



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281頁的『附錄C. 注意事項』中的資訊。

第四版 (1998 年 9 月)

© Copyright International Business Machines Corporation 1997, 1998. All rights reserved.

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
第1章 Net.Data 巨集語言結構	1
Net.Data 巨集檔語法	1
常見語法元素	5
變數名稱	5
變數參照	5
字串	6
巨集語言結構	6
備註區塊	8
DEFINE 區塊或陳述式	10
ENVVAR 陳述式	15
EXEC 區塊或陳述式	16
FUNCTION 區塊	18
函數呼叫 (@)	25
HTML 區塊	28
IF 區塊	31
INCLUDE 陳述式	38
INCLUDE_URL 陳述式	41
LIST 陳述式	43
MACRO_FUNCTION 區塊	45
MESSAGE 區塊	49
REPORT 區塊	54
ROW 區塊	57
TABLE 陳述式	60
WHILE 區塊	62
第2章 變數	67
使用者定義的變數	68
條件式變數	68
環境變數	69
執行變數	70
隱藏變數	71
列示變數	72
表格變數	73
Net.Data 表格處理程序變數	74
Nn	75
NLIST	76

NUM_COLUMNS.	77
NUM_ROWS	78
ROW_NUM	79
TOTAL_ROWS	80
V_columnName	81
VLIST	82
Vn	83
Net.Data 報告變數	84
ALIGN	85
DTW_DEFAULT_REPORT.	86
DTW_HTML_TABLE	87
RPT_MAX_ROWS	88
START_ROW_NUM.	89
Net.Data 語言環境變數	92
DATABASE	93
DB_CASE	95
DB2PLAN	96
DB2SSID	97
DTW_APPLET_ALTTEXT	98
DTW_EDIT_CODES	99
DTW_MBMODE	100
DTW_SAVE_TABLE_IN	101
DTW_SET_TOTAL_ROWS.	102
LOCATION	103
LOGIN	104
NULL_RPT_FIELD	105
PASSWORD	106
SHOWSQL	107
SQL_STATE	108
TRANSACTION_SCOPE	109
Net.Data 雜項變數	111
DTW_CURRENT_FILENAME.	112
DTW_CURRENT_LAST_MODIFIED	113
DTW_DEFAULT_MESSAGE	114
DTW_LOG_LEVEL	115
DTW_MACRO_FILENAME	116
DTW_MACRO_LAST_MODIFIED	117
DTW_MP_PATH	118
DTW_MP_VERSION.	119
DTW_PRINT_HEADER.	120
DTW_REMOVE_WS.	121
RETURN_CODE	122
第3章 Net.Data 內建函數.	123
函數名稱	123
輸入與輸出參數	123

函數結果格式	124
函數參數規則	124
一般函數	124
DTW_ADDQUOTE	126
DTW_CACHE_PAGE	128
DTW_DATE	132
DTW_EXIT.	134
DTW_GETCOOKIE	135
DTW_GETENV	137
DTW_GETINIDATA.	138
DTW_HTMLENCODER	139
DTW_QHTMLENCODER.	141
DTW_SENDMAIL	142
DTW_SETCOOKIE	146
DTW_SETENV	149
DTW_TIME	150
DTW URLESCSEQ	152
算術函數	154
DTW_ADD.	155
DTW_DIVIDE.	156
DTW_DIVREM	158
DTW_FORMAT	160
DTW_INTDIV.	163
DTW_MULTIPLY	164
DTW_POWER.	165
DTW_SUBTRACT	166
字串函數	168
DTW_ASSIGN.	169
DTW_CONCAT	170
DTW_DELSTR	171
DTW_INSERT.	172
DTW_LASTPOS	174
DTW_LENGTH	175
DTW_LOWERCASE.	176
DTW_POS	177
DTW_REVERSE	178
DTW_STRIP	179
DTW_SUBSTR	180
DTW_TRANSLATE	182
DTW_UPPERCASE	184
字組函數	185
DTW_DELWORD	186
DTW_SUBWORD	188
DTW_WORD	190
DTW_WORDINDEX.	191
DTW_WORDLENGTH	192

DTW_WORDPOS	193
DTW_WORDS	195
表格函數	196
DTW_TB_APPENDROW	197
DTW_TB_COLS	198
DTW_TB_DELETEROW	199
DTW_TB_DLIST	200
DTW_TB_DUMPH	202
DTW_TB_DUMPV	203
DTW_TB_GETN	204
DTW_TB_GETV	205
DTW_TB_HTMLENCODE	206
DTW_TB_INPUT_CHECKBOX	207
DTW_TB_INPUT_RADIO	208
DTW_TB_INPUT_TEXT	209
DTW_TB_INSERTCOL	211
DTW_TB_INSERTROW	212
DTW_TB_LIST	213
DTW_TB_MAXROWS	215
DTW_TB_QUERYCOLNONJ	216
DTW_TB_ROWS	217
DTW_TB_SELECT	218
DTW_TB_SETCOLS	219
DTW_TB_SETN	220
DTW_TB_SETV	221
DTW_TB_TABLE	222
DTW_TB_TEXTAREA	224
純本文檔介面函數	225
存取純本文檔資料來源	225
純本文檔介面區隔字元	228
鎖定檔案	228
DTWF_APPEND	230
DTWF_CLOSE	232
DTWF_DELETE	233
DTWF_INSERT	235
DTWF_OPEN	237
DTWF_READ	239
DTWF_REMOVE	241
DTWF_SEARCH	242
DTWF_UPDATE	244
DTWF_WRITE	246
Web 登記函數	248
DTWR_ADDENTRY	249
DTWR_CLEARREG	250
DTWR_CLOSEREG	251
DTWR_CREATEREG	252

DTWR_DELENTY	253
DTWR_DELREG	254
DTWR_LISTREG	255
DTWR_LISTSUB	256
DTWR_OPENREG	257
DTWR_RTVENTRY	258
DTWR_UPDATEENTRY	260
持續巨集函數	261
DTW_ACCEPT	262
DTW_COMMIT	264
DTW_ROLLBACK	265
DTW_RTVHANDLE	266
DTW_STATIC	267
DTW_TERMINATE	268
附錄A. DB2 WWW 連線	269
EXEC_SQL	269
HTML_INPUT	269
HTML_REPORT	269
SQL	269
SQL_MESSAGE	270
SQL_REPORT	271
SQL_CODE	271
附錄B. Net.Data 作業系統參考手冊	273
附錄C. 注意事項	281
商標	282
名詞解釋	283
索引	285

前言

感謝您選擇 Net.Data 版本 2，IBM 建立動態網頁的開發工具！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。

Net.Data 版本 2 提供了明顯地改進執行效能與新增供您建置及部署網際網路商業解決方案的特性。

關於 Net.Data

使用 IBM 的 Net.Data 產品，您可以使用從關聯式與非關聯式資料庫管理系統（DBMS）中的資料建立動態的網頁，其中包含 DB2、IMS 與 ODBC-enabled 資料庫，及以 Java、JavaScript、Perl、C、C++ 與 REXX 等程式設計語言寫成之應用程式。

您可以將 Net.Data 想像成一個在 Web 伺服器上當作 middleware 執行的巨集處理器。您可以撰寫稱為巨集的 Net.Data 應用程式，依使用者輸入的資料、您資料庫的現行狀態、現有企業邏輯與您巨集中的因數等自行設定的內容，此應用程式經 Net.Data 解譯後來建立動態的網頁。

由 Netscape 或 Internet Explorer 等瀏覽器，以 URL（通用資源位置）形式發出的要求，到達 Web 伺服器，然後將要求轉送給 Net.Data 執行。Net.Data 找到該巨集並加以執行，然後依您所寫的函數，建置一個自行設定的網頁。這些函數可以：

- 將企業邏輯封裝在 Perl scripts、C 與 C++ 應用程式、或 REXX 程式中
- 存取資料庫，如 DB2

Net.Data 支援工業標準介面，如「超本文轉送通信協定（HTTP）」與「通用閘道介面（CGI）」。HTTP 使用在瀏覽器與 Web 伺服器之間，而 CGI 則使用在 Web 伺服器與 Net.Data 之間。如此可以讓您使用 Net.Data 時可以選取您喜好的瀏覽器或 Web 伺服器。Net.Data 也支援 FastCGI 與多重作業系統上的主要 Web 伺服器 API。

關於本書

本書將對 Net.Data 語言架構、變數、和功能用法以及語法做一般性的說明。

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

使用於本書中的範例將會盡量簡化，以說明特定的觀念，而非顯現 *Net.Data* 結構的每種用法。某些範例只是片段而已，並不可單獨作業。

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。

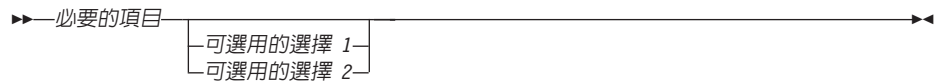
▶▶——必要的項目——
┐可選用的項目┌——▶▶

- 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。



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，而其他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、數學運算子或其他符號，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

本章說明使用在 Net.Data 巨集檔中的 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，並執行其他特殊作業，例如併入檔。

本章說明：

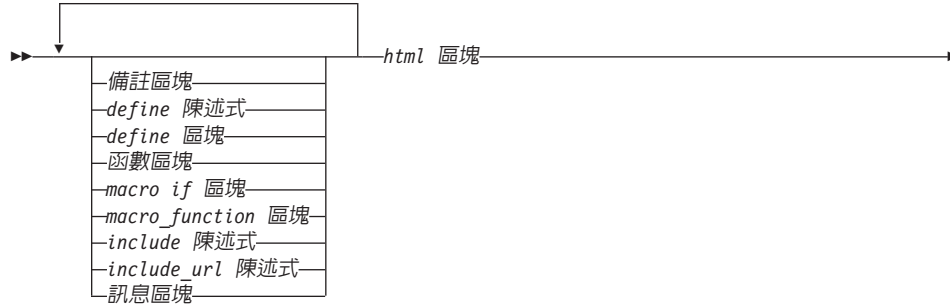
- 『Net.Data 巨集檔語法』
- 第5頁的『常見語法元素』
- 第6頁的『巨集語言結構』

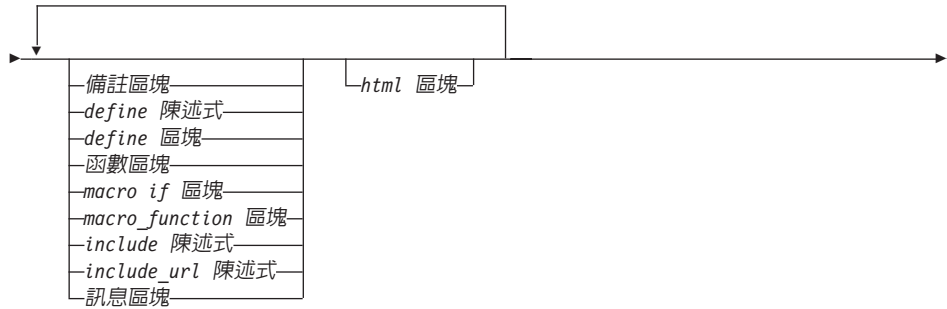
Net.Data 巨集檔語法

Net.Data 巨集是一個純文字檔案，含有一系列的 Net.Data 巨集語言結構，可用於：

- 設定 Web 網頁的佈置
- 定義變數與函數
- 呼叫定義於巨集檔中或由 Net.Data 傳遞給語言環境進行處理的函數。
- 將處理程序輸出製作成 HTML 格式，並傳回 Web 瀏覽器。

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1章 Net.Data 巨集語言結構』。





Net.Data 巨集包含兩個部分：宣告部分以及 HTML 部分。您可以任何順序來多次使用這些部分。

- 宣告部份包含巨集檔中變數及函數的定義。
- HTML 部份包含 HTML 區塊，而區塊中含有 HTML 陳述式，可用於設定 Web 首頁的佈置。這個部份並含有報告區段。

圖1顯示宣告及巨集檔的 HTML 部份。

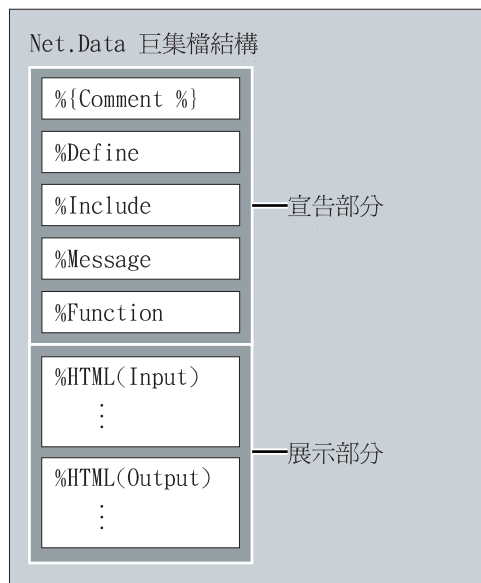


圖 1. 巨集檔結構

必須先定義在宣告或 HTML 部份中使用的變數及函數，這些變數及函數才可被用於變數參照或函數呼叫。

第4頁的圖2 示範巨集檔的部份。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. 巨集檔模版格式

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，及其他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269頁的『附錄A. DB2 WWW 連線』，但仍建議您使用 Net.Data 語言結構。

這些範例顯示許多有關語言結構、變數、函數、和在您巨集檔案中的其他元素之使用方式。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6頁的『字串』

變數名稱

目的：

識別一或多個名稱；每一個後續的名稱連接在句點之後(.)。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何組合。

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。

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。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74頁的『Net.Data 表格處理程序變數』。）。

語法：



變數參照

目的：

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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。

變數被參考前必需先被定義於 Net.Data 巨集中。如果未定義變數，則傳回空字串。

語法：

►►\$—(—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。變數參照目前解譯為字串的一部份，且不會解譯變數參照或函數呼叫，除非另有指示。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。

若要在引號內的字串中指定雙引號，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。

僅適用於 OS/400 使用者：作為函數引數的字串、項目與變數可含有變數參照及函數呼叫。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

本章說明使用在 Net.Data 巨集檔中的語言結構。

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，並提供其他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8頁的『備註區塊』
- 第10頁的『DEFINE 區塊或陳述式』
- 第15頁的『ENVVAR 陳述式』
- 第16頁的『EXEC 區塊或陳述式』
- 第18頁的『FUNCTION 區塊』
- 第25頁的『函數呼叫 (@)』
- 第28頁的『HTML 區塊』
- 第31頁的『IF 區塊』
- 第38頁的『INCLUDE 陳述式』
- 第41頁的『INCLUDE_URL 陳述式』
- 第43頁的『LIST 陳述式』
- 第45頁的『MACRO_FUNCTION 區塊』
- 第49頁的『MESSAGE 區塊』
- 第54頁的『REPORT 區塊』
- 第57頁的『ROW 區塊』
- 第60頁的『TABLE 陳述式』
- 第62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 區塊可以使用在巨集檔中任何地方，所以沒有在其他語法圖內製作文件。

