2005	呼叫系統功能失敗。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 範例

### 範例 1：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
    myWait = "1000"  
}%  
@DTWF_READ(myFile, "DELIMITED", ";", myTable, myWait)
```

### 範例 2：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
    myWait = "0"  
    myRows = "0"  
    myStartrow = "1"  
    myColumns = ""  
}%  
@DTWF_READ(myFile, "DELIMITED", ";", myTable, myWait, myRows,  
            myStartrow, myColumns)
```

### 範例 3：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
}%  
@DTWF_READ(myFile, "ASCIITEXT", ";", myTable)  
@DTW_TB_TABLE(myTable, "BORDER", "")
```

## DTWF\_REMOVE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

刪除整個檔案。

### 語法

@DTWF\_REMOVE(filename, retry)

@DTWF\_REMOVE(filename)

### 參數

表 179. DTW\_REMOVE 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	所要刪除的檔案名稱。呼叫順利完成時，這個參數會傳回全部完整的檔名。
整數	<i>retry</i>	IN	如果檔案無法立刻刪除的話，所要重試的次數。預設值是不重試。

### 回覆碼

表 180. DTWF\_REMOVE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
2000	純本文檔介面內建函數找不到指定的檔案。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

### 範例

#### 範例 1：

```
%DEFINE myFile = "c:/private/myfile"  
@DTWF_REMOVE(myFile)
```

#### 範例 2：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myWait = "2000"  
}%  
@DTWF_REMOVE(myFile, myWait)
```

## DTWF\_SEARCH

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

搜尋文字檔的字串，將結果傳回 Net.Data 表格。

### 語法

@DTWF\_SEARCH(filename, transform, delimiter, table, searchFor, retry, rows, startrow)

@DTWF\_SEARCH(filename, transform, delimiter, table, searchFor, retry, rows)

@DTWF\_SEARCH(filename, transform, delimiter, table, searchFor, retry)

@DTWF\_SEARCH(filename, transform, delimiter, table, searchFor)

### 參數

表 181. DTWF\_SEARCH 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	所要搜尋之檔案的名稱。呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>transform</i>	IN	檔案的格式： <ul style="list-style-type: none"><li>• ASCIITEXT - 將表格寫入檔案，各直欄值之間放入換行字，並忽略 <i>delimiter</i> 參數。</li><li>• DELIMITED - 使用指定在 <i>delimiter</i> 參數中的區隔字元來將表格寫入檔案中。</li></ul> 檔案中的換行字元指出 ASCIITEXT 與 DELIMITED 轉換的 Net.Data 巨集表格的橫列結尾。
字串	<i>delimiter</i>	IN	一個指示值終止的字元字串。這個參數是區分大小寫的。忽略，如果 <i>transform</i> 是 ASCIITEXT。
表格	<i>table</i>	OUT	一個置放結果的表格變數。如果轉換是 DELIMITED，會傳回三個直欄： <ul style="list-style-type: none"><li>• 找到有相符者的列。</li><li>• 找到有相符者的欄。</li><li>• 來自檔案的相符欄。</li></ul> 對非 OS/400 使用者而言：FFI 表格中橫列的最大長度為 16383 個字元。此限制包括 Net.Data 巨集表格中每一直欄的空值字元。
字串	<i>searchFor</i>	IN	所要搜尋的字元字串。
整數	<i>retry</i>	IN	如果檔案無法立刻搜尋的話，所要重試的次數。預設值是不重試。
整數	<i>rows</i>	INOUT	要讀取入 <i>table</i> 中之最大的列數。預設值是讀取所有的列，或直到 <i>table</i> 已滿。指定 0 來讀取置檔案的尾端。在結果表格中的列數是由這個參數傳回的。

表 181. DTWF\_SEARCH 參數 (繼續)

資料類型	參數	使用	說明
整數	<i>startrow</i>	IN	在檔案中開始進行搜尋的記錄。預設值是 1，其將從第一個記錄開始搜尋。

## 回覆碼

表 182. DTWF\_SEARCH 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
1010	資料會被寫入表格中，直到表格沒有空位為止，剩餘資料則捨棄。
2000	純本文檔介面內建函數找不到指定的檔案。
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2003	純本文檔介面內建函數無法將一系列資料讀入表格變數中，因為橫列的位元組數超出支援的最大位元組數。
2004	純本文檔介面內建函數試圖尋找檔案，但發現 FFI_PATH 架構檔變數中有一個路徑的長度超過支援的最大位元組數 4095。
2005	呼叫系統功能失敗。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 用法注意事項

1. 因 DTWF\_SEARCH 傳回的表格具有三欄。前兩欄含有找到相符字元的橫列與直欄號碼；最後一直欄含有的直欄值具有 *SearchFor* 參數中所指定的字元。例如，如果檔案的第 4 列含有第 3 欄的相符字元，則傳回的表格將具有在第一欄具有號碼 4 一列，來指出檔案的橫列的出處；它在第 2 欄具有號碼 3，指出檔案的哪一欄具有相符字元；它在第 3 欄中具有完整的直欄值
2. *SearchFor* 參數無法包括 *delimiter* 參數的內容

## 範例

### 範例 1：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
    myWait = "1000"
    mySearch = "0123456789abcdef"
@DTWF_SEARCH(myFile, "DELIMITED", ";",
    myTable, mySearch, myWait)
```

## 範例 2：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
    mySearch = "answer:"  
    myWait = "0"  
    myRows = "0"  
    myStartrow = "1"  
%}  
@DTWF_SEARCH(myFile, "DELIMITED", ";", myTable,  
             mySearch, myWait, myRows, myStartrow)
```

## DTWF\_UPDATE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

以 Net.Data 表格中的資料更新文字檔中某些行。

### 語法

@DTWF\_UPDATE(filename, transform, delimiter, table, retry, rows, startrow)

@DTWF\_UPDATE(filename, transform, delimiter, table, retry, rows)

@DTWF\_UPDATE(filename, transform, delimiter, table, retry)

@DTWF\_UPDATE(filename, transform, delimiter, table)

### 參數

表 183. DTWF\_UPDATE 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	其記錄是從表格變數來更新的檔案名稱。 呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>transform</i>	IN	檔案的格式： <ul style="list-style-type: none"><li>• ASCII TEXT - 將表格寫入檔案，各直欄值之間放入換行字，並忽略 <i>delimiter</i> 參數。</li><li>• DELIMITED - 使用指定在 <i>delimiter</i> 參數中的區隔字元來將表格寫入檔案中。</li></ul> 檔案中的換行字元指出 ASCII TEXT 與 DELIMITED 轉換的 Net.Data 巨集表格的橫列結尾。
字串	<i>delimiter</i>	IN	一個指示值終止的字元字串。這個參數是區分大小寫的。忽略，如果 <i>transform</i> 是 ASCII TEXT。
表格	<i>table</i>	IN	檔案記錄從該處開始更新的表格變數。  <b>對非 OS/400 使用者而言：</b> FFI 表格中橫列的最大長度為 16383 個字元。此限制包括 Net.Data 巨集表格中每一直欄的空值字元。
整數	<i>retry</i>	IN	如果檔案無法立刻被寫入的話，所要重試的次數。預設值是不重試。
整數	<i>rows</i>	IN	從 <i>table</i> 所要更新之記錄的最大數目。預設值是更新所有的記錄。0 值表示更新所有在檔案中的列。
整數	<i>startrow</i>	INOUT	要更新的第一個檔案。預設值是 1，其表示在檔案的開始即進行更新。如果值大於檔案中記錄的數目，則該值會被變更為指示在檔案中最後一個記錄的號碼，並傳回錯誤。



## 回覆碼

表 184. DTWF\_UPDATE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
2000	純本文檔介面內建函數找不到指定的檔案。
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2003	純本文檔介面內建函數無法將一系列資料讀入表格變數中，因為橫列的位元組數超出支援的最大位元組數。
2004	純本文檔介面內建函數試圖尋找檔案，但發現 FFI_PATH 架構檔變數中有一個路徑的長度超過支援的最大位元組數 4095。
2005	呼叫系統功能失敗。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 用法注意事項

如果檔案不存在時，應該指定檔名的絕對路徑，且將建立檔案的目錄須符合 FFI\_PATH 中指定的目錄。如果未使用絕對路徑，將開啓現行工作目錄中的檔案。

## 範例

### 範例 1：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
    myWait = "1500"  
    myRows = "2"  
%}  
@DTWF_UPDATE(myFile, "Delimited", "|", myTable, myWait, myRows)
```

### 範例 2：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
    myStart = "1"  
    myRows = "2"  
%}  
@DTWF_UPDATE(myFile, "Asciitext", "|", myTable, "0", myRows, myStart)
```

## DTWF\_WRITE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X	X	X		X	X

### 目的

將 Net.Data 表格目次寫入文字檔中。

### 語法

@DTWF\_WRITE(filename, transform, delimiter, table, retry, rows, startrow)

@DTWF\_WRITE(filename, transform, delimiter, table, retry, rows)

@DTWF\_WRITE(filename, transform, delimiter, table, retry)

@DTWF\_WRITE(filename, transform, delimiter, table)