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

備註可以放置在一個 Net.Data 巨集中任何兩個 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="next">
%endif
%if (submit == "both" || submit == "prev_only")
  <input name="submit" type="submit" value="previous">
%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 %}
%}

```

DEFINE 區塊或陳述式

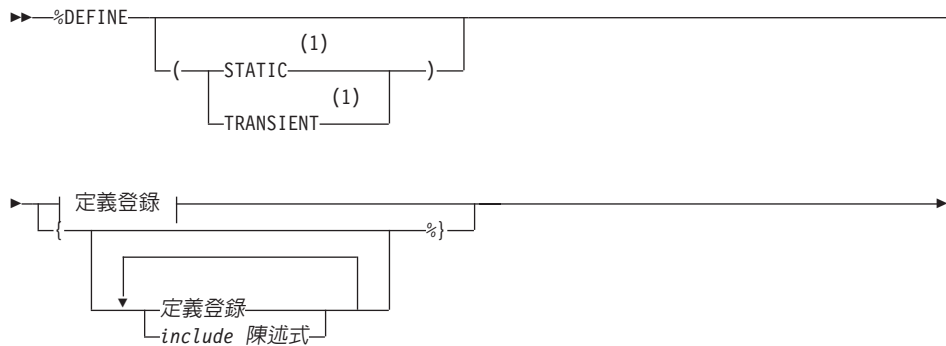
目的

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。

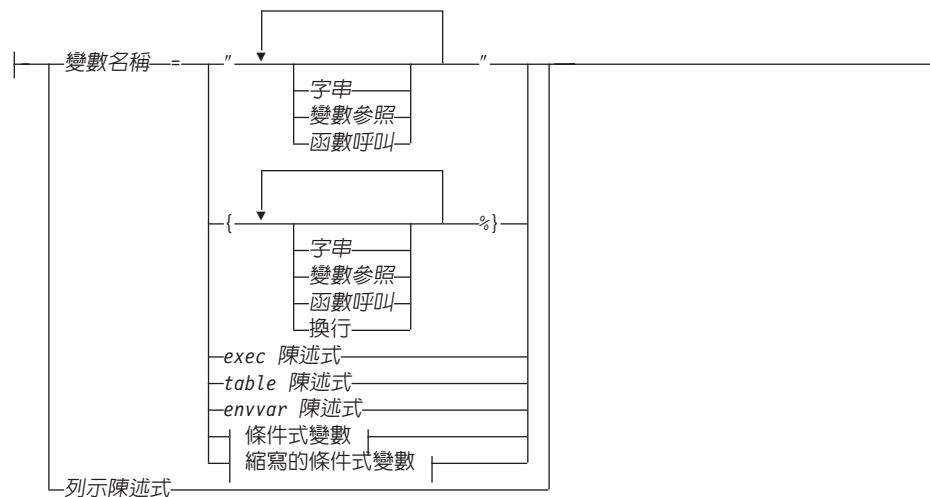
- 使用陳述式，可以一次定義一個變數
- 使用區塊，可以定義數個變數

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。

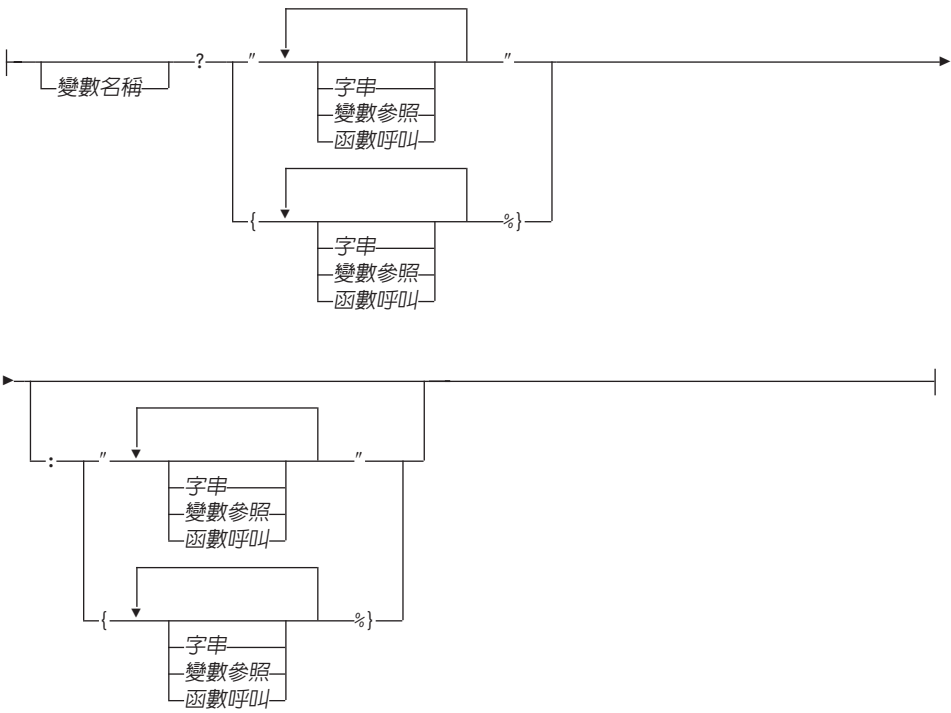
語法



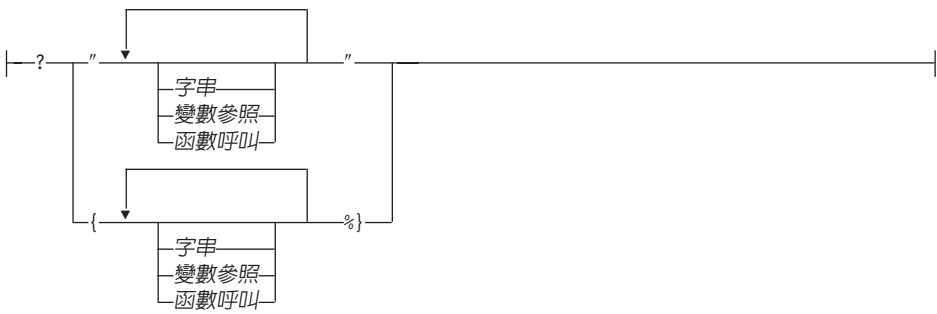
定義登錄



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

可為一或多個名稱，每一個額外名稱連接在句點之後(.)。有關語法的詳細資訊，請參閱第5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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

函數呼叫

呼叫一或多個先前定義的 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25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16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60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15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43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38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 巨集宣告部份的其他區塊之外。

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。使用這個方法來參考環境變數比使用 DTW_GETENV 更有效率。有關 DTW_GETENV 的詳細資訊，請參閱第137頁的『DTW_GETENV』。

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。這個變數可取得巨集檔中任何地方的環境變數值。

限制

ENVVAR 陳述式不可以包含其他元素。

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* 在巨集檔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的程式：

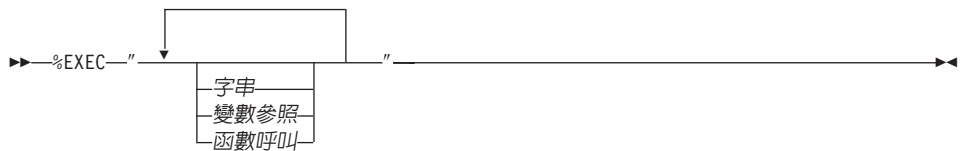
1. 它會搜尋 *Net.Data* 起始設定檔案中的 `EXEC_PATH`。有關 `EXEC_PATH` 的詳細資訊，請參閱 *Net.Data* 管理程式設計指南中的架構章節。
2. 如果 *Net.Data* 找不到該程式，它會搜尋系統所定義的目錄。如果找到了可執行的程式，*Net.Data* 會執行該程式。

授權要訣： 確定 Web 伺服器具有存取權，可以使用任何被 EXEC 陳述式或區塊所參考到的檔案。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。

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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

函數呼叫

呼叫一或多個先前定義的 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25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

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 從巨集檔呼叫來的次常式。 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 區塊時使用的方法，取決於該個特定語言環境。僅有 REXX、系統、以及 Perl Net.Data 所支援的語言環境，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.prl %}  
%}
```


- 使用變數參照來指定 %{ 字元：

```
%define percent_openbrace = "%{"

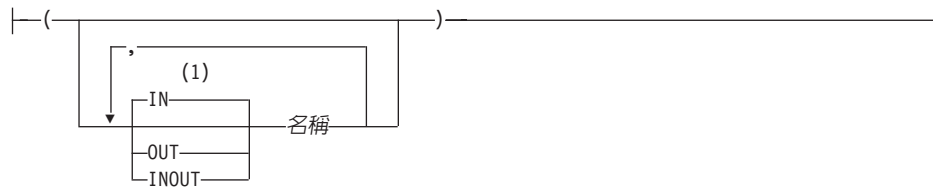
%function(DTW_PERL) func() {
    ...
    for $num_words (sort bynumber keys $(percent_openbrace) $Rtitles{$num} )) {
        &make_links($Rtitles{$num}{$num_words});
    }
    ...
}%
```

語法

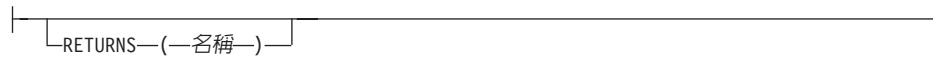
►►%FUNCTION—(—語言環境—)—函數名稱— 參數傳送規格 —————►

┌──┴──┐
└──傳回規格──┐{──┴──┐ 函數本體 └──%──┐—————►

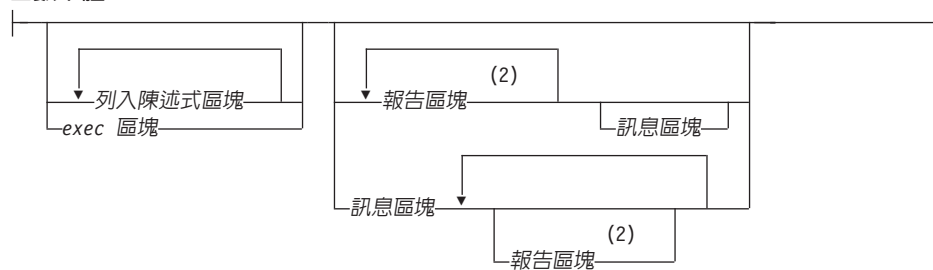
參數傳送規格



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：
 - 當處理會傳回 OS/2, Windows NT 與 UNIX 作業系統的多個結果設定的儲存程序時的資料庫語言環境。
 - 呼叫 OS/400 作業系統的任何語言環境的函數。

值

%FUNCTION

此關鍵字用於定義 Net.Data 從巨集檔呼叫來的次常式。

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 語言環境參考手冊*。

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。

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。

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。有關詳細資訊，請參閱 *Net.Data 語言環境參考手冊*。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。 **限制：** 最長的連續列入陳述式區塊，若不包括任何 `Net.Data` 變數參照或函數呼叫，其長度限制如下：

- OS/2 及 NT：64KB
- AIX：256KB
- OS/390：256KB
- OS/400：256KB

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16頁的『EXEC 區塊或陳述式』。

報告區塊

REPORT 區塊。為函數呼叫的輸出製作格式的指示。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54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49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

僅有 REXX、系統、以及 Perl `Net.Data` 所支援的語言環境，支援 EXEC 區塊。

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 */
/* 測試引數 */
num_args = arg();
say '有' num_args '個引數';
do i = 1 to num_args;
    say '引數' 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 = ''; /* 重設 a 為空字串 */
b = ''; /* 重設 b 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.pl" %}
    %}
%HTML(INPUT) {
    @today()
    %}
```

- Perl 程式 today.pl：

```
$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()
}%
```

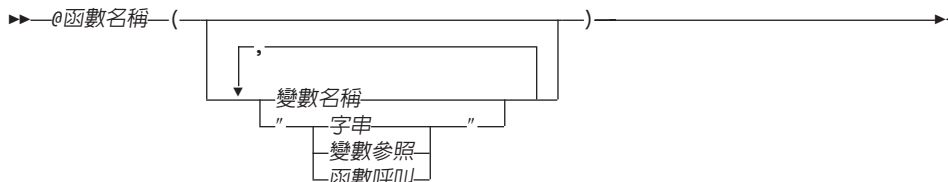
您可以比較本例與「範例 4」中 EXEC 陳述式的使用方式。範例 4 的「系統」語言環境不瞭解如何解譯 Perl 程式，但知道如何呼叫外部程式。EXEC 陳述式要它以外部程式的方式呼叫 perl 程式。實際的 Perl 語言陳述式，是由外部 Perl 程式解譯。範例 5 的 Perl 語言環境可以直接解譯 Perl 語言陳述式，所以沒有 EXEC 陳述式。

函數呼叫 (@)

目的

呼叫一個先前定義的 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

可為一或多個名稱，每一個額外名稱連接在句點之後(.)。有關語法的詳細資訊，請參閱第5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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

函數呼叫

呼叫一或多個先前定義的 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
- 函數呼叫無法包含在函數定義中用來定義 OUT 或 INOUT 參數的任何變數參照及函數呼叫。

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="NAME">Name
<INPUT NAME="queryVal" TYPE="CHECKBOX" VALUE="MAIL">E-mail
<INPUT NAME="queryVal" TYPE="CHECKBOX" VALUE="FAX">FAX
<INPUT TYPE="SUBMIT" VALUE="Submit request">
%}

%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)
%}

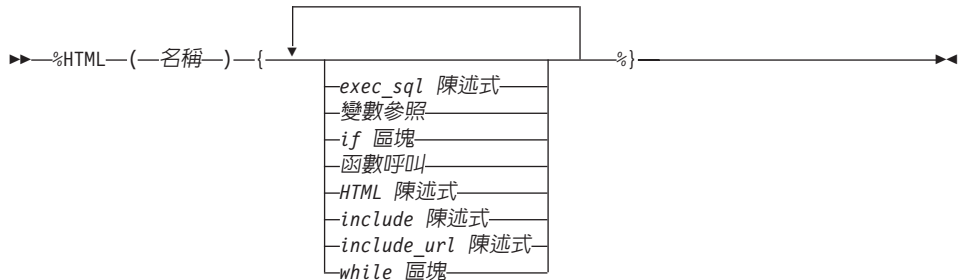
```

HTML 區塊

目的

包含任何將被從屬站的 Web 瀏覽器或任何可辨識 HTML 的工具所處理的 HTML 標籤或本文。HTML 區塊中也有大部份的 Net.Data 巨集語言陳述式，這些陳述式在執行時被求值及執行。Net.Data 尋找 Net.Data 巨集陳述式並執行它們。Net.Data 假設所有其他的本文皆為 HTML，並將它送給 Web 瀏覽器。

語法



值

%HTML

這個關鍵字指定從屬站瀏覽器顯示之 HTML 標籤及本文的所在區塊。

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，包括句點 (OS/390 除外)。

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269頁的『附錄 A. DB2 WWW 連線』或「DB2 全球資訊網版次 1」文件。

變數參照

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5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31頁的『IF 區塊』。

函數呼叫

呼叫一或多個先前定義的 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25頁的『函數呼叫 (@)』。

HTML 陳述式

包含任何英數字元，以及為從屬站瀏覽器製作格式的 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38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41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62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` 不支援非整數數字的小數比較，如，浮點數。

巢狀 IF 區塊： IF 區塊語法的規則由巨集檔中區塊的位置決定。如果 IF 區塊在宣告部份任何其他區塊之外的 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 區塊中的其他區塊之中建立巢狀，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。

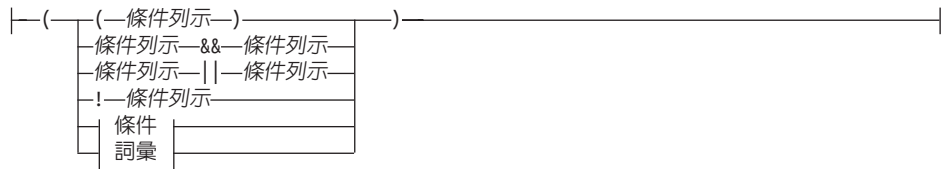
```
%IF 區塊
...
%HTML 區塊
...
%IF 區塊
```

請參閱在這個章節隨後所列的限制。

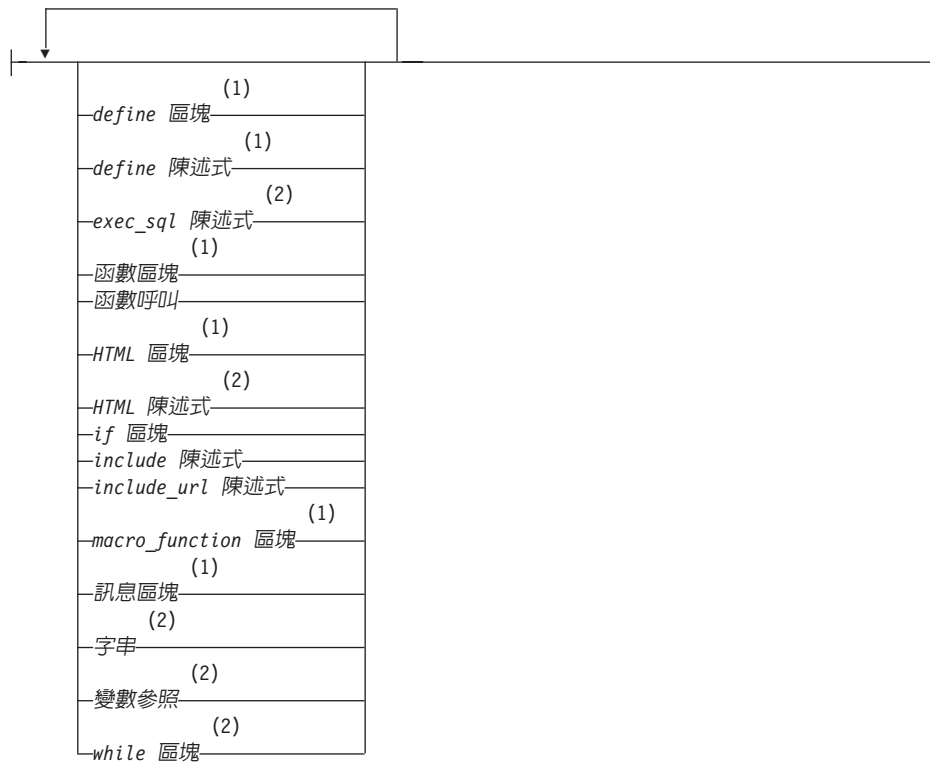
語法

► `%IF` | 條件列示 | 陳述式區塊 | `else_if` 規格 | `%ENDIF` ◀

條件列示



陳述式區塊



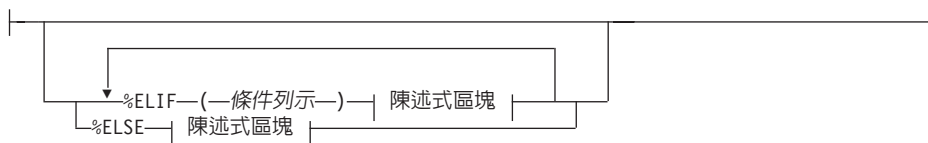
條件



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他區塊之外時，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10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269頁的『附錄A. 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18頁的『FUNCTION 區塊』。

函數呼叫

呼叫一或多個先前定義的 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25頁的『函數呼叫 (@)』。

HTML 區塊

包含任何英數字元，以及為從屬站瀏覽器製作格式的 HTML 標籤。

HTML 陳述式

包含任何英數字元，以及為從屬站瀏覽器製作格式的 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38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41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45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49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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5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62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 (-)，且沒有其他白色空間。

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他區塊之外
- **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他區塊之外時，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

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。

只有在處理巨集並插入參考檔案內容在巨集檔中 `INCLUDE` 陳述式的位置時，*Net.Data* 才會執行 `INCLUDE` 陳述式。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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5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。

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。

只有在處理巨集並插入參考檔案內容在巨集檔中 INCLUDE_URL 陳述式的位置時，Net.Data 才會執行 INCLUDE_URL 陳述式。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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

上下文

在這些上下文中可找到 INCLUDE_URL 陳述式：

- HTML 區塊
- REPORT 區塊
- ROW 區塊
- WHILE 區塊

- MACRO_FUNCTION 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

INCLUDE_URL 檔案有下列檔案大小限制：

- OS/2 與 Windows NT： 64 KB
- AIX： 256 KB
- OS/390： 256 KB

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他 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)"
```

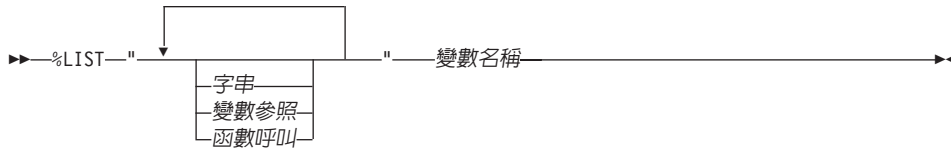
在這個範例中，巨集檔 hotpic.mac 被呼叫而 custno 被作為變數傳送。如果字串是以斜線開始，則 Net.Data 將從區域 Web 伺服器取回 INCLUDE 檔案。

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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

函數呼叫

呼叫一或多個先前定義的 `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25頁的『函數呼叫 (@)』。

變數名稱

可為一或多個名稱，每一個額外名稱連接在句點之後(.)。有關語法的詳細資訊，請參閱第5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

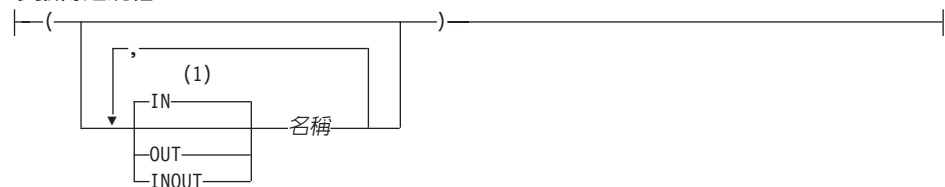
定義一個可以從 Net.Data 巨集呼叫的次常式。在 MACRO_FUNCTION 區塊中的可執行的陳述式必須是 Net.Data 巨集語言來源陳述式。

語法

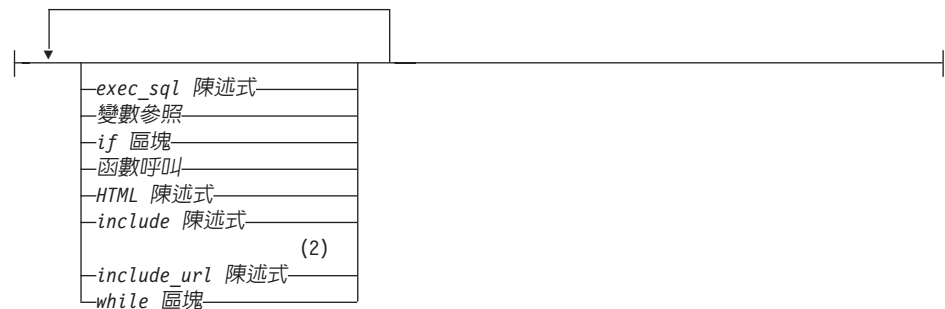
► %MACRO_FUNCTION—函數名稱—參數傳送規格—{—函數本體—



參數傳送規格



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. 只有 OS/400 才支援 MACRO_FUNCTION 區塊包含 REPORT 區塊。重複的報表區塊對呼叫語言環境的函數有效。

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。

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269頁的『附錄A. DB2 WWW 連線』或「DB2 全球資訊網版次 1」文件。

變數參照

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5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

呼叫一或多個先前定義的 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25頁的『函數呼叫 (@)』。

HTML 陳述式

包含任何英數字元，以及為從屬站瀏覽器製作格式的 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38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41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62頁的『WHILE 區塊』。

報告區塊

REPORT 區塊。為函數呼叫的輸出製作格式的指示。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54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

這個結構不適用於 OS/390 作業系統。

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
- 只有 OS/400 才支援
- 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

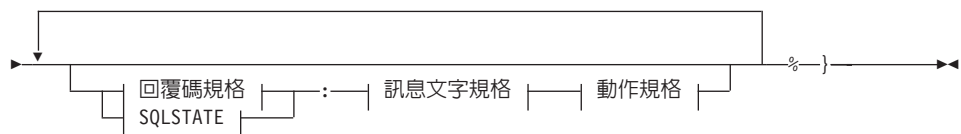
根據函數回覆碼，指定顯示的訊息及採行的建議動作。

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

►►MESSAGE—{—————→



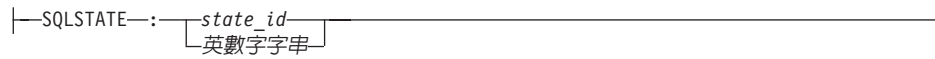
動作規格



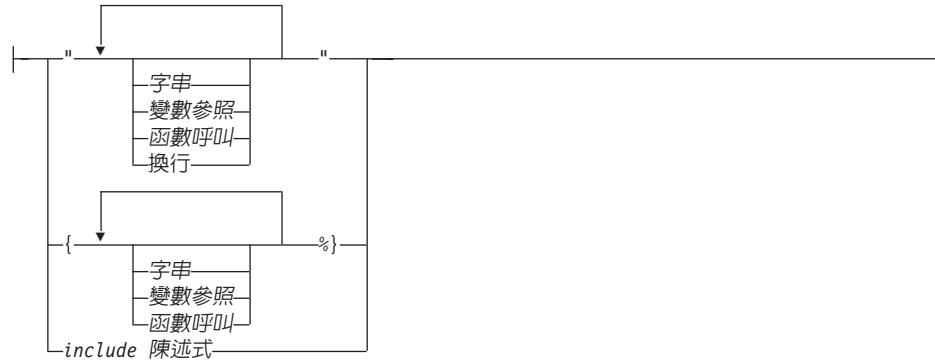
回覆碼規格



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。**限制：**在 OS/400 平台上不受支援。

state_id

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。

英數字字串

英文字或數字字串可以包含任何英文字母或數值字元的組合。它無法包含標點。

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

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

函數呼叫

呼叫一或多個先前定義的 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25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38頁的『INCLUDE 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

SQLSTATE 在 OS/400 平台上不受支援。

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">Return</a>  
    %}  
default: "<H3>無法繼續執行，因為發生錯誤 $(RETURN_CODE)</H3>"%}      : exit  
    %}
```

範例2：整體 MESSAGE 區塊

```
%{ global message block %}  
    %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

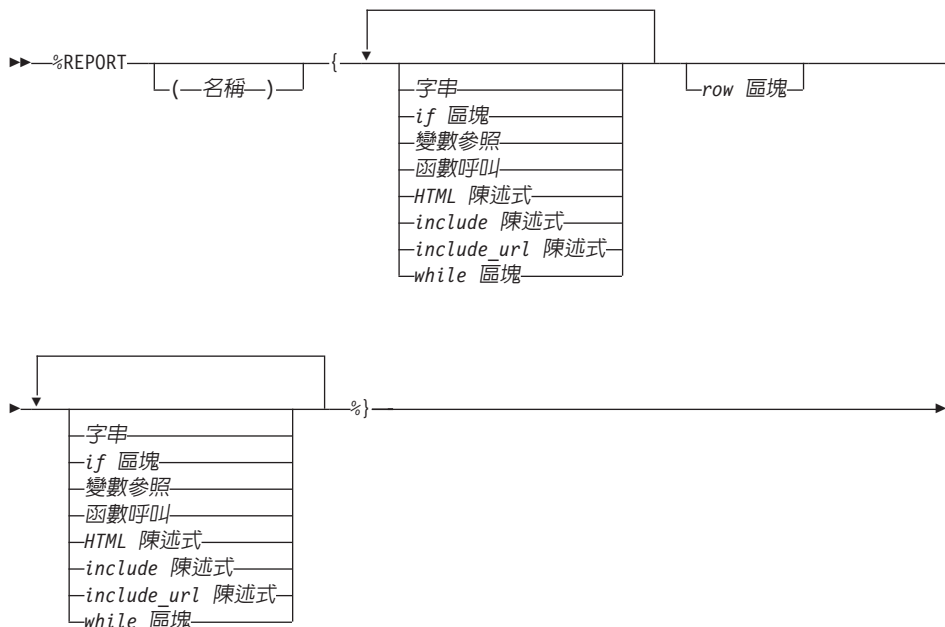
```
%message {  
  %include "rc1000.msg"  
  %include "rc2000.msg"  
  %include "defaults.msg"  
%}
```

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31頁的『IF 區塊』。

變數參照

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

函數呼叫

呼叫一或多個先前定義的 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25頁的『函數呼叫 (@)』。

HTML 陳述式

包含任何英數字元，以及為從屬站瀏覽器製作格式的 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38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41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57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62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
- MACRO_FUNCTION 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
- 在 **OS/390** 平台上：SQL 函數內無法呼叫 SQL 函數。
- HTML 陳述式
 - 字串
 - 變數參照

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>
%}
```

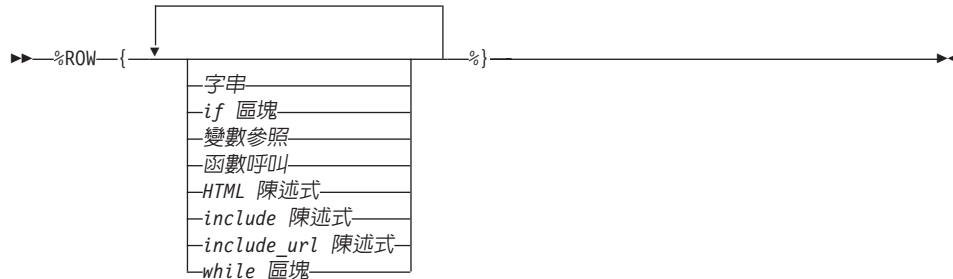
選取表格中的一個名稱，可呼叫 *name.mac* Net.Data 巨集的明細 HTML 區塊，將這二個值當作 URL 的一部份來傳送給它。您可以在本例的 *name.mac* 中使用這些值來尋找名稱的其它明細。

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31頁的『IF 區塊』。

變數參照

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5頁的『變數參照』。

函數呼叫

呼叫一或多個先前定義的 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25頁的『函數呼叫 (@)』。

HTML 陳述式

包含任何英數字元，以及為從屬站瀏覽器製作格式的 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38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41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62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

在 OS/390 平台上：SQL 函數內無法呼叫 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。它裡面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。表格陳述式只能用於 **DEFINE** 陳述式或區塊。

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

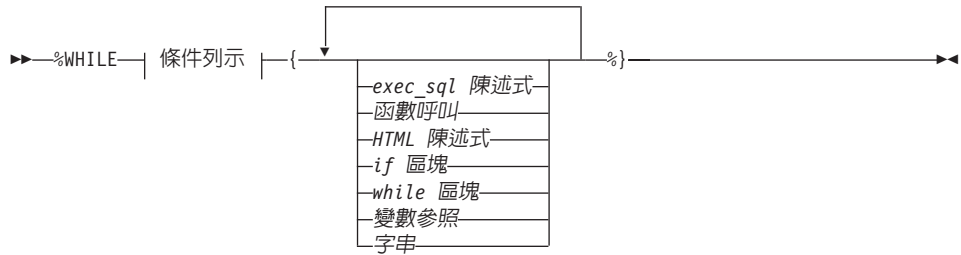
```
%DEFINE myTable3=%TABLE(ALL)
```

WHILE 區塊

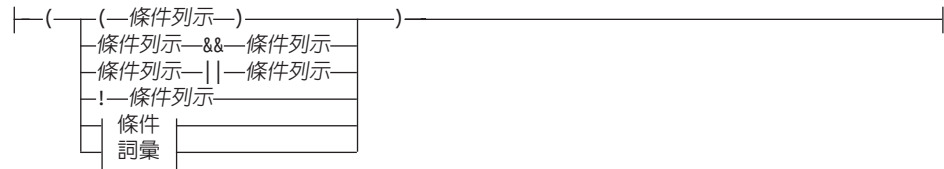
目的

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



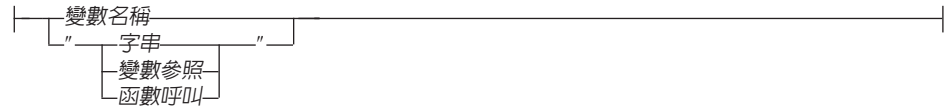
條件列示



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 (-)，且沒有其他白色空間。

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269頁的『附錄 A. DB2 WWW 連線』或「DB2 全球資訊網版次 1」文件。

函數呼叫

呼叫一或多個先前定義的 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25頁的『函數呼叫 (@)』。

HTML 陳述式

包含任何英數字元，以及為從屬站瀏覽器製作格式的 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31頁的『IF 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62頁的『WHILE 區塊』。

變數參照

傳回先前定義的變數值，並以 \$ 號及 () 號加以指定。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5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

可為一或多個名稱，每一個額外名稱連接在句點之後(.)。有關語法的詳細資訊，請參閱第5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

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68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68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69頁的『環境變數』
使用 ENVVAR 語言結構去參考環境變數。
- 第70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71頁的『隱藏變數』
隱藏 HTML 原始程式中的變數參照。
- 第72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73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，但其他變數則必需由 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74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，否則它們只有在 REPORT 區塊中才能被認得。
- 第84頁的『Net.Data 報告變數』
可幫助您使用函數自行設定報告。在參照這些變數之前，必須先定義它們。您可以在任何 Net.Data 巨集區塊中，定義或參照報告變數。
- 第92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111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，您無法在巨集檔中起始設定一個長度大於 256 KB 的變數或如此設定其為預設值。

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：

HP-UX	Hewlett Packard UNIX 作業系統
SCO	Santa Cruz UNIX 作業系統
SUN	Sun Solaris UNIX 作業系統
Win NT	Microsoft's 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。您可以在巨集檔中定義這些變數。

條件式變數

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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15頁的『*ENVVAR* 陳述式』。

執行變數

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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16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的程式：

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。若要使用隱藏變數，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)。

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="$(name)"> Name
<OPTION VALUE="$(addr)"> Address
.
.
.
</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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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43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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

表格變數中含有一個陣列的值以及相關的直欄名稱。每一個陣列中的元素就是一橫列。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。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60頁的『**TABLE** 陳述式』。

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。

限制： 不要在 DEFINE 區段中定義這些變數的值。

- 第75頁的『Nn』
- 第76頁的『NLIST』
- 第77頁的『NUM_COLUMNS』
- 第78頁的『NUM_ROWS』
- 第79頁的『ROW_NUM』
- 第80頁的『TOTAL_ROWS』
- 第81頁的『V_columnName』
- 第83頁的『Vn』
- 第82頁的『VLIST』

Nn

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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*。

這些變數均是預先定義的變數，所以無法修改這些值。請將這些變數當做變數參照，並指定您要參考的直欄號碼。

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169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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

| 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：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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

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： 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	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60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。

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：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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。這個變數有如一個計數器，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。

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 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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

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。

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/2、Windows NT 和 UNIX 作業系統上，這個變數只能在 REPORT 標底上被參考。

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102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：$(V3)
%}
</UL>
所顯示的名稱：$(NUM_ROWS)<BR>
所找到的名稱：$(TOTAL_ROWS)
%}
```

V_columnName

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

這些變數均是預先定義的變數，所以無法修改這些值。請將這些變數當做變數參照，並指定您要參考的直欄名稱。

值

V_columnName

表 1. V_columnName 值

值	說明
columnName	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{  
  
  NAME 直欄在 $(ROW_NUM) 橫列中為 $(V_NAME).<BR>  
%}  
%}  
%}
```