### 參數

表 185. DTWF\_WRITE 參數

資料類型	參數	使用	說明
字串	<i>filename</i>	INOUT	一個寫入表格變數之記錄的檔案名稱。呼叫順利完成時，這個參數會傳回全部完整的檔名。
字串	<i>transform</i>	IN	檔案的格式： <ul style="list-style-type: none"><li>• ASCIITEXT - 將表格寫入檔案，各直欄值之間放入換行字，並忽略 <i>delimiter</i> 參數。</li><li>• DELIMITED - 使用指定在 <i>delimiter</i> 參數中的區隔字元來將表格寫入檔案中。</li></ul> 檔案中的換行字元指出 ASCIITEXT 與 DELIMITED 轉換的 Net.Data 巨集表格的橫列結尾。
字串	<i>delimiter</i>	IN	一個指示值終止的字元字串。這個參數是區分大小寫的。忽略，如果 <i>transform</i> 是 ASCIITEXT。
表格	<i>table</i>	IN	使用來將列匯出至檔案的表格變數。  <b>對非 OS/400 使用者而言：</b> FFI 表格中橫列的最大長度為 16383 個字元。此限制包括 Net.Data 巨集表格中每一直欄的空值字元。
整數	<i>retry</i>	IN	如果檔案無法立刻被寫入的話，所要重試的次數。預設值是不重試。
整數	<i>rows</i>	IN	所要撰寫之檔案記錄的最大數目。預設值是寫入整個表格。0 值表示撰寫所有至檔案尾端的記錄。
整數	<i>startrow</i>	INOUT	開始撰寫至檔案中的記錄號碼。預設值是 1，其表示在第一個記錄開始進行。如果一個值位於所指定的檔案之後，檔案的最後一列將會與錯誤一起傳回。

## 回覆碼

表 186. DTWF\_WRITE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
2000	純本文檔介面內建函數找不到指定的檔案。
2001	純本文檔介面內建函數無法開啓指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。
2003	純本文檔介面內建函數無法將一系列資料讀入表格變數中，因為橫列的位元組數超出支援的最大位元組數。
2004	純本文檔介面內建函數試圖尋找檔案，但發現 FFI_PATH 架構檔變數中有一個路徑的長度超過支援的最大位元組數 4095。
2005	呼叫系統功能失敗。
2006	純本文檔介面內建函數無法存取指定的檔案，因為檔案正被此處理或另一個處理使用，無法在指定模式之下共用。

## 用法注意事項

如果檔案不存在時，應該指定檔名的絕對路徑，且將建立檔案的目錄須符合 FFI\_PATH 中指定的目錄。如果未使用絕對路徑，將開啓現行工作目錄中的檔案。

## 範例

### 範例 1：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
%}  
@DTWF_WRITE(myFile, "DELIMITED", ";", myTable)
```

### 範例 2：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
%}  
@DTWF_WRITE(myFile, "ASCIITEXT", ";", myTable, "5000")
```

### 範例 3：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myTable = %TABLE  
%}  
@DTWF_WRITE(myFile, "ASCIITEXT", ";", myTable, "5000", "10", "50")
```

---

## Web 登記函數

Web 登記是一個讓您能輕易地新增、取回和刪除登錄而由 Net.Data 來維護鍵的檔案，您可以在一個單一系統上建立多重 Net.Data Web 登記。每個登記都具有一個名稱，並且可以包含多重登錄。Net.Data 提供函數來維護登記和其所包含之登錄。

- 第263頁的『DTWR\_ADDENTRY』
- 第265頁的『DTWR\_CLEARREG』
- 第266頁的『DTWR\_CLOSEREG』
- 第267頁的『DTWR\_CREATEREG』
- 第269頁的『DTWR\_DELENTY』
- 第270頁的『DTWR\_DELREG』
- 第271頁的『DTWR\_LISTREG』
- 第273頁的『DTWR\_LISTSUB』
- 第275頁的『DTWR\_OPENREG』
- 第276頁的『DTWR\_RTVENTRY』
- 第278頁的『DTWR\_UPDATEENTRY』

### 限制：

- 當使用 OS/2 時，不要對 *registry*、*registryVariable* 與 *registryData* 參數使用星號 (\*)。
- 傳送到「Web 登記」函數的每一個參數不能超過 2048 個字元。

## DTWR\_ADDENTRY

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

將登錄新增至 Web 登記。

### 語法

@DTWR\_ADDENTRY(registry, registryVariable, registryData, index)

@DTWR\_ADDENTRY(registry, registryVariable, registryData)

### 參數

表 187. DTWR\_ADDENTRY 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	一個要新增登錄的登記名稱。
字串	<i>registryVariable</i>	IN	所要新增之登記 <i>registryVariable</i> 字串部分的值。
字串	<i>registryData</i>	IN	所要新增之登記 <i>registryData</i> 字串部分的值。
字串	<i>index</i>	IN	在所要新增之索引登錄中， <i>registryVariable</i> 字串索引部分的值。這個參數是可選用的。如果有指定的話，將會把一個索引的登錄新增至所指定的登記中。

### 回覆碼

表 188. DTWR\_ADDENTRY 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3003	Web 登記內建函數無法在指定的登記中新增登錄，因為指定的登錄已存在。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3006	Web 登記內建函數無法建立指定的登記，因為登記名稱中的路徑不存在。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 範例

範例 1：

```
@DTWR_ADDENTRY("Myregistry", "Jones", "http://Advantis.com/~Jones/webproj")
```

**範例 2 :**

```
@DTWR_ADDENTRY("URLLIST", "SMITH", "http://www.software.ibm.com/",  
    "WORK_URL,")
```

## DTWR\_CLEARREG

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

從 Web 登記清除登錄。

### 語法

@DTWR\_CLEARREG(registry)

### 參數

表 189. DTWR\_CLEARREG 參數

資料類型	參數	使用	說明
字串	registry	IN	所要清除的登記名稱。

### 回覆碼

表 190. DTWR\_CLEARREG 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3006	Web 登記內建函數無法建立指定的登記，因為登記名稱中的路徑不存在。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 範例

#### 範例 1：

```
@DTWR_CLEARREG("Myregistry")
```

## DTWR\_CLOSEREG

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

關閉 Web 登記

### 語法

@DTWR\_CLOSEREG(registry)

### 參數

表 191. DTWR\_CLOSEREG 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	要關閉的登記名稱。  <b>限制：</b> 在 Web 登記名稱中，不要使用特殊字元，例如星號 (*) 與反斜線 (\)。

### 回覆碼

表 192. DTWR\_CLOSEREG 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。

### 範例

範例 1：關閉登記

```
@DTWR_CLOSEREG("/qsys.lib/mylib.lib/myreg.file")
```



## DTWR\_CREATEREG

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

建立一個新的 Web 登記。

### 語法

@DTWR\_CREATEREG(registry, security)

@DTWR\_CREATEREG(registry)

### 參數

表 193. DTWR\_CREATEREG 參數

資料類型	參數	使用	說明
字串	registry	IN	所要建立的登記名稱。  <b>限制：</b> 在 Web 登記名稱中，不要使用特殊字元，例如星號 (*) 與反斜線 (\)。
字串	security	IN	要建立 registry 所要有的保密類型。在 UNIX 作業系統上，預設安全與建立登記的目錄是相同的。為 3 個安全群組指定安全：使用者、群組、和公用。R 提供讀取許可權、W 提供寫入許可權、而 X 提供執行許可權。例如，若要提供所有 3 個群組完全的權限，請將這個參數指定為 *RWX，*RWX，*RWX。這個參數是可選用的。

### 回覆碼

表 194. DTWR\_CREATEREG 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3002	Web 登記內建函數無法刪除指定的登記。
3006	Web 登記內建函數無法建立指定的登記，因為登記名稱中的路徑不存在。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。
3008	Web 登記內建函數無法建立指定的登記，原因不明。

### 範例

範例 1：

```
@DTWR_CREATEREG("myRegistry")
```

**範例 2 :**

```
@DTWR_CREATEREG("URLLIST", "*RWX, *RWX, *R")
```

## DTWR\_DELENTY

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

從 Web 登記中刪除登錄。

### 語法

@DTWR\_DELENTY(registry, registryVariable, index)

@DTWR\_DELENTY(registry, registryVariable)

### 參數

表 195. DTWR\_DELENTY 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	從中除去登錄的登記名稱。
字串	<i>registryVariable</i>	IN	所要除去之登錄的 <i>registryVariable</i> 字串部分的值。
字串	<i>index</i>	IN	在索引的登錄中， <i>registryVariable</i> 字串之索引部分的值。這是一個可選用的參數。如果指定的話，從登記中將索引的登錄除去。

### 回覆碼

表 196. DTWF\_DELENTY 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3004	Web 登記內建函數無法在指定的登記中除去或取回登錄，因為指定的登錄不存在。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 範例

#### 範例 1：

```
@DTWR_DELENTY("Myregistry", "Jones")
```

#### 範例 2：

```
@DTWR_DELENTY("URLLIST", "SMITH", "WORK_URL")
```

## DTWR\_DELREG

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

刪除 Web 登記

### 語法

@DTWR\_DELREG(registry)

### 參數

表 197. DTWR\_DELREG 參數

資料類型	參數	使用	說明
字串	registry	IN	要刪除登記的名稱。

### 回覆碼

表 198. DTWR\_DELREG 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 範例

範例 1：

```
@DTWR_DELREG("Myregistry")
```

## DTWR\_LISTREG

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

列示 Web 登記的目次。

### 語法

@DTWR\_LISTREG(registry, registryTable)

### 參數

表 199. DTWR\_LISTREG 參數

資料類型	參數	使用	說明
字串	registry	IN	所要列示的登記名稱。
表格	registryTable	OUT	放置登記之表格變數的名稱。

### 回覆碼

表 200. DTWR\_LISTREG 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1004	函數呼叫上傳遞的參數必須是 Net.Data 巨集表變數，但卻是不同的變數類型。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 用法注意事項

DTWR\_LISTREG 傳回由使用者所傳送之 OUT 表格變數中的有關登記的資訊。表格變數在被當作參數般傳送至 FUNCTION 區塊以供 LISTREG 登記作業使用之前，已在使用者巨集中定義過了。

如果使用者為表格的最大列數而使用 ALL 選項來定義表格變數，則這個作業會列示所有在表格中可用的登記，每個表格列都有一個。換句話說，如果使用者將表格列的最大數目指定為值 X，然後若在所指定的登記中的登錄多於 X 個登錄，則將只有前 X 個登錄會被列示出來且將會傳回一個錯誤碼，以表示因為可用來列示其它登錄的表格列不足，所以只可完成部分的列示。如果值 X 超過在指定之登記中可用登錄的數目的話，則將會列示出所有的登記登錄。

在表格中一定是 2 直欄。表格的「直欄」表頭設定為 `REGISTRY_VARIABLE` 和 `REGISTRY_DATA`。

## 範例

範例 1：