VLIST

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

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：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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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>Name</TD><TD>E-mail address</TD><TD>Location</TD>
%ROW{
<TR><TD>$(V1)</TD>
<TD><a href="mailto:$(V2)">$(V2)</a></TD>
<TD>$(V3)</TD>
%}
</TABLE>
找到了符合您所說明的 $(NUM_ROWS) 模式。
%}
```

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。您必須先定義這些變數，才能使用它們。

- 第85頁的『ALIGN』
- 第86頁的『DTW_DEFAULT_REPORT』
- 第87頁的『DTW_HTML_TABLE』
- 第88頁的『RPT_MAX_ROWS』
- 第89頁的『START_ROW_NUM』

ALIGN

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

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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

指定顯示在由函數 `REPORT` 區塊產生的表格中的列數

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

OS/400, Windows NT, OS/2 與 UNIX 的效能要訣： 將 `RPT_MAX_ROWS` 傳遞到資料庫語言環境將協助您限制從語言環境傳回的列數，因而可改善執行效能。若要將 `RPT_MAX_ROWS` 傳遞到語言環境（例如，`DTW_SQL`），請在您使用的資料庫語言環境的 `ENVIRONMENT` 陳述式中，併入此變數作為該陳述式的 `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。

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 <code>ALL</code> 。
號碼	一個正整數，用於顯示在由函數 <code>call</code> 區塊產生的表格中的列數。 如果 <code>FUNCTION</code> 區塊包含 <code>REPORT</code> 及 <code>ROW</code> 區塊，則這個數字所指定的是執行 <code>ROW</code> 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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X		X	X	X	X

目的

指定一個開始顯示 Net.Data 表格結果的列號。將這個變數與 RPT_MAX_ROWS 一起使用，可以將具有大型結果表格的查詢分成幾個小組，並使用「下一步」按鈕，來導覽結果表格。

OS/400, OS/2, Windows NT 與 UNIX 的效能要訣： 確定 START_ROW_NUM 指定為資料庫 LE 的 ENVIRONMENT 陳述式上的 IN 變數。不然，資料庫 LE 會傳回開始於第一列的結果設定。如果未在 ENVIRONMENT 陳述式上指定，Net.Data 須捨棄直到指定列的所有列，如此 LE 才不會傳回第一位置中的橫列。

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
號碼	<p>一個正整數，代表開始顯示報告的列號。</p> <p>如果 START_ROW_NUM 在起始設定檔案中的資料庫語言環境之 ENVIRONMENT 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 START_ROW_NUM 沒有被傳送到語言環境，則這個號碼所指定的是 Net.Data 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>
%}

```

Net.Data 語言環境變數

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。在參照這些變數之前，您可能必須先定義它們。

- 第93頁的『DATABASE』
- 第95頁的『DB_CASE』
- 第96頁的『DB2PLAN』
- 第97頁的『DB2SSID』
- 第98頁的『DTW_APPLET_ALTTEXT』
- 第99頁的『DTW_EDIT_CODES』
- 第100頁的『DTW_MBMODE』
- 第101頁的『DTW_SAVE_TABLE_IN』
- 第102頁的『DTW_SET_TOTAL_ROWS』
- 第103頁的『LOCATION』
- 第104頁的『LOGIN』
- 第105頁的『NULL_RPT_FIELD』
- 第106頁的『PASSWORD』
- 第107頁的『SHOWSQL』
- 第108頁的『SQL_STATE』
- 第109頁的『TRANSACTION_SCOPE』

DATABASE

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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 命令字元。

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	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。

必要條件： 在 Net.Data 起始設定檔案中的 DTW_SQL ENVIRONMENT 陳述式上，必需設有這個變數，但在巨集檔中則為選用性地設定。如果這個變數沒有被指定在 OS/390 起始設定檔案內或只在巨集內而不在起始設定檔案中時，一旦當巨集嘗試執行 SQL 函數，就會發生錯誤。

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

```
%DEFINE DB2PLAN="DTWGAV21"
```

DB2SSID

AIX	HP-UX	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。

必要條件： 這個變數必要在 Net.Data 起始設定檔案中被指定，或在巨集檔中選用性地被指定。如果這個變數沒有在 Net.Data for OS/390 起始設定檔案中指定，也沒有在巨集中定義，這時候，若巨集想要執行 SQL 函數就會發生錯誤。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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
<i>HTML_text_and_tags</i>	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	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_MBMODE

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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案中設定這個變數，但也可以在巨集檔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。

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 不處理被移轉到 OS/400 作業系統的巨集檔內所含的此變數。

值

`DTW_MBMODE="YES" | "NO"`

表 13. *DTW_MBMODE* 值

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

範例

範例 1： 置換 INI 檔案中的值

INI 檔：

```
DTW_MBMODE "NO"
...
ENVIRONMENT (DTW_DEFAULT) [d11] (IN DTW_MBMODE, OUT
RETURN_CODE )
```

巨集檔：

```
%DEFINE DTW_MBMODE = "YES"
```

DTW_SAVE_TABLE_IN

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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
<i>table_name_var</i>	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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

若要自動傳送變數，可將它作為一個 `IN` 變數併入資料庫語言環境陳述式中，該陳述式位於 `Net.Data` 起始設定檔案內。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。

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。 重要事項： 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> 無法在巨集檔中被參考。 <code>NO</code> 是預設值。

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。

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>傳回了 $(NUM_ROWS)。您的查詢限於 $(TOTAL_ROWS) 列之內。
%}
%}
```

LOCATION

AIX	HP-UX	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	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
null_char	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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

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。

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

範例 2：設定是否使用 HTML 套表輸入來顯示 SQL

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

SQL_STATE

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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 命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。

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

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 區塊中對其他資料庫所作的更新，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 巨集檔的資料庫更新不相影響。

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 命令順利完成，Net.Data 就會發出 COMMIT。
MULTIPLE	Net.Data 只會在 HTML 區塊中所有的 SQL 命令都順利完成之後，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112頁的『DTW_CURRENT_FILENAME』
- 第113頁的『DTW_CURRENT_LAST_MODIFIED』
- 第114頁的『DTW_DEFAULT_MESSAGE』
- 第115頁的『DTW_LOG_LEVEL』
- 第116頁的『DTW_MACRO_FILENAME』
- 第117頁的『DTW_MACRO_LAST_MODIFIED』
- 第118頁的『DTW_MP_PATH』
- 第119頁的『DTW_MP_VERSION』
- 第120頁的『DTW_PRINT_HEADER』
- 第121頁的『DTW_REMOVE_WS』
- 第122頁的『RETURN_CODE』

DTW_CURRENT_FILENAME

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

目的

現行輸入檔案的名稱與副檔名。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>這個檔案為 <I>\$(DTW_CURRENT_FILENAME)</I>，
且已在 \$(DTW_CURRENT_LAST_MODIFIED) 上更新。

DTW_CURRENT_LAST_MODIFIED

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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 巨集檔，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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				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 巨集檔中的任何部分使用 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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

必要條件： 您必需在 Net.Data 起始設定檔案中定義 `DTW_LOG_DIR` 來起始日誌；否則當您在巨集檔中設定 `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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

前次修改 Net.Data 巨集의 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_MP_PATH

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。其輸出格式如下：

Net.Data 版本 2.1

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

如果您正在使用 DTW_PRINT_HEADER 製作自己的表頭 (DTW_PRINT_HEADER="NO")，您必需設定 DTW_REMOVE_WS="NO"。

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 23. 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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。

如果您正在使用 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。您可以在巨集檔中定義這個值，將它置換。如果巨集檔中沒有定義 DTW_REMOVE_WS，則它會使用起始設定檔案中的值。

值

DTW_REMOVE_WS="YES" | "NO"

表 24. 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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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 巨集檔中的任何部分參照 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，且不適用於其他的種類。請參閱第124頁的『一般函數』。
- **算術函數**執行算術作業，請參閱第154頁的『算術函數』。
- **字串操作函數**修改字串與字元。請參閱第168頁的『字串函數』。
- **字組操作函數**修改字組或字組集。請參閱第185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196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225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248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261頁的『持續巨集函數』。

在下列說明中，函數參數是描述成字串、整數、浮點數與表格的類型。所有 Net.Data 變數都是字串類型，但是整數或浮點數這些項目是分別用來代表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。這些參數傳遞的規格是以下列關鍵字來指定：

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)。

- 其他以 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。在最後一個輸入參數設定之前，您可以指定所有輸入參數，或將其指定為空值（『』）以接受預設值。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，且不適用於其他的種類。下列函數都是一般目的函數：

- 第126頁的『DTW_ADDQUOTE』
- 第128頁的『DTW_CACHE_PAGE』
- 第132頁的『DTW_DATE』

- 第134頁的『DTW_EXIT』
- 第135頁的『DTW_GETCOOKIE』
- 第137頁的『DTW_GETENV』
- 第138頁的『DTW_GETINIDATA』
- 第139頁的『DTW_HTMLENCODER』
- 第141頁的『DTW_QHTMLENCODER』
- 第142頁的『DTW_SENDMAIL』
- 第146頁的『DTW_SETCOOKIE』
- 第149頁的『DTW_SETENV』
- 第150頁的『DTW_TIME』
- 第152頁的『DTW_URLESCSEQ』

DTW_ADDQUOTE

AIX	HP-UX	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。當字串包含單引號時，需要置換以使 SQL 陳述式可以正確的處理。

請考慮對所有的 SQL INPUT 陳述式使用這個函數。例如，如果您如同下列的範例一般，將 O'Brien 輸入為姓，則單引號將會造成一個錯誤：

```
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')
```

語法

@DTW_ADDQUOTE(stringIn, stringOut)

@DTW_rADDQUOTE(stringIn)

@DTW_mADDQUOTE(stringMult, stringMult2, ..., stringMultn)

值

表 25. 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 可以有多个輸入字串。
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">在輸入上：包含字串的變數。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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X							

目的

在巨集檔中函數的位置後開始快取所有 HTML 輸出。當被呼叫時，這個函數會嘗試從快取記憶體取回指定的頁面，並將它傳送到 Web 瀏覽器，就如同此頁面是由巨集所產生的。如果頁面找到而且未過期，Net.Data 就停止巨集的處理程序、跳出巨集檔、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。

決定 DTW_CACHE_PAGE 函數在巨集檔中的位置：

- 對於大部份快取應用程式而言，請將 DTW_CACHE_PAGE 指定在巨集的頂端以快取所有的 HTML 輸出。當有新報告區塊加入時，這個技術使巨集檔的維護變得更容易些。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。
- 對於更進階的快取應用，當您需要在處理過程中的一個特定點決定實施快取時，您可以將函數放置在 HTML 的輸出區段，而不是在巨集檔的開頭。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。

語法

@DTW_CACHE_PAGE(cacheid, url, age, status)

值

表 26. 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。

表 26. DTW_CACHE_PAGE 參數 (繼續)

參數	使用	說明
<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"> • ok：當巨集執行終止時，輸出頁面會被送到快取記憶體。 • new：頁面不在快取記憶體中。 • renew：頁面在快取記憶體中，但已過期。 • no_cache：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。 • inactive：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。 • busy：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。 • error：嘗試與快取記憶體通信時發生錯誤。

範例

範例 1：將 DTW_CACHE_PAGE 函數放置在巨集檔的開頭，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(){
    從客戶選取 *
%}

%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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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()

值

表 27. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X		X	X	X	X

目的

指定立即離開巨集。Net.Data 會確定，讓巨集到現在為止所建立的頁面傳送到瀏覽器。

執行效能要訣：當輸出已經產生時，爲了節省 Net.Data 處理整個檔案的時間，使用 DTW_EXIT 來停止巨集檔的處理程序。

重要事項！ 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() 將使得 Net.Data 在遇到此函數的呼叫時，可停止處理巨集檔案，這可避免您攫取在 DTW_EXIT() 函數已處理過後所發生的錯誤。

語法

```
@DTW_EXIT()
```

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X			X	X	X

目的

指定待讀取之 cookie 的名稱，並傳回 cookie 的值。

要訣：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。

語法

@DTW_GETCOOKIE(IN cookie_name, OUT cookie_value)

@DTW_rGETCOOKIE(IN cookie_name)

值

表 28. 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。

使用

如果找不到 cookie，則會傳回用法的注意事項 8000。cookie 可能因下列理由而找不到：

- cookie 從未被設定。
- cookie 已過期。
- cookie 沒有到期日，因此無法持續；接收 cookie 的 Web 瀏覽器已跳出或被中止。
- cookie 被設定安全選項，而現行 HTTP 要求是透過非安全通道傳送。
- 當 Net.Data 提出設定 cookie 要求時，Web 瀏覽器不接受 cookies，或不執行 JavaScript 程式。
- cookie 已被 Web 瀏覽器刪除。當 cookie 的數目超出瀏覽器的限制時會發生這個現象。說明於 Netscape 規格與當它出版時的限制為：
 - 300 個 cookies 總數
 - 一個 cookie 為 4 千位元組，其中名稱與數值共同組成 4 千位元組的限制。
 - 每個伺服器或領域可以有 20 cookie。(請注意，完整指定的主電腦與領域被視為不同的實體，每一個均可以有 20 cookie 的限制，而非總合)

伺服器不應預期從屬站能超出這些限制。當 300 cookie 限制或每個伺服器 20 cookie 的限制超出時，從屬站應先刪除最近最少使用的 cookie。當發現 cookie 大於 4 千位元組時，應調整 cookie，但只要小於 4 千位元組，名稱應保持完整。

請參閱 Netscape 的規格以取得『持續從屬站狀態 HTTP Cookies』之最新資訊，可由下列網址獲得：

http://search.netscape.com/newsref/std/cookie_spec.html

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，瀏覽器會要求使用者輸入姓名，並將它置於 HTML 記憶區段，如此將使用者名稱設定到如下的 cookie NDC_name：

```
%HTML(remember) {
    <html>
    <body>
    <H1>Net.Data 俱樂部</H1>
    @DTW_SETCOKIE("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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將所指定之環境變數的值傳回。您也可以使用 **ENVVAR** 來參考環境變數的值。有關詳細資訊，請參閱第15頁的『**ENVVAR** 陳述式』。

語法

```
@DTW_GETENV(envVarName, envVarValue)
```

```
@DTW_rGETENV(envVarName)
```

值

表 29. *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。

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/path"

範例 2：傳回 PATH 陳述式的值

```
@DTW_rGETENV(myPath)
```

- 輸入：myPath = "PATH"
- 傳回："/usr/path"

範例 3：傳回伺服器名稱的值

伺服器是 @DTW_rGETENV("SERVER_NAME")。

- 傳回："www.software.ibm.com"

DTW_GETINIDATA

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

目的

傳回所指定之架構變數的值。若找不到值的話將會傳回一個空字串。

限制：對於非 OS/400 作業系統而言，架構路徑變數（MACRO_PATH，EXEC_PATH，與 INCLUDE_PATH），以及 ENVIRONMENT 陳述式無法由這個呼叫取回。在 OS/400 作業系統上，這個限制只能應用到 ENVIRONMENT 陳述式。

語法

@DTW_GETINIDATA(iniVarName, iniVarValue)

@DTW_rGETINIDATA(iniVarName)

值

表 30. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

編碼字元將標準 HTML 十進位跳出碼 (escape code) 使用在大部份的字元上。您可以使用這個函數來對您不想要 Web 瀏覽器解譯為 HTML 的資料來進行編碼。 例如，藉由使用適當的跳出字元，您可以顯示小於 (<) 及大於 (>) 符號，這些經常保留作 HTML 標籤的符號。

在第二個範例中，下列在 HTML 中的字串於每個數字之間只顯現一個空白。

1 2 3

使用 DTW_HTMLENCODE 來確定所顯現的空白數目正確。

表31 顯示由 DTW_HTMLENCODE 函數所編碼的字元。

表 31. HTML 十進位跳出字元

字元	名稱	程式碼
SPACE	空格	
"	雙引號	"
#	號碼記號	#
%	百分比	%
&	& 記號	&
[左方括弧	(
]	右方括弧)
+	加	+
\	斜線	/
:	冒號	:
;	分號	;
<	小於	<
=	等於	=
>	大於	>
?	問號	?
@	At 記號	@
/	反斜線	\
^	Carat	^
{	左大括弧	{
	直線	|

表 31. HTML 十進位跳出字元 (繼續)

}	右大括弧	}
-	波型符號	~

語法

@DTW_HTMLENCODE(stringIn, stringOut)

@DTW_rHTMLENCODE(stringIn)

值

表 32. 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	一個包含已修改之輸入字串的變數，在該輸入字串中有某些特定的字元已被 HTML 跳出字元編碼所置換。

範例

範例 1：對空格字元編碼

@DTW_HTMLENCODE(string1,string2)

- 輸入：string1 = "Jim's dog"
- 傳回：string2 = "Jim's dog"

範例 2：對空格、小於符號與等號編碼

@DTW_rHTMLENCODE("X <= 10")

- 傳回："X <= 10"

DTW_QHTMLENCODE

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

目的

執行如同 @DTW_HTMLENCODE 的功能，但亦將單引號字元 (') 編碼為 '。DTW_QHTMLENCODE 所使用的 HTML 十進位跳出字元將顯現在第139頁的表31。

語法

@DTW_QHTMLENCODE(stringIn, stringOut)

@DTW_rQHTMLENCODE(stringIn)

值

表 33. 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> ，其中有某些特定的字元已被編碼的 HTML 跳出字元所置換。

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's dog"

範例 2：對單引號、空格與 & 符號編碼

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

- 傳回："John's & Jane's"

DTW_SENDMAIL

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

目的

動態建立與傳輸電子郵件（e-mail）訊息。

此函數與選用的架構變數 `DTW_SMTP_SERVER` 一起使用，指定用來傳輸電子郵件訊息的 SMTP 伺服器。這個變數的值可以是主電腦名稱或是 IP 位址。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。請參閱 *Net.Data 管理及程式設計手冊* 內的架構章以取得這些變數的詳細資訊。