```
%DEFINE RegistryTable = %TABLE(ALL)

@DTWR_LISTREG("URLLIST", RegistryTable)
```

## DTWR\_LISTSUB

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
								X

### 目的

列示在 Web 登記中的立即子鍵。

### 語法

@DTWR\_LISTSUB(registry, registryTable)

### 參數

表 201. DTWR\_LISTSUB 參數

資料類型	參數	使用	說明
字串	registry	IN	所要列示的登記名稱。
表格	registryTable	OUT	放置登記之表格變數的名稱。

### 回覆碼

表 202. DTWR\_LISTSUB 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 用法注意事項

1. DTWR\_LISTSUB 傳回有關由使用者傳回在 OUT 表格參數中的登記。在表格變數被當作參數傳送至 LISTSUB 登記作業之前，將會在巨集中對表格變數下定義。

如果使用者為表格的最大列數而使用 ALL 選項來定義表格變數，則這個作業會列示在表格中所有可用的登記，每個表格列都有一個。換句話說，如果使用者將表格列的最大數目指定為 X，然後若在所指定的登記中的登錄多於 X 個登錄，則將只有前 X 個登錄會被列示出來且將會傳回一個錯誤碼，以表示因為可用來列示其它登錄的表格列不足，所以只可完成部分的列示。如果值 X 超過在指定之登記中可用登錄的數目的話，則將會列示出所有的登記登錄。在表格中的直欄數一定是一。

表格的直欄表頭將被設定為 "REGISTRY\_SUBKEY"。

2. 這個函數只有在「Windows95 系統登記」相容的作業系統上才有效。

## 範例

### 範例 1：

```
%DEFINE RegistryTable = %TABLE(ALL)

@DTWR_LISTSUB("URLLIST", RegistryTable)
```



## DTWR\_OPENREG

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

開啓 Web 登記。

### 語法

@DTWR\_OPENREG(registry, commit)

@DTWR\_OPENREG(registry)

### 參數

表 203. DTWR\_OPENREG 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	要開啓的登記名稱。
字串	<i>commit</i>	IN	單一符號或文字字串，用來指定是否要在確定控制下開啓登記。可能值如下： <b>Y</b> 在確定控制下開啓登記。 <b>N</b> 不在確定控制下開啓登記。 預設值爲 N

### 回覆碼

表 204. DTWR\_OPENREG 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 範例

範例 1：在確定控制下開啓登記

```
@DTWR_OPENREG("/qsys.lib/mylib.lib/myreg.file", "Y")
```

## DTWR\_RTENTRY

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

從 Web 登記取回登記字串值。

### 語法

@DTWR\_RTENTRY(registry, registryVariable, registryData, index)

@DTWR\_RTENTRY(registry, registryVariable, registryData)

@DTWR\_rRTENTRY(registry, registryVariable, index)

@DTWR\_rRTENTRY(registry, registryVariable)

### 參數

表 205. DTWR\_RTENTRY 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	具有所要取回登錄之登記的名稱。
字串	<i>registryVariable</i>	IN	其 <i>registryData</i> 字串被取回之登記登錄之 <i>registryVariable</i> 字串部分的值。
字串	<i>registryData</i>	OUT	傳回符合 <i>registryVariable</i> 之登記之 <i>registryData</i> 字串部分的值。
字串	<i>index</i>	IN	其被傳回之 <i>registryData</i> 字串之索引登錄中的 <i>registryVariable</i> 字串索引部分的值。這是一個可選用的參數。如果有指定的話，將會傳回索引之登錄的 <i>registryData</i> 字串。

### 回覆碼

表 206. DTWR\_RTENTRY 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。
1007	參數包含的值無效。
3004	Web 登記內建函數無法在指定的登記中除去或取回登錄，因為指定的登錄不存在。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

## 範例

### 範例 1：

```
%DEFINE RegistryData = ""  
@DTWR_RTENTRY("Myregistry", "Jones", RegistryData)
```

### 範例 2：

```
@DTWR_RTENTRY("URLLIST", "SMITH", RegistryData, "WORK_URL")
```

### 範例 3：

```
@DTWR_rRTENTRY("Myregistry", "Jones")
```

### 範例 4：

```
@DTWR_rRTENTRY("URLLIST", "SMITH", "WORK_URL")
```

## DTWR\_UPDATEENTRY

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X			X		X			X

### 目的

更新 Web 登記中的登記字串值。

### 語法

@DTWR\_UPDATEENTRY(registry, registryVariable, newData, index)

@DTWR\_UPDATEENTRY(registry, registryVariable, newData)

### 參數

表 207. DTWR\_UPDATEENTRY 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	具有所要更新之登錄的登記名稱。
字串	<i>registryVariable</i>	IN	所要更新之登記的 <i>registryVariable</i> 字串部分的值。
字串	<i>newData</i>	IN	所要更新之登記的 <i>registryData</i> 字串部分的新值。
字串	<i>index</i>	IN	在所要更新之索引登錄中， <i>registryVariable</i> 字串索引部分的值。這是一個可選用的參數。如果指定的話，將會更新索引登錄。

### 回覆碼

表 208. DTWR\_UPDATEENTRY 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1002	輸入參數包含的字串有空字元結尾。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
3003	Web 登記內建函數無法在指定的登記中新增登錄，因為指定的登錄已存在。
3004	Web 登記內建函數無法在指定的登記中除去或取回登錄，因為指定的登錄不存在。
3005	Web 登記內建函數無法使用指定的登記，因為找不到此登記。
3007	Web 登記內建函數無法完成指定的作業，因為要求者對於指定的登記並無適當的權限。

### 用法注意事項

不能變更對應到值的登錄項目名稱。

## 範例

### 範例 1：

```
@DTWR_UPDATEENTRY("Myregistry", "Jones", "http://advantis.com/~Jones/personal")
```

### 範例 2：

```
@DTWR_UPDATEENTRY("URLLIST", "SMITH", "http://www.software.ibm.com/personal", "WORK_URL")
```

---

## 持續巨集函數

持續巨集函數可經由協助您定義單一異動內哪些巨集區塊是持續的，來支援 Net.Data 中的異動處理。使用這些函數，可定義異動的開頭與結尾，哪些 HTML 區塊在整個異動中是持續的、異動內的變數範圍，以及是否要確定或回捲異動內的變更。

- 第281頁的『DTW\_ACCEPT』
- 第283頁的『DTW\_COMMIT』
- 第284頁的『DTW\_ROLLBACK』
- 第285頁的『DTW\_RTVHANDLE』
- 第286頁的『DTW\_STATIC』
- 第287頁的『DTW\_TERMINATE』

## DTW\_ACCEPT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

定義用來呼叫持續巨集的異動 `handle`。

### 語法

@DTW\_ACCEPT(handle, timeout)

@DTW\_ACCEPT(handle)

### 參數

表 209. DTW\_ACCEPT 參數

資料類型	參數	使用	說明
字串	<i>handle</i>	IN	一個變數或文字字串，指定將在 URL 中使用，以便在此持續異動中進行後續巨集呼叫的異動 <code>handle</code> 。
整數	<i>timeout</i>	IN	一個變數或文字字串，指定伺服此埠的工作等待回應的時間量 (秒)。這個值會置換 DTW_STATIC() 函數上所指定的任何逾時值。

### 回覆碼

表 210. DTW\_ACCEPT 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
8200	未啟用巨集持續。
8201	未按順序呼叫持續的內建函數。

### 用法注意事項

1. Net.Data 需要異動 `handle` 包括在 URL 中，如此 URL 方可呼叫巨集，為來自 Web 瀏覽器的回應。當一個要求來到 Web 伺服器時，伺服器會使用異動 `handle`，將要求遞送到正在處理異動的 CGI 處理。

須在巨集中的每一 HTML 區塊的開頭呼叫異動 `handle`，直到最後一個含有 DTW\_TERMINATE() 的呼叫的邏輯區塊為止。在任何文字輸出到瀏覽器之前，若找不到 DTW\_ACCEPT() 或 DTW\_TERMINATE() 的呼叫，將發生 Net.Data 錯誤。

2. 您可以指定此頁的逾時值，來置換 @DTW\_STATIC() 函數上所指定的逾時值。Web 伺服器會等待指定的時間量 (秒)，讓使用者回應此要求。
3. 當巨集不在持續狀態時，若呼叫這個函數，將發生 Net.Data 錯誤。

4. 含有異動 handle 的 URL 可撰寫為套表按鈕上的動作，或撰寫為瀏覽器頁面上的超本文鏈結。

## 範例

### 範例 1：

```
%DEFINE handle = ""
@DTW_RTVHANLDE(handle)

%HTML(REPORT){
@DTW_ACCEPT(handle)
...
%}
```



## DTW\_COMMIT

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

使自從上一個確定界限後，在確定控制下對資源所做的任何擱置變更成為永久變更，並建立新的確定界限。

### 語法

@DTW\_COMMIT()

### 參數

無。

### 回覆碼

表 211. DTW\_COMMIT 回覆碼

回覆碼	說明
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。

### 範例

範例 1：指定確定

```
@DTW_COMMIT()  
%HTML(report){  
%}
```

## DTW\_ROLLBACK

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

重新建立上一個確定界限作為現行確定界限。自從退出上一個確定界限後，在 `Net.Data` 執行的處理的確定控制下，對資源所做的所有變更。

### 語法