國家語言議題：「標準簡易郵件轉送通信協定」(SMTP) 伺服器只接受 7 位元資料如美式 ASCII 字元。如果您的訊息有 8 位元字元，建議您指定擴充式簡易郵件轉送通信協定 (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。

疑難排解：下列列表說明 `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「擴充式簡易郵件轉送通信協定 (ESMTP)」，但指定的電子郵件卻包含非美式 ASCII 字元。

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)

值

表 34. 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">• 姓名 <使用者 @ 領域>• <使用者 @ 領域>• 使用者 @ 領域
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">• 姓名 <使用者 @ 領域>• <使用者 @ 領域>• 使用者 @ 領域
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。
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。
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。
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。有效的回覆格式為： <ul style="list-style-type: none">• 姓名 <使用者 @ 領域>• <使用者 @ 領域>• 使用者 @ 領域
字串	<i>Organization</i>	IN	包含寄件者組織名稱的變數或文字字串。

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 使用者 1 <ibmuser1@ibm.com>", "IBM 使用者  
2 <ibmuser2@ibm.com>, IBM 使用者 3 <ibmuser3@ibm.com>, IBM 使用者  
4 <ibmuser4@ibm.com>", "在 9:30 有一個會議。", "狀態會議", "IBM 使用者  
5 <ibmuser5@ibm.com>", "IBM 使用者  
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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X			X	X	X

目的

定義 cookie 的名稱、值與選項，例如到期日與安全需求。

要取回 cookie，請使用 DTW_GETCOOKIE() 函數。請參閱第135頁的『DTW_GETCOOKIE』以學習如何定義 cookie。

當安全需求還未定義時，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。

要訣：

- 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。
- 為了簡單因素，避免在 cookie 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"))
```

限制：

- 如果從屬站 Web 瀏覽器不支援 Java Script，瀏覽器便不設定 cookie。
- 因為 DTW_SETCOOKIE 會產生 Java Script，請勿在 <SCRIPT> 或 <NOSCRIPT> HTML 元素中呼叫 DTW_SETCOOKIE。

語法

```
@DTW_SETCOOKIE(IN cookie_name, IN cookie_value, IN advanced_options)
```

```
@DTW_SETCOOKIE(IN cookie_name, IN cookie_value)
```

值

表 35. 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。

表 35. 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。這些屬性為：</p> <p>到期 = 日期 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>領域 = 領域名稱 指定 cookie 的領域屬性，使用在領域屬性匹配。</p> <p>路徑 = 路徑 指定 cookie 是有效的領域中 URL 的子集。</p> <p>安全 指定 cookie 只能透過安全的通道傳輸到 HTTPS 伺服器。</p>

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")
```

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。若找不到上一個值的話將會傳回一個空字串。

語法

```
@DTW_SETENV(envVarName, envVarValue, prevValue)
```

```
@DTW_rSETENV(envVarName, envVarValue)
```

值

表 36. *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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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()

值

表 37. 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 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

以其跳出值來置換 URL 中不容許的字元，也稱作 URL 編碼的值。您必須使用這個函數來將任何列示在表38 的字元傳送到另一個巨集檔或 HTML 區塊。

表 38. URL 中不容許的字元

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
=	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

語法

@DTW_URLESCSEQ(stringIn, stringOut)

@DTW_rURLESCSEQ(stringIn)

值

表 39. *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。

範例

範例 1：以它們 URL 跳出碼來置換 *string1* 中的空格與&符號，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：將空格與&符號轉換成 URL 編碼的格式

```
@DTW_rURLESCSEQ("Guys & Dolls")
```

- 傳回："Guys%20%26%20Dolls"

範例 3：在 ROW 區塊中使用 *DTW_rURLESCSEQ*，並將空格與 at 符號轉換成 URL 編碼的格式

```
%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>
```

當應用程式使用者在名稱上按一下時，名稱和電子郵件位址會以編碼的值作為 *name* 與 *email* 變數，被傳送至 Net.Data 巨集 fullrpt.mac 的輸入區塊。

算術函數

這些函數可讓您作算術的計算。

UNIX、Windows NT 與 OS/2 的執行效能要訣：經由將 Net.Data 起始設定檔案或巨集檔中的 DTW_OPTIMIZE_MATH 架構值設定為 YES，您可以使算術函數達到最佳的執行效能。

- 當設定為 YES 時，Net.Data 使用 C 算術格式而使函數執行更快；但是，輸出格式則與無此變數者不同。小數點後的零都不顯示。
- 當 DTW_OPTIMIZE_MATH 設定為 NO，Net.Data 使用 REXX 算術格式。函數執行較慢，但其輸出格式則與前一版 Net.Data 所產生的輸出一樣。預設值是 NO。

請參閱 *Net.Data* 管理及程式設計手冊內的架構變數章節，以學習如何架構這些變數。

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 伺服器上的「語言環境」設定

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155頁的『DTW_ADD』
- 第156頁的『DTW_DIVIDE』
- 第158頁的『DTW_DIVREM』
- 第160頁的『DTW_FORMAT』
- 第163頁的『DTW_INTDIV』
- 第164頁的『DTW_MULTIPLY』
- 第165頁的『DTW_POWER』
- 第166頁的『DTW_SUBTRACT』

DTW_ADD

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

目的

加總兩個參數的值。

語法

@DTW_ADD(number1, number2, precision, result)

@DTW_ADD(number1, number2, result)

@DTW_rADD(number1, number2, precision)

@DTW_rADD(number1, number2)

值

表 40. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將第一個參數的值除以第二個參數的值。

語法

@DTW_DIVIDE(number1, number2, precision, result)

@DTW_DIVIDE(number1, number2, result)

@DTW_rDIVIDE(number1, number2, precision)

@DTW_rDIVIDE(number1, number2)

值

表 41. 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	包含將 <i>number1</i> 除以 <i>number2</i> 之結果的變數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將第一個參數除以第二個參數並傳回餘數。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。

語法

@DTW_DIVREM(number1, number2, precision, result)

@DTW_DIVREM(number1, number2, result)

@DTW_rDIVREM(number1, number2, precision)

@DTW_rDIVREM(number1, number2)

值

表 42. 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	包含將 <i>number1</i> 除以 <i>number2</i> 之餘數的變數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

自行設定數字的格式。如果只有指定 *number* 參數，則結果將會格式化成 @DTW_rADD(number,『0』) 已執行的樣子。如果指定了任何其他的選項，則會依據下列的規則來對數字進行格式化：

- *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。
- 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。
- 此外，*expp* 和 *expt* 參數會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。
- 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。
- 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。

語法

```
@DTW_FORMAT(number, before, after, expp, expt, precision, result)
@DTW_FORMAT(number, before, after, expp, expt, result)
@DTW_FORMAT(number, before, after, expp, result)
@DTW_FORMAT(number, before, after, result)
@DTW_FORMAT(number, before, result)
@DTW_FORMAT(number, result)
@DTW_rFORMAT(number, before, after, expp, expt, precision)
@DTW_rFORMAT(number, before, after, expp, expt)
@DTW_rFORMAT(number, before, after, expp)
@DTW_rFORMAT(number, before, after)
@DTW_rFORMAT(number, before)
@DTW_rFORMAT(number)
```

值

表 43. *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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將第一個參數除以第二個參數，並傳回結果的整數部分。

語法

@DTW_INTDIV(number1, number2, precision, result)

@DTW_INTDIV(number1, number2, result)

@DTW_rINTDIV(number1, number2, precision)

@DTW_rINTDIV(number1, number2)

值

表 44. 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> 除以 <i>number2</i> 所得商的整數部份的變數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

兩個參數相乘，並傳回結果。

語法

@DTW_MULTIPLY(number1, number2, precision, result)

@DTW_MULTIPLY(number1, number2, result)

@DTW_rMULTIPLY(number1, number2, precision)

@DTW_rMULTIPLY(number1, number2)

值

表 45. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將第二個參數作為第一個參數的次方，並傳回結果。

語法

@DTW_POWER(number1, number2, precision, result)

@DTW_POWER(number1, number2, result)

@DTW_rPOWER(number1, number2, precision)

@DTW_rPOWER(number1, number2)

值

表 46. 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> 的 <i>number2</i> 次方結果的變數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將第二個參數的值從第一個參數的值中減去，並傳回結果。

語法

@DTW_SUBTRACT(number1, number2, precision, result)

@DTW_SUBTRACT(number1, number2, result)

@DTW_rSUBTRACT(number1, number2, precision)

@DTW_rSUBTRACT(number1, number2)

值

表 47. 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> 差的變數。

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169頁的『DTW_ASSIGN』
- 第170頁的『DTW_CONCAT』
- 第171頁的『DTW_DELSTR』
- 第172頁的『DTW_INSERT』
- 第174頁的『DTW_LASTPOS』
- 第175頁的『DTW_LENGTH』
- 第176頁的『DTW_LOWERCASE』
- 第177頁的『DTW_POS』
- 第178頁的『DTW_REVERSE』
- 第179頁的『DTW_STRIP』
- 第180頁的『DTW_SUBSTR』
- 第182頁的『DTW_TRANSLATE』
- 第184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` 起始設定檔案中設定這個值；預設值是不支援。您可以在 `Net.Data` 巨集檔中，經由設定 `DTW_MBMODE` 變數來置換起始設定檔案中的值。請參閱 *Net.Data 管理及程式設計手冊* 中的架構變數部分，與第100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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

指定一個輸入變數給輸出變數。您亦可以使用這個函數在巨集中變更變數。

語法

```
@DTW_ASSIGN(stringOut, stringIn)
```

值

表 48. *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。

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_CONCAT

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

值

表 49. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將所指定字串中第 n 個字元起 $length$ 個子字串刪除。

語法

@DTW_DELSTR(stringIn, n, length, stringOut)

@DTW_DELSTR(stringIn, n, stringOut)

@DTW_rDELSTR(stringIn, n, length)

@DTW_rDELSTR(stringIn, n)

值

表 50. 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	n	IN	要開始刪除子字串的字元位置。如果 n 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。

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_INSERT

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

值

表 51. 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> 。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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)

值

表 52. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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)

值

表 53. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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)

值

表 54. 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">在輸入上：包含字串的變數。在輸出上：一個包含被轉換為小寫之輸入字串的變數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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)

值

表 55. 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。

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_REVERSE

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

目的

將輸入字串反轉。

語法

@DTW_REVERSE(stringIn, stringOut)

@DTW_rREVERSE(stringIn)

值

表 56. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

除去在輸入字串中的前導空白、尾隨之空白，或兩者。

語法

@DTW_STRIP(stringIn, option, stringOut)

@DTW_STRIP(stringIn, stringOut)

@DTW_rSTRIP(stringIn, option)

@DTW_rSTRIP(stringIn)

值

表 57. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將輸入字串的子字串，與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)

值

表 58. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

使用輸入和輸出轉換表格、*tableI* 和 *tableO* 轉換輸入字串中的字元。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。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。

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)

值

表 59. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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)

值

表 60. 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">在輸入上：包含字串的變數。在輸出上：一個包含被轉換為大寫之輸入字串的變數。

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 起始設定檔案中設定這個值；預設值是不支援。您可以在 Net.Data 巨集檔中，經由設定 DTW_MBMODE 變數來置換起始設定檔案中的值。請參閱 *Net.Data 管理及程式設計手冊* 中的架構變數部分，與第100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186頁的『DTW_DELWORD』
- 第188頁的『DTW_SUBWORD』
- 第190頁的『DTW_WORD』
- 第191頁的『DTW_WORDINDEX』
- 第192頁的『DTW_WORDLENGTH』
- 第193頁的『DTW_WORDPOS』
- 第195頁的『DTW_WORDS』

DTW_DELWORD

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

目的

傳回輸入字串的子字串。由字組 *n* 開始起 *length* 個字組會被刪除。

語法

@DTW_DELWORD(stringIn, n, length, stringOut)

@DTW_DELWORD(stringIn, n, stringOut)

@DTW_rDELWORD(stringIn, n, length)

@DTW_rDELWORD(stringIn, n)

值

表 61. 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	<i>n</i>	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 <i>n</i> 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

傳回輸入字串的子字串。子字串由字組 *n* 開始，一直到 *length* 所指定的字組數目。

語法

@DTW_SUBWORD(stringIn, n, length, stringOut)

@DTW_SUBWORD(stringIn, n, stringOut)

@DTW_rSUBWORD(stringIn, n, length)

@DTW_rSUBWORD(stringIn, n)

值

表 62. 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	<i>n</i>	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 <i>n</i> 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 <i>n</i> 到 <i>stringIn</i> 尾端的字組。
字串	<i>stringOut</i>	OUT	一個包含由 <i>n</i> 和 <i>length</i> 所指定之 <i>stringIn</i> 子字串的變數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

從所指定之輸入字串的位置中傳回一個單一字組。

語法

@DTW_WORD(stringIn, n, stringOut)

@DTW_rWORD(stringIn, n)

值

表 63. 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	<i>n</i>	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 <i>n</i> 之字組的變數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

傳回輸入字串中第 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)

值

表 64. 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	<i>n</i>	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

傳回輸入字串中第 n 個字組的長度。

語法

@DTW_WORDLENGTH(stringIn, n, stringOut)

@DTW_rWORDLENGTH(stringIn, n)

值

表 65. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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。多重空白將以一個單一空白來看待，以供比較。對比較而言，是會區分大小寫的。

語法

@DTW_WORDPOS(stringIn1, stringIn2, n, stringOut)

@DTW_WORDPOS(stringIn1, stringIn2, stringOut)

@DTW_rWORDPOS(stringIn1, stringIn2, n)

@DTW_rWORDPOS(stringIn1, stringIn2)

值

表 66. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
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)
```

值

表 67. *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。

範例

範例 1：

```
@DTW_WORDS("Now is the time", result)
```

- 傳回：

```
result = "4"
```

範例 2：

```
@DTW_rWORDS(" ")
```

- 傳回："0"

表格函數

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197頁的『DTW_TB_APPENDROW』
- 第198頁的『DTW_TB_COLS』
- 第199頁的『DTW_TB_DELETEROW』
- 第200頁的『DTW_TB_DLIST』
- 第202頁的『DTW_TB_DUMP』
- 第203頁的『DTW_TB_DUMPV』
- 第204頁的『DTW_TB_GETN』
- 第205頁的『DTW_TB_GETV』
- 第206頁的『DTW_TB_HTMLENCODE』
- 第207頁的『DTW_TB_INPUT_CHECKBOX』
- 第208頁的『DTW_TB_INPUT_RADIO』
- 第209頁的『DTW_TB_INPUT_TEXT』
- 第211頁的『DTW_TB_INSERTCOL』
- 第212頁的『DTW_TB_INSERTROW』
- 第213頁的『DTW_TB_LIST』
- 第215頁的『DTW_TB_MAXROWS』
- 第216頁的『DTW_TB_QUERYCOLNONJ』
- 第217頁的『DTW_TB_ROWS』
- 第218頁的『DTW_TB_SELECT』
- 第219頁的『DTW_TB_SETCOLS』
- 第220頁的『DTW_TB_SETN』
- 第221頁的『DTW_TB_SETV』
- 第222頁的『DTW_TB_TABLE』
- 第224頁的『DTW_TB_TEXTAREA』

DTW_TB_APPENDROW

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

目的

將一列或多列新增到表格尾端。在橫列附加到表格之後，以 DTW_TB_SETV() 函數指定新列的表格值。

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。

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。

語法

```
@DTW_TB_APPENDROW(table, rows)
```

值

表 68. DTW_TB_APPENDROW 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	附加橫列的巨集表格變數。
整數	<i>rows</i>	IN	附加至 <i>table</i> 中的橫列數。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X		X	X	X	X

目的

傳回表格中現行直欄數目。

語法

```
@DTW_TB_COLS(table, cols)
```

```
@DTW_TB_rCOLS(table)
```

值

表 69. *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。

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()
...
@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()
...
<P>我的表格包含 @DTW_TB_rCOLS(myTable) 個直欄。
```

DTW_TB_DELETEROW

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

目的

刪除開始於 `start_row` 的橫列的一列或多列。

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。

語法

```
@DTW_TB_DELETEROW(table, start_row, rows)
```

值

表 70. *DTW_TB_DELETEROW* 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

從巨集表格中傳回 HTML 定義列示。

語法

@DTW_TB_DLIST(table, term, def, termstyle, defstyle, link, link_u, image, image_u)

值

表 71. 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）。
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。

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_DUMP

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

目的

傳回巨集表格變數的內容。每個表格的列都被列示在不同的行上。整個表格是由 `<PRE></PRE>` 標籤所包括。

語法

@DTW_TB_DUMP(table)

值

表 72. DTW_TB_DUMP 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	指定要顯示之巨集表格變數的符號。

範例

範例 1：

@DTW_TB_DUMP(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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

傳回巨集表格變數的內容。每一表格值都位於不同的行上。整個表格是由 `<PRE></PRE>` 標籤所含括。

語法

@DTW_TB_DUMPV(table)