```
@DTW_ROLLBACK()
```

### 參數

無。

### 回覆碼

表 212. `DTW_ROLLBACK` 回覆碼

回覆碼	說明
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。

### 範例

**範例 1：**指定回捲

```
@DTW_ROLLBACK()  
%HTML(report){  
%}
```

## DTW\_RTVHANDLE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

產生並傳回一個異動 `handle`，它對於跨越個別呼叫的這個巨集而言是唯一的，且會依據緒資訊、時間戳記及現行使用者的組合來計算。

### 語法

@DTW\_RTVHANDLE(handle)

### 參數

表 213. DTW\_RTVHANDLE 參數

資料類型	參數	使用	說明
字串	<i>handle</i>	OUT	含有現行持續巨集的唯一異動 <code>handle</code> 的變數。

### 回覆碼

表 214. DTW\_RTVHANDLE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 <code>Net.Data</code> 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1006	文字字串傳遞給函數呼叫中的參數，但此參數必須是輸出參數。

### 用法注意事項

異動 `handle` 可用來確定指定為持續異動一部份的 URL 對 HTTP 伺服器是唯一的，且可穩穩地識別為有效的要求。

### 範例

**範例 1：**定義用來取回異動 `handle` 的 `handle`變數

```
%DEFINE handle = ""
@DTW_RTVHANDLE(handle)
```

## DTW\_STATIC

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

指出整個巨集是持續的。

### 語法

@DTW\_STATIC(timeout)

@DTW\_STATIC()

### 參數

表 215. DTW\_STATIC 參數

資料類型	參數	使用	說明
整數	<i>timeout</i>	IN	一個變數或文字字串，指定處理此異動的處理應等待回應的時間量 (秒)。

### 回覆碼

表 216. DTW\_STATIC 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1001	輸入參數包含 NULL 值。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
1005	函數呼叫上傳遞的參數必須是字串變數，但卻是不同的變數類型。
1007	參數包含的值無效。
8202	無法啟用持續。

### 用法注意事項

1. DTW\_STATIC 應該是巨集中第一個陳述式。在這個函數呼叫後於巨集中定義的所有變數，在多個巨集呼叫之間均是持續的，除非另有指定，以及直到呼叫 DTW\_TERMINATE() 或終止處理為止。
2. 可在函數呼叫上設定逾時值 (秒)，來指出 Net.Data 執行的處理等待瀏覽器回應的時間量。如果逾時值到期，則處理將終止，且會回捲自從上一個確定界限後，於確定控制下對資源所做的全部變更。
3. 如果在後續 @DTW\_ACCEPT() 呼叫上指定了逾時值，Net.Data 將以後續呼叫中的值置換這個值。如果未在這個呼叫或後續 @DTW\_ACCEPT() 呼叫上指定逾時值，將使用 Web 伺服器的預設逾時值。

### 範例

範例 1：指定逾時值 60 秒的 DTW\_STATIC() 的呼叫

```
@DTW_STATIC("60")
```

## DTW\_TERMINATE

AIX	HP-UX	Linux	OS/2	OS/390	OS/400	SCO	SUN	Win NT
					X			

### 目的

終止持續異動。自從上一個確定界限後，於確定控制下對資源所做的變更均成為永久變更。

### 語法

@DTW\_TERMINATE()

### 參數

無

### 回覆碼

表 217. DTW\_TERMINATE 回覆碼

回覆碼	說明
-1001	伺服器無法處理 Net.Data 的配置記憶體要求。
1003	函數呼叫中傳遞的參數超過容許數目，或少於函數要求的最小數目。
8200	未啟用巨集持續。
8201	未按順序呼叫持續的內建函數。

### 用法注意事項

1. 在任何文字輸出到瀏覽器之前，於持續異動的上一個邏輯 HTML 區塊的開頭呼叫 DTW\_TERMINATE 函數。如果有任何文字出現在區塊內的函數之前，將發生 Net.Data 錯誤。請注意，可能有多個上一個邏輯 HTML 區塊，得視應用程式的撰寫方式而定。
2. 當巨集不在持續狀態時，若呼叫這個函數，將發生 Net.Data 錯誤。

### 範例

範例：終止持續異動

```
%HTML(QUIT){  
@DTW_TERMINATE()  
...  
%}
```



## 附錄A. Net.Data 技術圖書庫

Net.Data Technical Library 位於 Net.Data Web 網站，網址如下：

<http://www.software.ibm.com/data/net.data/library.html>

文件	說明
<ul style="list-style-type: none"><li>• <i>Net.Data Administration and Programming Guide for OS/390</i></li><li>• <i>OS/2 版、Windows NT 版和 UNIX 版 Net.Data 管理及程式設計指南</i></li><li>• <i>OS/400 版 Net.Data 管理及程式設計指南</i></li></ul>	包含關於安裝、架構和呼叫 Net.Data 的概念和作業資訊。也說明如何撰寫 Net.Data 巨集，使用 Net.Data 效能技術，使用 Net.Data 語言環境，管理連線以及使用 Net.Data 記錄和追蹤故障檢修及效能調整。
<i>Net.Data 參考手冊</i>	說明 Net.Data 巨集語言、變數和內建函數。
<i>Net.Data 語言環境介面參考手冊</i>	說明 Net.Data 語言環境介面。
<i>Net.Data 訊息與訊息碼參考手冊</i>	列出 Net.Data 錯誤訊息和回覆碼。





---

## 附錄B. DB2 WWW 連線

如果您擁有 DB2 WWW 連線，則您可以使用 `Net.Data` 來執行現有的應用程式。為充分利用 `Net.Data` 版本 2 的特性，建議更新您的應用程式。

DB2 WWW 語言結構為：

- 『EXEC\_SQL』
- 『HTML\_INPUT』
- 『HTML\_REPORT』
- 『SQL』
- 第292頁的『SQL\_MESSAGE』
- 第292頁的『SQL\_REPORT』
- 第292頁的『SQL\_CODE』

---

### EXEC\_SQL

這個語言結構呼叫一個 SQL 區塊。建議您將 SQL 陳述式以函數的方式來呼叫。請參閱第16頁的『FUNCTION 區塊』以取得有關詳細資訊。

---

### HTML\_INPUT

這個語言結構與稱為 INPUT 的 HTML 區塊相同。請參閱第26頁的『HTML 區塊』以取得有關詳細資訊。

---

### HTML\_REPORT

這個語言結構與稱為 REPORT 的 HTML 區塊相同。請參閱第26頁的『HTML 區塊』以取得有關詳細資訊。

---

### SQL

這個語言結構與在 `Net.Data` 中以 `FUNCTION(DTW_SQL)` 呼叫的函數相同。

它可包含 `SQL_REPORT` 和 `SQL_MESSAGE` 陳述式，其亦來自於 DB2 WWW 連線。DB2 WWW 連線並不支援所指名的 `%SQL` 區塊。

範例：

範例 1：DB2 WWW 連線巨集

```
%SQL{  
UPDATE $(dbtb1) SET URL='$(URL)' WHERE ID=$(ID)  
%SQL_MESSAGE{  
100：“<B>所選取的 URL 已不再存在於表格中</B>。”：繼續
```

```

%}
%}

%HTML_INPUT{
<HTML>
...
%EXEC_SQL
</HTML>
%}

%HTML_REPORT{
<HTML>
...
</HTML>
%}

```

#### 範例 1：對等的 Net.Data 巨集

```

%FUNCTION(DTW_SQL) URLquery(){
UPDATE $(dbtbl) SET URL='$(URL)' WHERE ID=$(ID)
%MESSAGE{
100：“<B>所選取的 URL 已不再存在於表格中</B>。”：繼續
%}
%}