值

表 73. DTW_TB_DUMPV 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	指定要顯示之巨集表格變數的符號。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X		X	X	X	X

目的

取回 *col* 中指定之直欄號碼的欄位標題。

呼叫 DTW_TB_GETN() 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。

語法

@DTW_TB_GETN(table, col, name)

@DTW_TB_rGETN(table, col)

值

表 74. 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。

範例

範例 1：取回直欄 4 的直欄名稱

```
%DEFINE myTable = %TABLE
%DEFINE name = ""
...
@FillTable()
...
@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()
...
<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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X		X	X	X	X

目的

取回 *row* 和 *col* 所指定的橫列與直欄號碼的表格值。

呼叫 DTW_TB_GETV() 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。

語法

```
@DTW_TB_GETV(table, row, col, value)
```

```
@DTW_TB_rGETV(table, row, col)
```

值

表 75. 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。

範例

範例 1：取回第 6 列、第 3 欄的表格值

```
%DEFINE myTable = %TABLE
%DEFINE value = ""
...
@FillTable()
...
@DTW_TB_GETV(myTable, "6", "3", value)
```

範例 2：取回橫列 1，直欄 1 的值

```
%DEFINE myTable = %TABLE
...
@FillTable()
...
<P>橫列 1，直欄 1 的表格值是 @DTW_TB_rGETV(myTable, "1", "1")。
```

DTW_TB_HTMLENCODE

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

目的

傳回具有這些 HTML 字元編碼的輸入巨集表格：

名稱	字元	程式碼
& 記號	&	&
雙引號	"	"
大於	>	>
小於	<	<

語法

```
@DTW_TB_HTMLENCODE(table, collist)
```

值

表 76. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

從巨集表格變數中傳回一個或多個 HTML 核對框輸入標籤。

語法

@DTW_TB_INPUT_CHECKBOX(table, prompt, namecol, valuecol, rows, checkedrows)

值

表 77. 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。
字串	<i>valuecol</i>	IN	在 <i>table</i> 中的直欄號碼或一個包含輸入欄位值的字串。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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

從巨集表格變數中傳回一個或多個 HTML 圓鈕輸入標籤。

語法

@DTW_TB_INPUT_RADIO(table, prompt, namecol, valuecol, rows, checkedrows)

值

表 78. 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。
字串	<i>valuecol</i>	IN	在 <i>table</i> 中的直欄號碼或一個包含輸入欄位值的字串。
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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

傳回 Net.Data 表格中所指定列的 HTML <INPUT> 標籤。例如，Net.Data 可以用 VALUE、SIZE 與 MAXLENGTH 屬性來產生 HTML <INPUT> 標籤：

```
<INPUT TYPE="TEXT" NAME="someName" VALUE="someValue" SIZE="20" MAXLENGTH="40">
```

語法

```
@DTW_TB_INPUT_TEXT(table, prompt, namecol, valuecol, size, maxlen, rows)
```

值

表 79. 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	在 <i>table</i> 中的直欄號碼或包含輸入欄位名稱的字串。
字串	<i>valuecol</i>	IN	在 <i>table</i> 中的欄號或一個包含為預設輸入欄位值的字串，其為 INPUT 標籤上 VALUE 屬性所指定的。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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				X			

目的

在 *after_col* 指定的直欄後，開始插入一欄或多欄。

語法

```
@DTW_TB_INSERTCOL(table, after_col, cols)
```

值

表 80. *DTW_TB_INSERTCOL* 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				X			

目的

在 *after_row* 中指定的橫列後，開始插入一系列或多列。

呼叫 DTW_TB_INSERTROW() 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。

語法

```
@DTW_TB_INSERTROW(table, after_row, rows)
```

值

表 81. DTW_TB_INSERTROW 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

傳回 HTML 列示。

語法

```
@DTW_TB_LIST(table, listtype, listitem, itemstyle, link_u, image_u)
```

值

表 82. 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 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。

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_MAXROWS

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

目的

傳回表格中容許的最大橫列數目。

語法

@DTW_TB_MAXROWS(table, maxrows)

@DTW_TB_rMAXROWS(table)

值

表 83. DTW_TB_MAXROWS 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	將傳回最大橫列數目的巨集表格變數。
整數	<i>maxrows</i>	OUT	含有 <i>table</i> 中容許的最大橫列數目的變數。定義表格時，若沒有定義橫列數目限制，將傳回 0。

範例

範例 1：取回表格中容許的最大橫列數目並傳回該值

```
%DEFINE myTable = %TABLE
%DEFINE maxrows = ""

@DTW_TB_MAXROWS(myTable, maxrows)
%IF (maxrows != "0")
    The maximum number of rows allowed is $(maxrows).
%ELSE
    There is no limit on the number of rows allowed.
%ENDIF
```

範例 2：取回並顯示最大橫列數目的值

```
%DEFINE myTable = %TABLE

%IF (@DTW_TB_rMAXROWS(myTable) != "0")
    The maximum number of rows allowed is @DTW_TB_rMAXROWS(myTable).
%ELSE
    There is no limit on the number of rows allowed.
%ENDIF
```

DTW_TB_QUERYCOLNONJ

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

目的

傳回與欄位標題有關聯的直欄號碼。若表格中不存在欄位標題，將傳回 0。

語法

@DTW_TB_QUERYCOLNONJ(table, name, col)

@DTW_TB_rQUERYCOLNONJ(table, name)

值

表 84. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X		X	X	X	X

目的

傳回表格中現行橫列數目。

語法

@DTW_TB_ROWS(table, rows)

@DTW_TB_rROWS(table)

值

表 85. 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。

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()
...
@DTW_TB_ROWS(myTable, rows)
```

範例 2：取回並顯示現行列數的值

```
%DEFINE myTable = %TABLE
...
@FillTable()
...
<P>橫列 1，直欄 1 的表格值是 @DTW_TB_rROWS(myTable, "1", "1")。
```

DTW_TB_SELECT

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

目的

傳回一個 HTML SELECT 功能表。

語法

@DTW_TB_SELECT(table, name, optioncol, size, multiple, rows, selectrows)

值

表 86. 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>selectedrows</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>rows</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。

範例

使用多重選擇來產生一個 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>
```


DTW_TB_SETCOLS

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

目的

設定表格中的直欄數目。以 DTW_TB_SETN() 函數指定欄位標題。一次僅能對一個表格使用 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。

語法

@DTW_TB_SETCOLS(table, cols)

值

表 87. DTW_TB_SETCOLS 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				X			

目的

指定欄位標題的名稱。若要刪除欄位標題，請將欄位標題值指定為 NULL。

在呼叫 `DTW_TB_SETN()` 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。

語法

```
@DTW_TB_SETN(table, name, col)
```

值

表 88. *DTW_TB_SETN* 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
				X			

目的

指定表格中的一個值。若要刪除表格值，請將該值指定為 NULL。

在呼叫 `DTW_TB_SETV()` 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。

語法

```
@DTW_TB_SETV(table, value, row, col)
```

值

表 89. *DTW_TB_SETV* 參數

資料類型	參數	使用	說明
表格	<i>table</i>	IN	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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

從巨集表格變數中傳回 HTML 表格。

語法

@DTW_TB_TABLE(table, options, collist, cellstyle, link_u, image_u, url_text, url_style)

值

表 90. 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">• BORDER• CELSPACING• WIDTH
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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

從巨集表格變數中傳回 HTML TEXTAREA 標籤。

語法

@DTW_TB_TEXTAREA(table, name, numrows, numcols, valuecol, rows)

值

表 91. 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。

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230頁的『DTWF_APPEND』
- 第232頁的『DTWF_CLOSE』
- 第233頁的『DTWF_DELETE』
- 第233頁的『DTWF_DELETE』
- 第237頁的『DTWF_OPEN』
- 第239頁的『DTWF_READ』
- 第241頁的『DTWF_REMOVE』
- 第242頁的『DTWF_SEARCH』
- 第244頁的『DTWF_UPDATE』
- 第246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228頁的『純本文檔介面區隔字元』
- 第228頁的『鎖定檔案』

存取純本文檔資料來源

您可以在 `Net.Data` 起始設定檔案中使用 `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：

```
@DTWF_READ("/macros/myfile.txt", ...)
```

下列各節討論：

- 第226頁的『`Net.Data` 如何決定純本文檔位置』

- 第227頁的『純本文檔架構規則』
- 第227頁的『安全建議』
- 第227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 起始設定檔案中搜尋 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

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。如果在任一目錄中均找不到檔案，Net.Data 將搜尋現行工作目錄來找出檔案。

如果指定了檔名 */mydir/myfile.txt*，Net.Data 將產生一個錯誤，因為目錄 */mydir* 不符合 FFI_PATH 中所設定的容許目錄之一。

判斷現行目錄：

當 FFI 函數中指定的檔案沒有目錄路徑，且在 FFI_PATH 指定的任一目錄中找不到檔案時，Net.Data 將使用現行目錄。

Net.Data 的現行目錄視您的 Web 伺服器的架構而定：

- 如果您正在使用 CGI，現行目錄即是 Net.Data 正從其中執行的目錄，通常為 *\www\cgi-bin*。
- 如果您正在使用 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_CREATE 函數來建立純本文檔時，請確定您指定的目錄路徑是在 FFI_PATH 中，或是確定知道哪一個目錄是現行目錄。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 起始設定檔案中的 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 (")。
- 與巨集檔案中的 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 起始設定檔案中的 FFI_PATH 陳述式存取的檔案。FFI 僅會搜尋陳述式中所列的路徑，所以其他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 或其他匯出作業，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 起始設定檔案的路徑新增到 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

爲了增進執行效能，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。爲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 (`ASCIITEXT`)

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案中的記錄，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`，作爲 `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。

若要在區隔字元中使用空值字元，請將 `DELIMITER` 參數指定爲反斜線及有雙引號的零 (`『/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，作爲區隔字元，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爲 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。這些函數容許您保留檔案供 `Net.Data` 處理使用，因爲它需要存取檔案，以便沒有其他處理可以讀取或更新檔案。

若要鎖定檔案，請使用 `DTWF_OPEN` 函數。此函數確定其他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。此函數會從 `Net.Data` 處理的控制中釋放檔案，以容許其他處理可讀取或更新檔案。

DTWF_APPEND

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

目的

將表格的變數的目次寫入檔案尾端。

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。

語法

@DTWF_APPEND(filename, transform, delimiter, table, retry, rows)

@DTWF_APPEND(filename, transform, delimiter, table, retry)

@DTWF_APPEND(filename, transform, delimiter, table)

值

表 92. 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">• 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。• 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。 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	一個從中讀取記錄的表格變數。 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。
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。

表 92. DTWF_APPEND 參數 (繼續)

資料類型	參數	使用	說明
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。

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, "ASCII TEXT", " ;", myTable)
```

範例 3：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
}%
@DTWF_APPEND(myFile, "ASCII TEXT", " ;", myTable, "0", "10")
```

DTWF_CLOSE

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

值

表 93. 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。

範例

範例 1：

```
@DTWF_CLOSE(myFile, "5")
```

DTWF_DELETE

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

目的

從檔案中刪除記錄。（並不刪除空檔案。）

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)

值

表 94. 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">• 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。• 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。 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將記錄插入檔案。

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)

值

表 95. 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">• 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。• 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。 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。 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。
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。

表 95. DTWF_INSERT 參數 (繼續)

資料類型	參數	使用	說明
整數	<i>startrow</i>	INOUT	開始插入之列的號碼。指定 1 表示從第一列開始插入。如果這個值大於在檔案中的列號碼，則該值會被變更爲最後一個記錄並以錯誤傳回。預設值是從 1 開始。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

明確地開啓一個檔案。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。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。

執行效能要訣：使用 DTWF_OPEN 來減少檔案開啓的次數。

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。

語法

@DTWF_OPEN(filename, mode, retry)

@DTWF_OPEN(filename, mode)

值

表 96. 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">• r - 開啓舊有的檔案以供讀取。• 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。）• 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。• r+ - 開啓一個舊有的檔案以供讀取和寫入。• 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。）• 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。
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。

範例

範例 1：

```
%DEFINE {  
    myFile = "c:/private/myfile"  
    myMode = "r+"  
}%  
@DTWF_OPEN(myFile, myMode, "1000")
```

DTWF_READ

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

目的

從檔案中將記錄讀取至表格變數。

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)

值

表 97. 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">• 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。• 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。 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。 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。
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。

表 97. DTWF_READ 參數 (繼續)

資料類型	參數	使用	說明
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。

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, "ASCIIITEXT", ";", myTable)
@DTW_TB_TABLE(myTable, "BORDER", "")
```

DTWF_REMOVE

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

目的

刪除整個檔案。

語法

```
@DTWF_REMOVE(filename, retry)
```

```
@DTWF_REMOVE(filename)
```

值

表 98. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將字串搜尋的結果傳回給表格變數。

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)

值

表 99. 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">• 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。• 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。 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">• 找到有相符者的列。• 找到有相符者的欄。• 來自檔案的相符欄。 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。

表 99. DTWF_SEARCH 參數 (繼續)

資料類型	參數	使用	說明
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 <table>中之最大的列數。預設值是讀取所有的列，或直到<table>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。</table></table>
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。

使用

- 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。
- 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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

從表格變數中更新檔案的記錄。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。

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)

值

表 100. 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">• 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。• 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。 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。 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。
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。

表 100. DTWF_UPDATE 參數 (繼續)

資料類型	參數	使用	說明
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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X	X	X	X	X	X

目的

將表格變數的目次寫入至檔案中。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。

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)

值

表 101. 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">• 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。• 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。 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	使用來將列匯出至檔案的表格變數。 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。
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。

表 101. DTWF_WRITE 參數 (繼續)

資料類型	參數	使用	說明
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。

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, "ASCII TEXT", ";", myTable, "5000")
```

範例 3：

```
%DEFINE {
    myFile = "c:/private/myfile"
    myTable = %TABLE
}%
@DTWF_WRITE(myFile, "ASCII TEXT", ";", 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249頁的『DTWR_ADDENTRY』
- 第250頁的『DTWR_CLEARREG』
- 第251頁的『DTWR_CLOSEREG』
- 第252頁的『DTWR_CREATEREG』
- 第253頁的『DTWR_DELENTY』
- 第254頁的『DTWR_DELREG』
- 第255頁的『DTWR_LISTREG』
- 第256頁的『DTWR_LISTSUB』
- 第257頁的『DTWR_OPENREG』
- 第258頁的『DTWR_RTVENTRY』
- 第260頁的『DTWR_UPDATEENTRY』

限制：當使用 OS/2 時，不要在 `registry`、`registryVariable` 與 `registryData` 參數中使用星號 (*)。

DTWR_ADDENTRY

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

值

表 102. 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。

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	OS/2	OS/390	OS/400	SCO	SUN	Win NT
X	X	X		X	X	X	X

目的

從 Web 登記清除登錄。

語法

@DTWR_CLEARREG(registry)

值

表 103. DTWR_CLEARREG 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	所要清除的登記名稱。

範例

範例 1：

@DTWR_CLEARREG("Myregistry")

DTWR_CLOSEREG

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

目的

關閉 Web 登記

語法

```
@DTWR_CLOSEREG(registry)
```

值

表 104. 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。 限制： 在 Web 登記名稱中，不要使用特殊字元，例如星號 (*) 與反斜線 (\)。

範例

範例 1：關閉登記

```
@DTWR_CLOSEREG("/qsys.lib/mylib.lib/myreg.file")
```

DTWR_CREATEREG

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

目的

建立一個新的 Web 登記。

Re

語法

@DTWR_CREATEREG(registry, security)

@DTWR_CREATEREG(registry)

值

表 105. DTWR_CREATEREG 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	所要建立的登記名稱。 限制： 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	<i>security</i>	IN	要建立 <i>registry</i> 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。

範例

範例 1：

```
@DTWR_CREATEREG("myRegistry")
```

範例 2：

```
@DTWR_CREATEREG("URLLIST", "*RWX, *RWX, *R")
```

DTWR_DELENTY

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

值

表 106. 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。

範例

範例 1：

```
@DTWR_DELENTY("Myregistry", "Jones")
```

範例 2：

```
@DTWR_DELENTY("URLLIST", "SMITH", "WORK_URL")
```

DTWR_DELREG

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

目的

刪除 Web 登記

語法

@DTWR_DELREG(registry)

值

表 107. DTWR_DELREG 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	要刪除登記的名稱。

範例

範例 1：

@DTWR_DELREG("Myregistry")

DTWR_LISTREG

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

目的

列示整個 Web 登記。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他登錄的表格列不足，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」表頭將由「Web 登記」語言環境設定為 "REGISTRY_VARIABLE" 和 "REGISTRY_DATA"。

語法

```
@DTWR_LISTREG(registry, registryTable)
```

值

表 108. 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。
字串	registryTable	OUT	放置登記之表格變數的名稱。

範例

範例 1：

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

```
@DTWR_LISTREG("URLLIST", RegistryTable)
```

DTWR_LISTSUB

AIX	HP-UX	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。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他登錄的表格列不足，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"。