%HTML(INPUT){
<HTML>
...
@URLquery
</HTML>
%}

%HTML(REPORT){
<HTML>
...
</HTML>
%}

```

---

## SQL\_MESSAGE

這語言結構與 Net.Data MESSAGE 陳述式相同。請參閱第44頁的『MESSAGE 區塊』以取得範例。

---

## SQL\_REPORT

這語言結構與 Net.Data REPORT 陳述式相同。請參閱第48頁的『REPORT 區塊』以取得範例。

---

## SQL\_CODE

這個語言結構來自 DB2 WWW 連線，且 Net.Data 支援它以便能夠與其相容，相等於第112頁的『RETURN\_CODE』。

## 附錄C. Net.Data 作業系統參考手冊

並非每一種作業系統都支援所有的 Net.Data 特性。本章節將顯示您的作業系統支援哪些特性。一個 **X** 表示有支援該特性。

表 218. Net.Data 語言環境

語言環境	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
Direct Call					X				
純本文檔介面	X			X	X	X	X	X	X
IMS Web	X			X				X	X
Java Applets	X	X	X	X	X	X	X	X	X
Java 應用程式	X			X		X		X	X
ODBC	X	X		X	X		X	X	X
Oracle	X								X
Perl	X	X	X	X	X			X	X
REXX	X		X	X	X	X	X	X	X
SQL	X	X	X	X	X	X	X	X	X
Sybase	X								X
系統	X	X	X	X	X	X	X	X	X
Web 登記	X			X		X			X

表 219. Net.Data 架構變數

架構變數	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DB2INSTANCE	X	X	X	X			X	X	X
DB2MSG5					X				
DB2PLAN					X				
DB2SSID					X				
DefaultDBCp					X				
DSNAOINI					X				
DTW_CACHE_HOST	X								X
DTW_CACHE_MACRO					X				
DTW_CACHE_PORT	X								X
DTW_CM_PORT	X	X		X					X
DTW_DIRECT_REQUEST	X	X		X	X		X	X	X
DTW_DO_NOT_CACHE_MACRO					X				
DTW_INST_DIR	X		X	X			X	X	X
DTW_LOG_DIR	X	X	X	X	X		X	X	X
DTW_LOG_LEVEL	X	X	X	X			X	X	X
DTW_MBMODE	X	X	X	X	X		X	X	X
DTW_REMOVE_WS					X				
DTW_SHOWSQL	X	X	X	X	X	X	X	X	X
DTW_SMTP_CHARSET						X			
DTW_SMTP_SERVER	X	X	X	X	X	X	X	X	X

表 219. Net.Data 架構變數 (繼續)

架構變數	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTW_SQL_ISOLATION						X			
DTW_SQL_NAMING_MODE						X			
DTW_UNICODE	X	X		X			X	X	X
DTW_VARIABLE_SCOPE	X	X		X			X	X	X

表 220. Net.Data 變數

變數	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
ALIGN	X	X	X	X	X	X	X	X	X
DATABASE	X	X	X	X		X	X	X	X
DB_CASE	X	X	X	X	X	X	X	X	X
DB2PLAN					X				
DB2SSID					X				
DTW_APPLET_ALTTEXT	X	X	X	X	X	X	X	X	X
DTW_CURRENT_FILENAME	X	X	X	X	X	X	X	X	X
DTW_CURRENT_LAST_MODIFIED	X	X	X	X	X	X	X	X	X
DTW_DEFAULT_MESSAGE	X	X	X	X		X	X	X	X
DTW_DEFAULT_REPORT	X	X	X	X	X	X	X	X	X
DTW_EDIT_CODES						X			
DTW_HTML_TABLE	X	X	X	X	X	X	X	X	X
DTW_LOG_LEVEL	X	X	X	X			X	X	X
DTW_MACRO_FILENAME	X	X	X	X	X	X	X	X	X
DTW_MACRO_LAST_MODIFIED	X	X	X	X	X	X	X	X	X
DTW_MBMODE	X	X	X	X	X		X	X	X
DTW_MP_PATH	X	X	X	X	X	X	X	X	X
DTW_MP_VERSION	X	X	X	X	X	X	X	X	X
DTW_PAD_PGM_PARMS						X			
DTW_PRINT_HEADER	X	X	X	X	X	X	X	X	X
DTW_REMOVE_WS	X	X	X	X	X	X	X	X	X
DTW_SAVE_TABLE_IN	X	X	X	X	X	X	X	X	X
DTW_SET_TOTAL_ROWS	X	X	X	X	X	X	X	X	X
LOCATION					X				
LOGIN	X	X	X	X		X	X	X	X
Nn	X	X	X	X	X	X	X	X	X
NLIST	X	X	X	X	X	X	X	X	X
NULL_RPT_FIELD						X			
NUM_COLUMNS	X	X	X	X	X	X	X	X	X
NUM_ROWS						X			
PASSWORD	X	X	X	X		X	X	X	X
RETURN_CODE	X	X	X	X	X	X	X	X	X
ROW_NUM	X	X	X	X	X	X	X	X	X
RPT_MAX_ROWS	X	X	X	X	X	X	X	X	X

表 220. Net.Data 變數 (繼續)

變數	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
SHOWSQL	X	X	X	X	X	X	X	X	X
SQL_CODE	X	X	X	X	X	X	X	X	X
SQL_STATE	X	X	X	X	X	X	X	X	X
START_ROW_NUM	X	X	X	X	X	X	X	X	X
TOTAL_ROWS	X	X	X	X	X	X	X	X	X
TRANSACTION_SCOPE	X	X	X	X	X	X	X	X	X
V_columnName	X	X	X	X	X	X	X	X	X
VLIST	X	X	X	X	X	X	X	X	X
Vn	X	X	X	X	X	X	X	X	X

表 221. Net.Data 函數

函數	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTW_ACCEPT						X			
DTW_ADD	X	X	X	X	X	X	X	X	X
DTW_ADDQUOTE	X	X	X	X	X	X	X	X	X
DTW_ASSIGN	X	X	X	X	X	X	X	X	X
DTW_CACHE_PAGE	X								X
DTW_COMMIT						X			
DTW_CHARTOHEX						X			
DTW_CONCAT	X	X	X	X	X	X	X	X	X
DTW_DATE	X	X	X	X	X	X	X	X	X
DTW_DELSTR	X	X	X	X	X	X	X	X	X
DTW_DELWORD	X	X	X	X	X	X	X	X	X
DTW_DIVIDE	X	X	X	X	X	X	X	X	X
DTW_DIVREM	X	X	X	X	X	X	X	X	X
DTW_EXIT	X	X	X	X	X	X	X	X	X
DTW_FORMAT	X	X	X	X	X	X	X	X	X
DTW_GETCOOKIE	X	X	X	X	X	X	X	X	X
DTW_GETENV	X	X	X	X	X	X	X	X	X
DTW_GETINIDATA	X	X	X	X	X	X	X	X	X
DTW_HEXTOCHAR						X			
DTW_HTMLENCODE	X	X	X	X	X	X	X	X	X
DTW_INSERT	X	X	X	X	X	X	X	X	X
DTW_INTDIV	X	X	X	X	X	X	X	X	X
DTW_LASTPOS	X	X	X	X	X	X	X	X	X
DTW_LENGTH	X	X	X	X	X	X	X	X	X
DTW_LOWERCASE	X	X	X	X	X	X	X	X	X
DTW_MULTIPLY	X	X	X	X	X	X	X	X	X
DTW_POS	X	X	X	X	X	X	X	X	X
DTW_POWER	X	X	X	X	X	X	X	X	X
DTW_QHTMLENCODE	X	X	X	X	X	X	X	X	X

表 221. Net.Data 函數 (繼續)

函數	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTW_REVERSE	X	X	X	X	X	X	X	X	X
DTW_ROLLBACK						X			
DTW_RVTHANDLE						X			
DTW_SENDMAIL	X	X	X	X	X	X	X	X	X
DTW_SETCOOKIE	X	X	X	X	X	X	X	X	X
DTW_SETENV	X	X	X	X	X	X	X	X	X
DTW_STATIC						X			
DTW_STRIP	X	X	X	X	X	X	X	X	X
DTW_SUBSTR	X	X	X	X	X	X	X	X	X
DTW_SUBTRACT	X	X	X	X	X	X	X	X	X
DTW_SUBWORD	X	X	X	X	X	X	X	X	X
DTW_TB_APPENDROW	X	X		X	X	X	X	X	X
DTW_TB_COLS	X	X	X	X	X	X	X	X	X
DTW_TB_DELETEROW	X	X		X	X	X	X	X	X
DTW_TB_DELETECOL	X	X		X	X	X	X	X	X
DTW_TB_DLIST	X	X	X	X	X	X	X	X	X
DTW_TB_DUMPH	X	X	X	X	X	X	X	X	X
DTW_TB_DUMPV	X	X	X	X	X	X	X	X	X
DTW_TB_GETN	X	X	X	X	X	X	X	X	X
DTW_TB_GETV	X	X	X	X	X	X	X	X	X
DTW_TB_HTMLENCODE	X	X	X	X	X	X	X	X	X
DTW_TB_INPUT_CHECKBOX	X	X	X	X	X	X	X	X	X
DTW_TB_INPUT_RADIO	X	X	X	X	X	X	X	X	X
DTW_TB_INPUT_TEXT	X	X	X	X	X	X	X	X	X
DTW_TB_INSERTCOL	X	X		X	X	X	X	X	X
DTW_TB_INSERTROW	X	X		X	X	X	X	X	X
DTW_TB_LIST	X	X	X	X	X	X	X	X	X
DTW_TB_MAXROWS						X			
DTW_TB_QUERYCOLNONJ	X	X		X	X	X	X	X	X
DTW_TB_ROWS	X	X	X	X	X	X	X	X	X
DTW_TB_SELECT	X	X	X	X	X	X	X	X	X
DTW_TB_SETCOLS	X	X		X	X	X	X	X	X
DTW_TB_SETN	X	X		X	X	X	X	X	X
DTW_TB_SETV	X	X		X	X	X	X	X	X
DTW_TB_TABLE	X	X	X	X	X	X	X	X	X
DTW_TB_TEXTAREA	X	X	X	X	X	X	X	X	X
DTW_TERMINATE						X			
DTW_TIME	X	X	X	X	X	X	X	X	X
DTW_TRANSLATE	X	X	X	X	X	X	X	X	X
DTW_UPPERCASE	X	X	X	X	X	X	X	X	X

表 221. Net.Data 函數 (繼續)

函數	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTW_URLESCSEQ	X	X	X	X	X	X	X	X	X
DTW_WORD	X	X	X	X	X	X	X	X	X
DTW_WORDINDEX	X	X	X	X	X	X	X	X	X
DTW_WORDLENGTH	X	X	X	X	X	X	X	X	X
DTW_WORDPOS	X	X	X	X	X	X	X	X	X
DTW_WORDS	X	X	X	X	X	X	X	X	X
DTWF_APPEND	X			X	X	X		X	X
DTWF_CLOSE	X			X	X	X		X	X
DTWF_DELETE	X			X	X	X		X	X
DTWF_INSERT	X			X	X	X		X	X
DTWF_OPEN	X			X	X	X		X	X
DTWF_READ	X			X	X	X		X	X
DTWF_REMOVE	X			X	X	X		X	X
DTWF_SEARCH	X			X	X	X		X	X
DTWF_UPDATE	X			X	X	X		X	X
DTWF_WRITE	X			X	X	X		X	X
DTWR_ADDENTRY	X			X		X			X
DTWR_CLEARREG	X			X		X			X
DTWR_CLOSEREG						X			
DTWR_CREATEREG	X			X		X			X
DTWR_DELENTY	X			X		X			X
DTWR_DELREG	X			X		X			X
DTWR_LISTREG	X			X		X			X
DTWR_LISTSUB	X			X					X
DTWR_OPENREG						X			
DTWR_RTVENTRY	X			X		X			X
DTWR_UPDATEENTRY	X			X		X			X

表 222. Net.Data 介面

介面類型	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
FastCGI	X							X	
CGI	X	X	X	X	X	X	X	X	X
Java Beans									X
網際網路連線 API (ICAPI)	X			X	X				X
網際網路伺服器 API (ISAPI)									X
現場連線	X			X				X	X
Lotus Domino Go Web Server (GWAPI)	X			X	X				X
Netscape API (NSAPI)	X							X	X
Servlets	X				X				X

表 223. *Net.Data* 工具

工具	AIX	HP	LINUX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
管理工具	X			X					X
NetObjects Fusion Plug-ins									X
精靈	X			X			X	X	X



---

## 附錄D. 注意事項

本書是針對 IBM 在美國所提供之產品與服務開發出來的。而在其它國家中，IBM 不見得有提供本書中所提的各項產品、服務、或功能。要知道在您所在之區是否可用到這些產品與服務時，請向當地的 IBM 服務代表查詢。本書在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，其它非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

本文件中包含著 IBM 所擁有之專利或暫准專利。使用者不得享有本書內容之專利權。您可以用書面方式來查詢特許權限，來函請寄到：

臺灣國際商業機器股份有限公司  
台北市基隆路一段 206 號  
法務部

若要查詢有關二位元組 (DBCS) 資訊的特許權限事宜，請聯絡您國家的 IBM 智慧財產部門，或者用書面方式寄到：

臺灣國際商業機器股份有限公司  
台北市基隆路一段 206 號  
法務部

下列段落若與該國之法律條款抵觸，即視為不適用：IBM 就本書僅提供『交附時之現況』保證，而並不提供任何明示或默示之保證，如默示保證書籍之適售性或符合客戶之特殊使用目的；有些地區在某些固定的交易上並不接受明示或默示保證的放棄聲明，因此此項聲明不見得適用於您。

本書中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。同時，IBM 會隨時改進並 (或) 變動本書中所提及的產品及 (或) 程式。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其它程式 (包括此程式) 之間更換資訊的方式 (2) 相互使用已交換之資訊方法 若有任何問題請聯絡：

臺灣國際商業機器股份有限公司  
台北市基隆路 1 段 206 號  
法務部

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之 [ IBM 客戶合約 ] (或任何同等合約) 條款，提供本書中所說的授權程式與其所有適用的授權資料。

本書所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其它公開管道取得。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性、或任何對產品的其它主張是否完全無誤。如果您對非 IBM 產品的性能有任何的疑問，請逕向該產品的供應商查詢。

著作權授權：

本書中有一些以原始語言列印出的範例應用程式，是用來說明各種作業平台上的程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式而不必向 IBM 付費。這些範例應用程式並未經過徹底的測試，因此，IBM 不提供明示或默示其可靠性、有用性或符合特定效用的保證。您可以基於研發、使用、銷售或散布符合 IBM 應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式而不必向 IBM 付費。

---

## 商標

下列專有名詞是 IBM 公司在美國或 (及) 其它國家的商標：

AIX	IMS
AS/400	Language Environment
CBIDO	MVS/ESA
CBPDO	Net.Data
CICS	OpenEdition
CustomPac	Operating System/400
DB2	OS/2
DB2 Universal Databas	OS/390
DataJoiner	OS/400
Distributed Relational	RACF
Database Architecture	SystemPac
DRDA	
IBM	

下列專有名詞是以下其它公司的商標：

Java 及所有 Java 型的商標或標誌是 Sun Microsystems, Inc. 在美國及 (或) 其它國家的商標。

Lotus 及 Domino Go Webserver 是 Lotus Development Corporation 在美國及 (或) 其它國家的商標。

Microsoft、Windows、Windows NT 及 Windows 標誌是 Microsoft Corporation 在美國及 (或) 其它國家的商標或註冊商標。

其它公司、產品與服務名稱 (以雙星號 \*\* 表示)，可能是其它公司的商標或服務標記。

---

## 名詞解釋

### 一劃

**一致資源定址器 (uniform resource locator).** 用來指出 HTTP 伺服器並可選擇性地指出目錄以及檔案名稱的位址，例如：  
`http://www.software.ibm.com/data/net.data/index.html`。

### 七劃

**防火牆 (firewall).** 一個含有軟體的電腦，其負責保護內部的網路不讓未經授權的外來者存取。

### 八劃

**空值 (null).** 一個代表沒有資訊的特殊值。

### 十劃

**純本文檔介面 (flat file interface).** 一組 Net.Data 內建式函數，可讓您讀取與寫入純文字檔中的資料。

### 十一劃

**埠 (port).** 一個 16 位元數字，用來供 TCP/IP 和高階通訊協定或應用程式通信用。

**通用閘道介面 (Common Gateway Interface, CGI).** 一種標準的方式，可讓 Web 伺服器藉此將控制傳給應用程式並收回資料。

### 十二劃

**超文字標示語言 (hypertext markup language).** 一種用來撰寫 Web 文件的標籤語言。

**超本文轉送通信協定 (hypertext markup language).** 用於 Web 伺服器與瀏覽器間的通訊協定。

### 十三劃

**傳輸控制通信協定 / 網際網路通信協定 (Transmission Control Protocol / Internet Protocol).** 一組支援區域及廣域網路之對等連接功能的通訊協定。

**資料庫 (database).** 集結了許多表格而成的集合，也可以是表格空間及索引空間的集合。

**資料庫管理系統 (database management system, DBMS).** 一個控制資料庫的建立、組織和修改以及存取儲存於其內之資料的軟體系統。

**資料類型 (data type).** 直欄與文字的屬性。

**路徑 (path).** 用來尋找檔案的搜尋路徑。

### 十四劃

**語言環境 (language environment).** 提供 Net.Data 巨集與外部資料來源 (如 DB2) 或程式設計語言 (如 Perl) 之存取的模組。

### 十七劃

**應用程式設計介面 (application programming interface, API).** 為一功能性介面，由作業系統或由一個可分開訂購的授權程式所提供，讓您可以使用高階語言撰寫應用程式來使用作業系統或授權程式之特定資料或函數。Net.Data 支援下列具有專利的 Web 伺服器 API，可增進 CGI 處理的執行效能：ICAPI、GWAPI、ISAPI 及 NSAPI。

## A

**API.** 應用程式設計介面。Net.Data 支援三種 Web 伺服器 API，透過 CGI 處理來改善效能。

**applet.** 一種包含在 HTML 頁面中的 Java 程式。Applet 使用具有 Java 功能的瀏覽器 (如 Netscape Navigator) 而且在處理 HTML 頁面時被載入。

## C

**CGI.** 通用閘道介面。

## D

**DBMS.** 資料庫管理系統。

**Domino Go Web 伺服器.** Lotus Corp. 與 IBM 共同開發的 Web 伺服器，提供一般與安全連線。此伺服器提供 ICAPI 與 GWAPI 介面。

## G

**GWAPI.** Go Web 伺服器 API。

## H

**HTML.** 超文字標示語言。

**HTTP.** 超本文轉送通信協定。

## I

**ICAPI.** Internet 連線 API (Internet Connection API) 請參閱。

**Internet.** 一個國際公用的 TCP/IP 電腦網路。

**Intranet.** 一個位於公司防火牆內的 TCP/IP 網路。

## J

**Java.** 一種不依附作業系統之物件導向型程式設計語言，特別適用於 Internet 的應用程式。

## L

**LOB.** 大型物件。

## P

**Perl.** 一種解譯式程式設計語言。

## T

**TCP/IP.** 傳輸控制通信協定 / 網際網路通信協定。

## U

**URL.** 一致資源定址器 (Uniform resource locator)。

## W

**Web 伺服器 (Web server).** 一部執行 HTTP 伺服器軟體 (如：Internet Connection) 的電腦。

# 索引

索引順序以中文字，英文字，及特殊符號之次序排列。

## 〔一劃〕

“下一步”按鈕，RPT\_MAX\_ROWS 80  
“前一步”按鈕，RPT\_MAX\_ROWS 80  
一般函數 114  
DTW\_ADDQUOTE 116  
DTW\_CACHE\_PAGE 118  
DTW\_DATE 121  
DTW\_EXIT 123  
DTW\_GETCOOKIE 124  
DTW\_GETENV 126  
DTW\_GETINIDATA 128  
DTW\_HTMLLENCODE 129  
DTW\_QHTMLLENCODE 131  
DTW\_SENMAIL 132  
DTW\_SETCOOKIE 136  
DTW\_SETENV 139  
DTW\_TIME 140  
DTW\_URLESCSEQ 142

## 〔三劃〕

上限 53

## 〔四劃〕

內建函數 113  
支援特性表格 292  
日期格式，UTF-8 121  
日期變數 100

## 〔五劃〕

巨集  
行長度限制 3  
宣告部分 1  
展示部份 1  
格式 2  
停止處理程序 123  
常見語法元素 4  
語言結構 1  
樣本 2  
整體語法 1  
平台支援參考手冊 292

## 〔六劃〕

列示有限制的字串 63

列示變數

值區隔符號 64  
說明 63  