這個函數只有在「Windows95 系統登記」相容的作業系統上才有效。

語法

```
@DTWR_LISTSUB(registry, registryTable)
```

值

表 109. DTWR_LISTSUB 參數

資料類型	參數	使用	說明
字串	<i>registry</i>	IN	所要列示的登記名稱。
字串	<i>registryTable</i>	OUT	放置登記之表格變數的名稱。

範例

範例 1：

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

```
@DTWR_LISTSUB("URLLIST", RegistryTable)
```

DTWR_OPENREG

AIX	HP-UX	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)

值

表 110. 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： Y 在確定控制下開啓登記。 N 不在確定控制下開啓登記。 預設值爲 N

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_RTVENTRY

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

目的

從 Web 登記中取回 registryData 字串。

語法

@DTWR_RTVENTRY(registry, registryVariable, registryData, index)

@DTWR_RTVENTRY(registry, registryVariable, registryData)

@DTWR_rRTVENTRY(registry, registryVariable, index)

@DTWR_rRTVENTRY(registry, registryVariable)

值

表 111. DTWR_RTVENTRY 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 registryData 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。

範例

範例 1：

```
%DEFINE RegistryData = ""
@DTWR_RTVENTRY("Myregistry", "Jones", RegistryData)
```

範例 2：

```
@DTWR_RTVENTRY("URLLIST", "SMITH", RegistryData, "WORK_URL")
```

範例 3：

```
@DTWR_rRTVENTRY("Myregistry", "Jones")
```

範例 4：


```
@DTWR_rRTENTRY("URLLIST", "SMITH", "WORK_URL")
```

DTWR_UPDATEENTRY

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

目的

將所指定之登記的舊有 *registryData* 字串值置換成由呼叫者所指定的新值。無法變更 *registryVariable* 字串。

語法

@DTWR_UPDATEENTRY(registry, registryVariable, newData, index)

@DTWR_UPDATEENTRY(registry, registryVariable, newData)

值

表 112. 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。

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262頁的『`DTW_ACCEPT`』
- 第264頁的『`DTW_COMMIT`』
- 第265頁的『`DTW_ROLLBACK`』
- 第266頁的『`DTW_RTVHANDLE`』
- 第267頁的『`DTW_STATIC`』
- 第268頁的『`DTW_TERMINATE`』

DTW_ACCEPT

AIX	HP-UX	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`。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。

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。

當巨集不在持續狀態時，若呼叫這個函數，將發生 Net.Data 錯誤。

要訣：含有異動 `handle` 的 URL 可編寫為格式按鈕上的動作，或呈現給瀏覽器的頁面的超本文鏈結。

語法

```
@DTW_ACCEPT(handle, timeout)
```

```
@DTW_ACCEPT(handle)
```

值

表 113. 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 <code>DTW_STATIC()</code> 函數上所指定的任何逾時值。

範例

範例 1：

```
%DEFINE handle = ""  
@DTW_RTVHANLDE(handle)
```

```
%HTML(REPORT){  
@DTW_ACCEPT(handle)  
...  
%}
```

DTW_COMMIT

AIX	HP-UX	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()
```

值

無。

範例

範例 1：指定確定

```
@DTW_COMMIT()  
%HTML(report){  
%}
```

DTW_ROLLBACK

AIX	HP-UX	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()
```

值

無。

範例

範例 1：指定回捲

```
@DTW_ROLLBACK()  
%HTML(report){  
%}
```

DTW_RTVHANDLE

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

目的

產生並傳回一個異動 `handle`，它對於經過多個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（若有的話）的結合來計算它的值。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。

語法

```
@DTW_RTVHANDLE(handle)
```

值

表 114. *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。

範例

範例 1：定義用來取回異動 `handle` 的 `handle` 變數

```
%DEFINE handle = ""  
@DTW_RTVHANLDE(handle)
```


DTW_STATIC

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

目的

指出整個巨集是持續的。它應該是巨集中的第一個陳述式。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。

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。

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。

語法

`@DTW_STATIC(timeout)`

`@DTW_STATIC()`

值

表 115. *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)。

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	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。

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。當巨集不在持續狀態時，若呼叫這個函數，將發生 Net.Data 錯誤。

語法

```
@DTW_TERMINATE()
```

值

無

範例

範例：終止持續異動

```
%HTML(QUIT){  
@DTW_TERMINATE()  
...  
%}
```

附錄A. 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270頁的『SQL_MESSAGE』
- 第271頁的『SQL_REPORT』
- 第271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18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28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28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 $(dbtbl) SET URL='$(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49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54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122頁的『RETURN_CODE』。

附錄B. 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。

列示在此的一些特性在一般可用性上仍不適用。

表 116. Net.Data 語言環境

語言環境	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
預設值	X	X	X	X	X	X	X	X
純本文檔介面	X	X	X	X	X	X	X	X
IMS Web	X			X				X
Java Applets	X	X	X	X		X	X	X
Java 應用程式	X		X				X	X
ODBC	X	X	X	X		X	X	X
Oracle	X							X
Perl	X		X	X				X
REXX	X		X	X	X	X	X	X
SQL	X	X	X	X	X	X	X	X
Sybase	X							X
系統	X	X	X	X	X	X	X	X
Web 登記	X	X	X		X	X	X	X

表 117. Net.Data 儲存程序資料類型

資料類型	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
BIGINT	X	X	X			X	X	X
BLOB	X	X	X			X	X	X
CHAR	X	X	X	X	X	X	X	X
CLOB	X	X	X			X	X	X
DATE	X	X	X		X	X	X	X
DBCLOB	X	X	X			X	X	X
DECIMAL	X	X	X	X	X	X	X	X
DOUBLE	X	X	X	X	X	X	X	X
DOUBLEPRECISION	X	X	X	X	X	X	X	X
FLOAT	X	X	X	X	X	X	X	X
INTEGER	X	X	X	X	X	X	X	X
GRAPHIC	X	X	X	X	X	X	X	X
LONGVARCHAR	X	X	X		X	X	X	X
LONGVARGRAPHIC	X	X	X		X	X	X	X

表 117. Net.Data 儲存程序資料類型 (繼續)

資料類型	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
REAL	X	X	X		X	X	X	X
SMALLINT	X	X	X	X	X	X	X	X
TIME	X	X	X		X	X	X	X
TIMESTAMP	X	X	X		X	X	X	X
VARCHAR	X	X	X	X	X	X	X	X
VARGRAPHIC	X	X	X	X	X	X	X	X

表 118. Net.Data 架構變數

架構變數	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
CACHE_MACHINE	X							X
CACHE_PORT	X							X
DefaultDBCp				X				
DB2INSTANCE	X	X	X			X	X	X
DB2MSG5				X				
DB2PLAN				X				
DB2SSID				X				
DSNAOINI				X				
DTW_CM_PORT	X	X	X			X	X	X
DTW_INST_DIR	X	X	X			X	X	X
DTW_LOG_DIR	X	X	X			X	X	X
DTW_MBMODE	X	X	X	X		X	X	X
DTW_OPTIMIZE_MATH	X	X	X			X	X	X
DTW_REMOVE_WS				X				
DTW_SMTP_SERVER	X	X	X			X	X	X
DTW_SQL_ISOLATION					X			
DTW_SQL_NAMING_MODE					X			
DTWR_CLOSE_REGISTRIES					X			

表 119. Net.Data 變數

變數	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
ALIGN	X	X	X	X	X	X	X	X
DATABASE	X	X	X		X	X	X	X
DB_CASE	X	X	X	X	X	X	X	X
DB2PLAN				X				
DB2SSID				X				
DTW_APPLET_ALTTEXT	X	X	X	X		X	X	X
DTW_CURRENT_FILENAME	X	X	X	X	X	X	X	X

表 119. Net.Data 變數 (繼續)

變數	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTW_CURRENT_LAST_MODIFIED	X	X	X	X	X	X	X	X
DTW_DEFAULT_MESSAGE					X			
DTW_DEFAULT_REPORT	X	X	X	X	X	X	X	X
DTW_EDIT_CODES					X			
DTW_HTML_TABLE	X	X	X	X	X	X	X	X
DTW_LOG_LEVEL	X	X	X			X	X	X
DTW_MACRO_FILENAME	X	X	X	X	X	X	X	X
DTW_MACRO_LAST_MODIFIED	X	X	X	X	X	X	X	X
DTW_MBMODE	X	X	X	X		X	X	X
DTW_MP_PATH	X	X	X	X	X	X	X	X
DTW_MP_VERSION	X	X	X	X	X	X	X	X
DTW_PRINT_HEADER	X	X	X	X	X	X	X	X
DTW_REMOVE_WS	X	X	X	X	X	X	X	X
DTW_SAVE_TABLE_IN	X	X	X	X	X	X	X	X
DTW_SET_TOTAL_ROWS	X	X	X		X	X	X	X
LOCATION				X				
LOGIN	X	X	X		X	X	X	X
Nn	X	X	X	X	X	X	X	X
NLIST	X	X	X	X	X	X	X	X
NULL_RPT_FIELD					X			
NUM_COLUMNS	X	X	X	X	X	X	X	X
NUM_ROWS					X			
PASSWORD	X	X	X		X	X	X	X
RETURN_CODE	X	X	X	X	X	X	X	X
ROW_NUM	X	X	X	X	X	X	X	X
RPT_MAX_ROWS	X	X	X	X	X	X	X	X
SHOWSQL	X	X	X	X	X	X	X	X
SQL_CODE	X	X	X	X	X	X	X	X
SQL_STATE	X	X	X	X	X	X	X	X
START_ROW_NUM	X	X	X		X	X	X	X
TOTAL_ROWS	X	X	X		X	X	X	X
TRANSACTION_SCOPE	X	X	X	X	X	X	X	X
V_columnName	X	X	X	X	X	X	X	X
VLIST	X	X	X	X	X	X	X	X
Vn	X	X	X	X	X	X	X	X

表 120. Net.Data 函數

函數	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTW_ACCEPT					X			
DTW_ADD	X	X	X	X	X	X	X	X
DTW_ADDQUOTE	X	X	X	X	X	X	X	X
DTW_ASSIGN	X	X	X	X	X	X	X	X
DTW_CACHE_PAGE	X							X
DTW_COMMIT					X			
DTW_CONCAT	X	X	X	X	X	X	X	X
DTW_DATE	X	X	X	X	X	X	X	X
DTW_DELSTR	X	X	X	X	X	X	X	X
DTW_DELOWORD	X	X	X	X	X	X	X	X
DTW_DIVIDE	X	X	X	X	X	X	X	X
DTW_DIVREM	X	X	X	X	X	X	X	X
DTW_EXIT	X	X	X		X	X	X	X
DTW_FORMAT	X	X	X	X	X	X	X	X
DTW_GETCOOKIE	X	X	X			X	X	X
DTW_GETENV	X	X	X	X	X	X	X	X
DTW_GETINIDATA	X	X	X	X	X	X	X	X
DTW_HTMLLENCODE	X	X	X	X	X	X	X	X
DTW_INSERT	X	X	X	X	X	X	X	X
DTW_INTDIV	X	X	X	X	X	X	X	X
DTW_LASTPOS	X	X	X	X	X	X	X	X
DTW_LENGTH	X	X	X	X	X	X	X	X
DTW_LOWERCASE	X	X	X	X	X	X	X	X
DTW_MULTIPLY	X	X	X	X	X	X	X	X
DTW_POS	X	X	X	X	X	X	X	X
DTW_POWER	X	X	X	X	X	X	X	X
DTW_QHTMLLENCODE	X	X	X	X	X	X	X	X
DTW_REVERSE	X	X	X	X	X	X	X	X
DTW_ROLLBACK					X			
DTW_RVTHANDLE					X			
DTW_SENDMAIL	X	X	X			X	X	X
DTW_SETCOOKIE	X	X	X			X	X	X
DTW_SETENV	X	X	X	X	X	X	X	X
DTW_STATIC					X			
DTW_STRIP	X	X	X	X	X	X	X	X
DTW_SUBSTR	X	X	X	X	X	X	X	X

表 120. Net.Data 函數 (繼續)

函數	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTW_SUBTRACT	X	X	X	X	X	X	X	X
DTW_SUBWORD	X	X	X	X	X	X	X	X
DTW_TB_APPENDROW					X			
DTW_TB_COLS	X	X	X		X	X	X	X
DTW_TB_DELETEROW					X			
DTW_TB_DLIST	X	X	X	X	X	X	X	X
DTW_TB_DUMPH	X	X	X	X	X	X	X	X
DTW_TB_DUMPV	X	X	X	X	X	X	X	X
DTW_TB_GETN	X	X	X		X	X	X	X
DTW_TB_GETV	X	X	X		X	X	X	X
DTW_TB_HTMLENCOD	X	X	X	X	X	X	X	X
DTW_TB_INPUT_CHECKBOX	X	X	X	X	X	X	X	X
DTW_TB_INPUT_RADIO	X	X	X	X	X	X	X	X
DTW_TB_INPUT_TEXT	X	X	X	X	X	X	X	X
DTW_TB_INSERTCOL					X			
DTW_TB_INSERTROW					X			
DTW_TB_LIST	X	X	X	X	X	X	X	X
DTW_TB_MAXROWS					X			
DTW_TB_QUERYCOLNONJ					X			
DTW_TB_ROWS	X	X	X		X	X	X	X
DTW_TB_SELECT	X	X	X	X	X	X	X	X
DTW_TB_SETCOLS					X			
DTW_TB_SETN					X			
DTW_TB_SETV					X			
DTW_TB_TABLE	X	X	X	X	X	X	X	X
DTW_TB_TEXTAREA	X	X	X	X	X	X	X	X
DTW_TERMINATE					X			
DTW_TIME	X	X	X	X	X	X	X	X
DTW_TRANSLATE	X	X	X	X	X	X	X	X
DTW_UPPERCASE	X	X	X	X	X	X	X	X
DTW URLESCSEQ	X	X	X	X	X	X	X	X
DTW_WORD	X	X	X	X	X	X	X	X
DTW_WORDINDEX	X	X	X	X	X	X	X	X
DTW_WORDLENGTH	X	X	X	X	X	X	X	X
DTW_WORDPOS	X	X	X	X	X	X	X	X
DTW_WORDS	X	X	X	X	X	X	X	X

表 120. Net.Data 函數 (繼續)

函數	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
DTWF_APPEND	X	X	X	X	X	X	X	X
DTWF_CLOSE	X	X	X	X	X	X	X	X
DTWF_DELETE	X	X	X	X	X	X	X	X
DTWF_INSERT	X	X	X	X	X	X	X	X
DTWF_OPEN	X	X	X	X	X	X	X	X
DTWF_READ	X	X	X	X	X	X	X	X
DTWF_REMOVE	X	X	X	X	X	X	X	X
DTWF_SEARCH	X	X	X	X	X	X	X	X
DTWF_UPDATE	X	X	X	X	X	X	X	X
DTWF_WRITE	X	X	X	X	X	X	X	X
DTWR_ADDENTRY	X	X	X		X	X	X	X
DTWR_CLEARREG	X	X	X		X	X	X	X
DTWR_CLOSEREG					X			
DTWR_CREATEREG	X	X	X		X	X	X	X
DTWR_DELENTY	X	X	X		X	X	X	X
DTWR_DELREG	X	X	X		X	X	X	X
DTWR_LISTREG	X	X	X		X	X	X	X
DTWR_LISTSUB	X	X	X			X	X	X
DTWR_OPENREG					X			
DTWR_RTVENTRY	X	X	X		X	X	X	X
DTWR_UPDATEENTRY	X	X	X		X	X	X	X

表 121. Net.Data 介面

介面類型	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
FastCGI	X							
CGI	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	X
Lotus Domino Go Web Server (GWAPI)	X		X	X				X
Netscape API (NSAPI)	X						X	X
Servlets	X							X

表 122. Net.Data 工具

工具	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
管理工具	X		X					X

表 122. Net.Data 工具 (繼續)

工具	AIX	HP	OS/2	OS/390	OS/400	SCO	SUN	Win NT
NetObjects Fusion Plug-ins								X
精靈	X	X	X			X	X	X

附錄C. 注意事項

本書是針對 IBM 在美國所提供之產品及服務程式開發出來的。在其他國家中，IBM 不見得有提供本書中所提及的各項產品、服務或特性。請連絡您當地的 IBM 業務代表，以取得產品、服務或特性的相關資訊。凡提及 IBM 產品、程式或服務時，並不表示或暗示只可使用 IBM 產品、程式或服務。只要不違反 IBM 的智慧財產權，任何功能上相等的產品、程式或服務，都可以用來代替 IBM 產品、程式或服務。但是，使用者對於任意非 IBM 產品、程式或服務所作的評估與驗證，須自行負責。

本文件中包含著 IBM 所擁有之專利或暫准專利。因此修改本文件，並不會讓您享有這些專利權的使用權。您可以用書面方式，將授權要求寄到：

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

欲取得二位元組字元集 (DBCS) 的相關資訊，請洽當地的「IBM 智慧財產權部門」，或是將您的要求寄至下列地址：

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

下列段落不適用於「英國」或任何法律與其相抵觸之其他國家：國際商業機器股份有限公司 (IBM) 係以『交付時之現狀』提供本書，而不提供任何明示或默示之保證，其中包括 (但不限於) 為特定目的而無傷害性、適售性的保證。有些國家並不允許在某些交易中不作明示或暗示的保證，因此，此聲明可能亦不適用。