範例 63

名詞解釋 301

字串

有分隔設定的值 63  
條件式處理程序 28, 55  
說明 5  
數字比較 28, 55

字串函數

DTW\_ASSIGN 163  
DTW\_CHARTOHEX 164  
DTW\_CONCAT 165  
DTW\_DELSTR 166  
DTW\_HEXTOCHAR  
DTW\_HEXTOCHAR 167  
DTW\_INSERT  
DTW\_INSERT 168  
DTW\_LASTPOS 170  
DTW\_LENGTH 172  
DTW\_LOWERCASE 173  
DTW\_POS 175  
DTW\_REPLACE 177  
DTW\_REVERSE 179  
DTW\_STRIP 180  
DTW\_SUBSTR 182  
DTW\_TRANSLATE 184  
DTW\_UPPERCASE 186  
MBCS 支援 162

字串的數字比較 28, 55

字組函數

DTW\_DELWORD 189  
DTW\_SUBWORD 191  
DTW\_WORD 193  
DTW\_WORDINDEX 194  
DTW\_WORDLENGTH 195  
DTW\_WORDPOS 196  
DTW\_WORDS 198  
MBCS 支援 188

存取純本文檔 238

安全

密碼 96  
登入 ID 94

安全建議，FFI\_PATH 240

有分隔設定的值串 63

行長度限制，巨集 3

## 〔七劃〕

位置, 純本文檔 238  
作業系統參考手冊 292  
系統語言環境, 傳送參數 21

## 〔八劃〕

併入檔 34  
函數  
    一般 114  
    字串 162  
    字組 188  
    命名慣例 113  
    表格 199  
    持續 280  
    純本文檔介面 (FFI) 238  
    傳送值群組 64  
    算術 144  
    說明 113  
    Web 登記 262  
函數呼叫  
    使用 INOUT 變數 24  
    處理表格列 51  
    語法 23  
    說明 23  
    輸出格式化 48  
呼叫  
    外部程式 14  
    函數 23  
呼叫 FFI 語言環境 238  
注意事項 299  
表格  
    結果在 HTML 78  
    Net.Data, 設定列數 79  
表格函數

DTW\_TB\_APPENDROW 200  
DTW\_TB\_COLS 201  
DTW\_TB\_DELETECOL 202  
DTW\_TB\_DELETEROW 203  
DTW\_TB\_DLIST 204  
DTW\_TB\_DUMPV 207  
DTW\_TB\_GETN 209  
DTW\_TB\_GETV 211  
DTW\_TB\_HTMLENCODE 213  
DTW\_TB\_INPUT\_CHECKBOX 214  
DTW\_TB\_INPUT\_RADIO 216  
DTW\_TB\_INPUT\_TEXT 218  
DTW\_TB\_INSERTCOL 220  
DTW\_TB\_INSERTROW 221  
DTW\_TB\_LIST 222  
DTW\_TB\_QUERYCOLNONJ 224

表格函數 (繼續)

DTW\_TB\_ROWS 226  
DTW\_TB\_SELECT 227  
DTW\_TB\_SETCOLS 229  
DTW\_TB\_SETN 230  
DTW\_TB\_SETV 232  
DTW\_TB\_TABLE 234  
DTW\_TB\_TEXTAREA 236

表格處理程序變數

為 SQL 語言環境而設 91  
說明 65  
NLIST 67  
NUM\_COLUMNS 68  
NUM\_ROWS 69  
Nn 66  
ROW\_NUM 70  
TOTAL\_ROWS 71  
VLIST 73  
Vn 74  
V\_columnName 72

表格變數

說明 64  
範例 64

表頭 34

## 〔九劃〕

宣告部份, 巨集 1  
持續巨集函數  
    DTW\_ACCEPT 281  
    DTW\_COMMIT 283  
    DTW\_ROLLBACK 284  
    DTW\_RTVHANDLE 285  
    DTW\_STATIC 286  
    DTW\_TERMINATE 287  
架構 FFI 語言環境 239  
限制資料庫存取 94, 96

## 〔十劃〕

展示部份, 巨集 1  
純本文檔  
    在現行目錄中建立 239  
    存取 238  
    存取時的建議 239  
    安全建議 240  
    位置  
        現行目錄 238, 239  
        FFL\_PATH 238  
    定義 238  
    架構規則 239  
    區隔字元 240  
    授權需求 240

純本文檔 (繼續)

符合 FFI\_PATH 239

絕對路徑 239

資料來源 238

鎖定檔案 241

迴路 55

## 〔十一劃〕

區域 DB2 子系統的 ID 87

區隔字元, FFI 語言環境

ASCIITEXT 240

DELIMITED 240

參數, 傳送 21

執行效能, DTW\_EXIT 123

執行變數

作為變數參照 62

與參數 62

說明 61

範例 62

從巨集傳送電子郵件 132

捲動, 使用“下一步”及“前一步”按鈕 80

授權需求, FFI\_PATH 240

條件式字串處理程序 28, 55

條件變數

以 LIST 陳述式定義 60

以變數參照定義 60

說明 60

範例 63

現行目錄, 決定純本文檔 239

規劃連接 DB2 子系統 86

設定 SQL 指令字體 85

設定大寫字體 85

設定小寫字體 85

連接 DB2 子系統

位置 93

的子系統 ID 87

DB2 規劃 86

連接 DB2 子系統的子系統 ID 87

連接 DB2 子系統的位置 93

連接資料庫, DATABASE 變數 84

## 〔十二劃〕

備註區塊

語法 7

說明 7

報告

置換 Net.Data 預設值 77

製作格式 48

報告變數

說明 75

ALIGN 76

報告變數 (繼續)

DTW\_DEFAULT\_REPORT 77

DTW\_HTML\_TABLE 78

RPT\_MAX\_ROWS 79

START\_ROW\_NUM 80

絕對路徑, 純本文檔的 239

## 〔十三劃〕

傳送值群組 64

傳送參數, 「系統」語言環境 21

資料庫一致性, 異動範圍 99

預設的文字訊息 103

## 〔十四劃〕

算術函數

DTW\_ADD 145

DTW\_DIVIDE 147

DTW\_DIVREM 149

DTW\_FORMAT 151

DTW\_INTDIV 154

DTW\_MULTIPLY 156

DTW\_POWER 158

DTW\_SUBTRACT 160

語言結構

巨集

語法 1

說明 5

字串 5

函數呼叫 23

常見語法元素 4

備註區塊 7

變數名稱 4

變數參照 4

DB2 WWW 連線 291

DEFINE 區塊或陳述式 9

ENVVAR 陳述式 13

EXEC 區塊或陳述式 14

FUNCTION 區塊 16

HTML 區塊 26

IF 區塊 28

INCLUDE 陳述式 34

INCLUDE\_URL 陳述式 36

LIST 陳述式 38

MACRO\_FUNCTION 區塊 40

MESSAGE 區塊 44

REPORT 區塊 48

ROW 區塊 51

TABLE 陳述式 53

WHILE 區塊 55

語言環境變數

說明 83

語言環境變數 (繼續)

DATABASE 84  
DB2PLAN 86  
DB2SSID 87  
DB\_CASE 85  
DTW\_APPLET\_ALTTEXT 88  
DTW\_EDIT\_CODES 89  
DTW\_MBMODE 107  
DTW\_PAD\_PGM\_PARMS 90  
DTW\_SAVE\_TABLE\_IN 91  
DTW\_SET\_TOTAL\_ROWS 92  
LOCATION 93  
LOGIN 94  
NULL\_RPT\_FIELD 95  
PASSWORD 96  
SHOWSQL 97  
SQL\_STATE 98  
TRANSACTION\_SCOPE 99

遠端 DB2 子系統的位置 93

## 〔十五劃〕

標底 34

## 〔十六劃〕

錯誤處理 44

## 〔十七劃〕

檔案位置變數 100

環境變數

說明 61  
範例 61  
ENVVAR 陳述式 13

隱藏變數

步驟 62  
說明 62  
範例，在 HTML 套表 62

隱藏變數名稱 62

## 〔十八劃〕

鎖定檔案, FFI 函數 241

雜項變數

說明 100  
DTW\_CURRENT\_FILENAME 101  
DTW\_CURRENT\_LAST\_MODIFIED 102  
DTW\_DEFAULT\_MESSAGE 103  
DTW\_MACRO\_LAST\_MODIFIED 106  
DTW\_MP\_PATH 108  
DTW\_MP\_VERSION 109

雜項變數 (繼續)

DTW\_PRINT\_HEADER 110  
DTW\_REMOVE\_WS 111  
RETURN\_CODE 112

## 〔二十劃〕

釋放檔案, FFI 函數 241

## 〔二十三劃〕

變數

列示 63  
表格 64, 65  
執行 61  
條件 60  
報告 75  
語言環境 83  
環境 61  
隱藏 62  
雜項 100  
Net.Data, 概觀 59

變數名稱 4

變數參照 4

顯示 SQL 狀態 98

## A

ALIGN 76  
APPLET 標籤，替代文字 88

## C

cookies

傳送 110  
DTW\_GETCOOKIE 124  
DTW\_PRINT\_HEADER 110  
DTW\_SETCOOKIE 136

## D

DATABASE 84  
DB2 WWW 連線，語言結構 291  
DB2PLAN 86  
DB2SSID 87  
DB\_CASE 85  
DEFINE 區塊  
語法 9  
說明 9  
DEFINE 陳述式  
語法 9  
說明 9



DTWF\_APPEND 241  
 DTWF\_CLOSE 241, 244  
 DTWF\_DELETE 245  
 DTWF\_INSERT 247  
 DTWF\_OPEN 241, 249  
 DTWF\_READ 251  
 DTWF\_REMOVE 254  
 DTWF\_SEARCH 255  
 DTWF\_UPDATE 258  
 DTWF\_WRITE 260  
 DTWR\_ADDENTRY 262  
 DTWR\_CLEARREG 265  
 DTWR\_CLOSEREG 266  
 DTWR\_CREATEREG 267  
 DTWR\_DELENTY 269  
 DTWR\_DELREG 270  
 DTWR\_LISTREG 271  
 DTWR\_LISTSUB 273  
 DTWR\_OPENREG 275  
 DTWR\_RTVENTRY 276  
 DTWR\_UPDATEENTRY 278  
 DTW\_ACCEPT 281  
 DTW\_ADD 144  
 DTW\_ADDQUOTE 116  
 DTW\_APPLET\_ALTTEXT 88  
 DTW\_ASSIGN 66, 162, 163  
 DTW\_CACHE\_PAGE 118  
 DTW\_CHARTOHEX 164  
 DTW\_COMMIT 283  
 DTW\_CONCAT 165  
 DTW\_CURRENT\_FILENAME 101  
 DTW\_CURRENT\_LAST\_MODIFIED 102  
 DTW\_DATE 121  
 DTW\_DEFAULT\_MESSAGE 103  
 DTW\_DEFAULT\_REPORT 77  
 DTW\_DELSTR 166  
 DTW\_DELWORD 188  
 DTW\_DIVIDE 147  
 DTW\_DIVREM 149  
 DTW\_EDIT\_CODES 89  
 DTW\_FORMAT 151  
 DTW\_GETCOOKIE 124  
 DTW\_GETENV 126  
 DTW\_GETINIDATA 128  
 DTW\_HEXTOCHAR 167  
 DTW\_HTMLENCODER 129  
 DTW\_HTML\_TABLE 78  
 DTW\_INSERT 168  
 DTW\_INTDIV 154  
 DTW\_LASTPOS 170  
 DTW\_LENGTH 172  
 DTW\_LOG\_LEVEL 104

DTW\_LOWERCASE 173  
 DTW\_MACRO\_FILENAME 105  
 DTW\_MACRO\_LAST\_MODIFIED 106  
 DTW\_MBMODE 107  
 DTW\_MP\_PATH 108  
 DTW\_MP\_VERSION 109  
 DTW\_MULTIPLY 156  
 DTW\_PAD\_PGM\_PARMS 90  
 DTW\_POS 175  
 DTW\_POWER 158  
 DTW\_PRINT\_HEADER 110  
 DTW\_QHTMLENCODE 131  
 DTW\_REMOVE\_WS 111  
 DTW\_REPLACE 177  
 DTW\_REVERSE 179  
 DTW\_ROLLBACK 284  
 DTW\_RTVHANDLE 285  
 DTW\_SAVE\_TABLE\_IN 91  
 DTW\_SENMAIL 132  
 DTW\_SETCOOKIE 136  
 DTW\_SETENV 139  
 DTW\_SET\_TOTAL\_ROWS 92  
 DTW\_STATIC 286  
 DTW\_STRIP 180  
 DTW\_SUBSTR 182  
 DTW\_SUBTRACT 160  
 DTW\_SUBWORD 191  
 DTW\_TB\_APPENDROW 200  
 DTW\_TB\_COLS 201  
 DTW\_TB\_deleteCOL 202  
 DTW\_TB\_DELETEROW 203  
 DTW\_TB\_DLIST 204  
 DTW\_TB\_DUMPH 206  
 DTW\_TB\_DUMPV 207  
 DTW\_TB\_GETN 209  
 DTW\_TB\_GETV 211  
 DTW\_TB\_HTMLENCODER 213  
 DTW\_TB\_INPUT\_CHECKBOX 214  
 DTW\_TB\_INPUT\_RADIO 216  
 DTW\_TB\_INPUT\_TEXT 218  
 DTW\_TB\_INSERTCOL 220  
 DTW\_TB\_INSERTROW 221  
 DTW\_TB\_LIST 220  
 DTW\_TB\_QUERYCOLNONJ 224  
 DTW\_TB\_ROWS 226  
 DTW\_TB\_SELECT 227  
 DTW\_TB\_SETCOLS 229  
 DTW\_TB\_SETN 230  
 DTW\_TB\_SETV 232  
 DTW\_TB\_TABLE 234  
 DTW\_TB\_TEXTAREA 236  
 DTW\_TERMINATE 287

DTW\_TIME 140  
DTW\_TRANSLATE 184  
DTW\_UPPERCASE 186  
DTW\_URLESCSEQ 142  
DTW\_WORD 193  
DTW\_WORDINDEX 194  
DTW\_WORDLENGTH 195  
DTW\_WORDPOS 196  
DTW\_WORDS 198

## E

ENVVAR 陳述式 61  
    語法 13  
    說明 13  
EXEC 區塊  
    語法 14  
    說明 14  
EXEC 陳述式 61  
    語法 14  
    說明 14  
EXEC\_PATH 14  
EXEC\_SQL 291

## F

FFI 函數  
    鎖定檔案 241  
    釋放檔案 241  
    DTWF\_APPEND 242  
    DTWF\_CLOSE 244  
    DTWF\_DELETE 245  
    DTWF\_INSERT 247  
    DTWF\_OPEN 249  
    DTWF\_READ 251  
    DTWF\_REMOVE 254  
    DTWF\_SEARCH 255  
    DTWF\_UPDATE 258  
    DTWF\_WRITE 260  
FFI 語言環境  
    存取檔案 238  
    安全建議 240  
    架構規則 239  
    區隔字元 240  
    授權需求 240  
    現行目錄 239  
    檔案位置 238  
FFI\_PATH  
    存取純本文檔 238  
    安全建議 240  
    使路徑與 *filename* 參數相符 239  
    架構規則 239  
    純本文檔位置 238

FFI\_PATH (繼續)  
    語法 238  
FUNCTION 區塊  
    語法 17  
    說明 16

## H

HTML  
    套表，輸入使用者 ID 94  
    套表，輸入密碼 96  
    隱藏變數名稱 62  
    顯示表格結果於 78  
HTML 區塊  
    語法 26  
    說明 26  
HTML\_INPUT 區塊 291  
HTML\_REPORT 區塊 291

## I

IF 區塊  
    語法 28  
    說明 28  
IN 關鍵字 18, 41, 113  
INCLUDE 陳述式  
    語法 34  
    說明 34  
INCLUDE\_PATH 34  
INCLUDE\_URL 陳述式  
    語法 36  
    說明 36  
INOUT 關鍵字 18, 41, 113  
INOUT 變數  
    範例 24

## L

LIST 陳述式  
    語法 38  
    說明 38  
LOCATION 93  
LOGIN 94

## M

MACRO\_FUNCTION 區塊  
    語法 40  
    說明 40  
MBCS 支援函數  
    字串函數 162  
    字組函數 188

MESSAGE 區塊  
語法 44  
說明 44

## N

Net.Data 表格  
上限 53  
定義 53  
NLIST 67  
NULL\_RPT\_FIELD 95  
NUM\_COLUMNS 68  
NUM\_ROWS 69  
Nn 66

## O

OUT 關鍵字 18, 41, 113

## P

PASSWORD 96

## R

REPORT 區塊  
表格變數 64  
語法 48  
說明 48  
ALIGN 76  
DTW\_DEFAULT\_REPORT 77  
DTW\_HTML\_TABLE 78  
NLIST 67  
NUM\_COLUMNS 68  
NUM\_ROWS 69  
Nn 66  
RPT\_MAX\_ROWS 79  
START\_ROW\_NUM 80  
TOTAL\_ROWS 71  
RETURNS 關鍵字 18, 41  
RETURN\_CODE 112  
ROW 區塊  
語法 51  
說明 51  
NLIST 67  
NUM\_COLUMNS 68  
NUM\_ROWS 69  
Nn 66  
ROW\_NUM 70  
TOTAL\_ROWS 71  
Vn 73, 74  
V\_columnName 72

ROW\_NUM 70  
RPT\_MAX\_ROWS 79

## S

SHOWSQL 97  
SQL  
設定指令字體 85  
隱藏或顯示 97  
SQL 區塊 291  
SQL\_CODE 292  
SQL\_MESSAGE 區塊 292  
SQL\_REPORT 區塊 292  
SQL\_STATE 98  
START\_ROW\_NUM 80

## T

TABLE 陳述式 64  
語法 53  
說明 53  
TOTAL\_ROWS 71  
TRANSACTION\_SCOPE 99

## U

UTF-8 格式  
日期 121

## V

VLIST 73  
Vn 74  
V\_columnName 72

## W

Web 登記函數  
DTWR\_ADDENTRY 263  
DTWR\_CLEARREG 265  
DTWR\_CLOSEREG 266  
DTWR\_CREATEREG 267  
DTWR\_DELENTY 269  
DTWR\_DELREG 270  
DTWR\_LISTREG 271  
DTWR\_LISTSUB 273  
DTWR\_OPENREG 275  
DTWR\_RTVENTRY 276  
DTWR\_UPDATEENTRY 278  
Web 瀏覽器的替代文字 88  
WHILE 區塊  
語法 55

WHILE 區塊 (繼續)  
說明 55

折疊線

台北市敦化南路一段二號十二樓

臺灣國際商業機器股份有限公司  
中文支援中心 啟

廣 告 回 信
台灣北區郵政管理局 登記
北台字第 0587 號

(免貼郵票)

寄件人 姓名：  
地址：

寄

折疊線

# 讀者意見表

爲使本書盡善盡美，本公司極需您寶貴的意見；懇請您使用過後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號（✓）；我們會在下一版中，作適當修訂，謝謝您的合作！

評估項目	評 估 意 見	備 註
正 確 性	內容說明與實際程序是否符合 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	參考書目是否正確 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
一 致 性	文句用語及風格，前後是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	實際畫面訊息與本書所提之畫面訊息是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
完 整 性	是否遺漏您想知道的項目 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	字句、章節是否有遺漏 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
術語使用	術語之使用是否恰當 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	術語之使用，前後是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
可 讀 性	文句用語是否通順 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	有否不知所云之處 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
內容說明	內容說明是否詳盡 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	例題說明是否詳盡 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
排版方式	本書的形狀大小，版面安排是否方便使用 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	字體大小，顏色編排，是否有助於閱讀 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
目錄索引	目錄內容之編排，是否便於查考 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	索引語錄之排定，是否便於查考 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	※評估意見為 "否" 者，請於備註欄說明。	

其他：（篇幅不夠時，請另紙說明。）

[illegible]

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。





Printed in Singapore