本書中可能有技術上或排版印刷上的訛誤。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，但恕不另行通知。

獲得本程式的授權者，如需其相關資訊作為下列用途：(i) 在獨立創作的程式與其他程式 (包括本程式在內) 之間交換資訊 (ii) 互相使用彼此交換的資訊，請洽：

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

上述資訊之取得有其特殊要件，在某些情況下，必須付費方得使用。

IBM 提供本資訊所描述的授權程式以及其所能使用所有授權資料，皆受「IBM 客戶合約」或與客戶之間的任何同等合約的管束。

有關非 IBM 產品的資訊皆取自該產品的提供者、其出版聲明或其他公開的可用來源。IBM 並無測試那些產品，也無法確認執行效能的精確度、相容性或任何與非 IBM 產品相關的要求。若有關非 IBM 產品的功能問題，您應向該產品的提供者申訴。

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

下列詞彙是 IBM 公司在美國或其它國家或此兩者的商標：

AIX	Lotus
DataJoiner	MVS
DB2	Net.Data
Domino	OS/2
IBM	OS/390
IMS	OS/400

下列詞彙是以下其它公司的商標：

Java 及 HotJava 是 Sun Microsystems, Inc. 的商標。

Microsoft、Windows、Windows NT® 以及 Windows 95 標誌是 Microsoft Corporation 的註冊商標。

UNIX 是透過 X/Open Company Limited 獨家授權，在美國以及其它國家的註冊商標。

其它的公司、產品與服務名稱可能為其它公司的註冊商標或服務標記。

名詞解釋

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。

通用資源位置 (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>

通用閘道介面 (Common Gateway Interface). 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。有些語言環境是由 Net.Data (如 REXX、Perl 及 Oracle) 所提供。您也可建立您自己的語言環境。

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 可支援下列具有專利的 Web 伺服器 API，可讓您可在 CGI 處理上提高執行效能：ICAPI、GWAPI、ISAPI 及 NSAPI。

A

API. 應用程式設計介面。Net.Data 支援三種週邊 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 是與可使用 Java 的瀏覽器（如 Netscape）一起使用，且是在當 HTML 頁面被載入時載入。

C

CGI. 通用閘道介面。

D

DBMS. 資料庫管理系統。

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)

ICS. Internet Connection Server。

ICSS. Internet Connection Secure Server。

Internet. 一個國際公用的 TCP/IP 電腦網路。

Internet Connection Secure Server. IBM 之具安全特性的 Web 伺服器。

Internet Connection Server. IBM 之無安全特性的 Web 伺服器。

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. 一種解譯過的程式設計語言。

T

TCP/IP. 傳輸控制通信協定 / 網際網路通信協定。

U

URL. 通用資源位置。

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 89

“前一步” 按鈕, RPT_MAX_ROWS 89

一般函數 124

DTW_ADDQUOTE 126

DTW_CACHE_PAGE 128

DTW_DATE 132

DTW_EXIT 134

DTW_GETCOOKIE 135

DTW_GETENV 137

DTW_GETINIDATA 138

DTW_HTMLENCODE 139

DTW_QHTMLENCODE 141

DTW_SENDEMIL 142

DTW_SETCOOKIE 146

DTW_SETENV 149

DTW_TIME 150

DTW_URLESCSEQ 152

〔三劃〕

上限 60

〔四劃〕

內建函數 123

支援特性表格 271

日期變數 111

〔五劃〕

巨集檔

行長度限制 4

宣告部分 2

格式 3

停止處理程序 134

常見語法元素 5

語言結構 1

樣本 3

巨集檔 (繼續)

整體語法 1

HTML 部份 2

平台支援參考手冊 271

〔六劃〕

列示有限制的字串 72

列示變數

值區隔符號 73

說明 72

範例 72

名詞解釋 282

字串

有分隔設定的值 72

條件式處理程序 31, 62

說明 6

數字比較 31, 62

字串函數

DTW_ASSIGN 169

DTW_CONCAT 170

DTW_DELSTR 171

DTW_INSERT 172

DTW_LASTPOS 174

DTW_LENGTH 175

DTW_LOWERCASE 176

DTW_POS 177

DTW_REVERSE 178

DTW_STRIP 179

DTW_SUBSTR 180

DTW_TRANSLATE 182

DTW_UPPERCASE 184

MBCS 支援 168

字串的數字比較 31, 62

字組函數

DTW_DELWORD 186

DTW_SUBWORD 188

DTW_WORD 190

DTW_WORDINDEX 191

DTW_WORDLENGTH 192

DTW_WORDPOS 193

DTW_WORDS 195

MBCS 支援 185

存取純本文檔 225

安全

密碼 106

登入 ID 104

安全建議, FFI_PATH 227

有分隔設定的值串 72

行長度限制, 巨集檔 4

〔七劃〕

位置, 純本文檔 226

作業系統參考手冊 271

系統語言環境, 傳送參數 24

〔八劃〕

併入檔 38

函數

一般 124

字串 168

字組 185

命名慣例 123

表格 196

持續 261

純本文檔介面 (FFI) 225

傳送值群組 73

算術 154

說明 123

Web 登記 248

函數呼叫

處理表格列 57

製作輸出格式 54

語法 25

說明 25

呼叫

外部程式 16

函數 25

呼叫 FFI 語言環境 225

注意事項 281

表格

結果在 HTML 87

Net.Data, 設定列數 88

表格函數

DTW_TB_APPENDROW 197

DTW_TB_COLS 198

DTW_TB_DELETEROW 199

DTW_TB_DLIST 200

DTW_TB_DUMPV 202

表格函數 (繼續)

DTW_TB_DUMPV 203

DTW_TB_GETN 204

DTW_TB_GETV 205

DTW_TB_HTMLLENCODE 206

DTW_TB_INPUT_CHECKBOX 207

DTW_TB_INPUT_RADIO 208

DTW_TB_INPUT_TEXT 209

DTW_TB_INSERTCOL 211

DTW_TB_INSERTROW 212

DTW_TB_LIST 213

DTW_TB_MAXROWS 215

DTW_TB_QUERYCOLNONJ 216

DTW_TB_ROWS 217

DTW_TB_SELECT 218

DTW_TB_SETCOLS 219

DTW_TB_SETN 220

DTW_TB_SETV 221

DTW_TB_TABLE 222

DTW_TB_TEXTAREA 224

表格處理程序變數

為 SQL 語言環境而設 101

說明 74

NLIST 76

NUM_COLUMNS 77

NUM_ROWS 78

Nn 75

ROW_NUM 79

TOTAL_ROWS 80

VLIST 82

Vn 83

V_columnName 81

表格變數

說明 73

範例 73

表頭 38

〔九劃〕

宣告部份, 巨集檔 2

持續巨集函數

DTW_ACCEPT 262

DTW_COMMIT 264

DTW_ROLLBACK 265

DTW_RTVHANDLE 266

DTW_STATIC 267

DTW_TERMINATE 268

架構 FFI 語言環境 227
限制資料庫存取 104, 106

〔十劃〕

純本文檔

在現行目錄中建立 226
存取 225
存取時的建議 226
安全建議 227
位置
現行目錄 226
FFI_PATH 226
定義 225
架構規則 227
區隔字元 228
授權需求 227
符合 FFI_PATH 226
絕對路徑 226
資料來源 225
鎖定檔案 228

迴路 62

〔十一劃〕

區域 DB2 子系統的 ID 97

區隔字元, FFI 語言環境

ASCII TEXT 228
DELIMITED 228

參數, 傳送 24

執行效能, DTW_EXIT 134

執行變數

作為變數參照 70
與參數 71
說明 70
範例 70

捲動, 使用“下一步”及“前一步”按鈕 89

授權需求, FFI_PATH 227

條件式字串處理程序 31, 62

條件變數

以 LIST 陳述式定義 69
以變數參照定義 68
說明 68
範例 72

現行目錄, 決定純本文檔 226

規劃連接 DB2 子系統 96

設定 SQL 指令字體 95

設定大寫字體 95

設定小寫字體 95

連接 DB2 子系統

位置 103

的子系統 ID 97

DB2 規劃 96

連接 DB2 子系統的子系統 ID 97

連接 DB2 子系統的位置 103

連接資料庫, DATABASE 變數 93

〔十二劃〕

備註區塊

語法 8

說明 8

報告

置換 Net.Data 預設值 86

製作格式 54

報告變數

說明 84

ALIGN 85

DTW_DEFAULT_REPORT 86

DTW_HTML_TABLE 87

RPT_MAX_ROWS 88

START_ROW_NUM 89

絕對路徑, 純本文檔的 226

〔十三劃〕

傳送值群組 73

傳送參數, 「系統」語言環境 24

資料庫一致性, 異動範圍 109

預設的文字訊息 114

〔十四劃〕

算術函數

DTW_ADD 155

DTW_DIVIDE 156

DTW_DIVREM 158

DTW_FORMAT 160

DTW_INTDIV 163

DTW_MULTIPLY 164

DTW_POWER 165

DTW_SUBTRACT 166

語言結構

巨集檔

語法 1

語言結構 (繼續)

巨集檔 (繼續)

說明 6

字串 6

函數呼叫 25

常見語法元素 5

備註區塊 8

變數名稱 5

變數參照 5

DB2 WWW 連線 269

DEFINE 區塊或陳述式 10

ENVVAR 陳述式 15

EXEC 區塊或陳述式 16

FUNCTION 區塊 18

HTML 區塊 28

IF 區塊 31

INCLUDE 陳述式 38

INCLUDE_URL 陳述式 41

LIST 陳述式 43

MACRO_FUNCTION 區塊 45

MESSAGE 區塊 49

REPORT 區塊 54

ROW 區塊 57

TABLE 陳述式 60

WHILE 區塊 62

語言環境變數

說明 92

DATABASE 93

DB2PLAN 96

DB2SSID 97

DB_CASE 95

DTW_APPLET_ALTTEXT 98

DTW_EDIT_CODES 99

DTW_MBMODE 100

DTW_SAVE_TABLE_IN 101

DTW_SET_TOTAL_ROWS 102

LOCATION 103

LOGIN 104

NULL_RPT_FIELD 105

PASSWORD 106

SHOWSQL 107

SQL_STATE 108

TRANSACTION_SCOPE 109

遠端 DB2 子系統的位置 103

〔十五劃〕

標底 38

〔十六劃〕

錯誤處理 49

〔十七劃〕

檔案位置變數 111

環境變數

說明 69

範例 69

ENVVAR 陳述式 15

隱藏變數

步驟 71

說明 71

範例，在 HTML 套表 71

隱藏變數名稱 71

〔十八劃〕

鎖定檔案, FFI 函數 228

雜項變數

說明 111

DTW_CURRENT_FILENAME 112

DTW_CURRENT_LAST_MODIFIED 113

DTW_DEFAULT_MESSAGE 114

DTW_MACRO_LAST_MODIFIED 117

DTW_MP_PATH 118

DTW_MP_VERSION 119

DTW_PRINT_HEADER 120

DTW_REMOVE_WS 121

RETURN_CODE 122

〔二十劃〕

釋放檔案, FFI 函數 228

〔二十三劃〕

變數

列示 72

表格 73, 74

執行 70

條件 68

變數 (繼續)

- 報告 84
- 語言環境 92
- 環境 69
- 隱藏 71
- 雜項 111
- Net.Data, 概觀 67
- 變數名稱 5
- 變數參照 5
- 顯示 SQL 狀態 108

A

- ALIGN 85
- APPLET 標籤, 替代文字 98

C

- cookies
 - 傳送 120
 - DTW_GETCOOKIE 135
 - DTW_PRINT_HEADER 120
 - DTW_SETCOOKIE 146

D

- DATABASE 93
- DB2 WWW 連線, 語言結構 269
- DB2PLAN 96
- DB2SSID 97
- DB_CASE 95
- DEFINE 區塊
 - 語法 10
 - 說明 10
- DEFINE 陳述式
 - 語法 10
 - 說明 10
- DTWF_APPEND 229
- DTWF_CLOSE 228, 232
- DTWF_DELETE 233
- DTWF_INSERT 235
- DTWF_OPEN 228, 237
- DTWF_READ 239
- DTWF_REMOVE 241
- DTWF_SEARCH 242
- DTWF_UPDATE 244
- DTWF_WRITE 246

- DTWR_ADDENTRY 248
- DTWR_CLEARREG 250
- DTWR_CLOSEREG 251
- DTWR_CREATEREG 252
- DTWR_DELENTY 253
- DTWR_DELREG 254
- DTWR_LISTREG 255
- DTWR_LISTSUB 256
- DTWR_OPENREG 257
- DTWR_RTVENTRY 258
- DTWR_UPDATEENTRY 260
- DTW_ACCEPT 262
- DTW_ADD 154
- DTW_ADDQUOTE 126
- DTW_APPLET_ALTTEXT 98
- DTW_ASSIGN 75, 168, 169
- DTW_CACHE_PAGE 128
- DTW_COMMIT 264
- DTW_CONCAT 170
- DTW_CURRENT_FILENAME 112
- DTW_CURRENT_LAST_MODIFIED 113
- DTW_DATE 132
- DTW_DEFAULT_MESSAGE 114
- DTW_DEFAULT_REPORT 86
- DTW_DELSTR 171
- DTW_DELWORD 185
- DTW_DIVIDE 156
- DTW_DIVREM 158
- DTW_EDIT_CODES 99
- DTW_FORMAT 160
- DTW_GETCOOKIE 135
- DTW_GETENV 137
- DTW_GETINIDATA 138
- DTW_HTML_ENCODE 139
- DTW_HTML_TABLE 87
- DTW_INSERT 172
- DTW_INTDIV 163
- DTW_LASTPOS 174
- DTW_LENGTH 175
- DTW_LOG_LEVEL 115
- DTW_LOWERCASE 176
- DTW_MACRO_FILENAME 116
- DTW_MACRO_LAST_MODIFIED 117
- DTW_MBMODE 100
- DTW_MP_PATH 118
- DTW_MP_VERSION 119

DTW_MULTIPLY 164
DTW_POS 177
DTW_POWER 165
DTW_PRINT_HEADER 120
DTW_QHTMLENCODE 141
DTW_REMOVE_WS 121
DTW_REVERSE 178
DTW_ROLLBACK 265
DTW_RTVHANDLE 266
DTW_SAVE_TABLE_IN 101
DTW_SENDMAIL 142
DTW_SETCOOKIE 146
DTW_SETENV 149
DTW_SET_TOTAL_ROWS 102
DTW_STATIC 267
DTW_STRIP 179
DTW_SUBSTR 180
DTW_SUBTRACT 166
DTW_SUBWORD 188
DTW_TB_APPENDROW 197
DTW_TB_COLS 198
DTW_TB_DELETEROW 199
DTW_TB_DLIST 200
DTW_TB_DUMP 202
DTW_TB_DUMPV 203
DTW_TB_GETN 204
DTW_TB_GETV 205
DTW_TB_HTML ENCODE 206
DTW_TB_INPUT_CHECKBOX 207
DTW_TB_INPUT_RADIO 208
DTW_TB_INPUT_TEXT 209
DTW_TB_INSERTCOL 211
DTW_TB_INSERTTROW 212
DTW_TB_LIST 211
DTW_TB_MAXROWS 215
DTW_TB_QUERYCOLNONJ 216
DTW_TB_ROWS 217
DTW_TB_SELECT 218
DTW_TB_SETCOLS 219
DTW_TB_SETN 220
DTW_TB_SETV 221
DTW_TB_TABLE 222
DTW_TB_TEXTAREA 224
DTW_TERMINATE 268
DTW_TIME 150
DTW_TRANSLATE 182

DTW_UPPERCASE 184
DTW_URLESCSEQ 152
DTW_WORD 190
DTW_WORDINDEX 191
DTW_WORDLENGTH 192
DTW_WORDPOS 193
DTW_WORDS 195

E

ENVVAR 陳述式 69
 語法 15
 說明 15
EXEC 區塊
 語法 16
 說明 16
EXEC 陳述式 70
 語法 16
 說明 16
EXEC_PATH 16
EXEC_SQL 269

F

FFI 函數
 鎖定檔案 228
 釋放檔案 228
 DTWF_APPEND 230
 DTWF_CLOSE 232
 DTWF_DELETE 233
 DTWF_INSERT 235
 DTWF_OPEN 237
 DTWF_READ 239
 DTWF_REMOVE 241
 DTWF_SEARCH 242
 DTWF_UPDATE 244
 DTWF_WRITE 246
FFI 語言環境
 存取檔案 225
 安全建議 227
 架構規則 227
 區隔字元 228
 授權需求 227
 現行目錄 226
 檔案位置 226
FFI_PATH
 存取純本文檔 225

FFI_PATH (繼續)

安全建議 227

使路徑與 *filename* 參數相符 226

架構規則 227

純本文檔位置 226

語法 225

FUNCTION 區塊

語法 19

說明 18

H

HTML

套表，輸入使用者 ID 104

套表，輸入密碼 106

隱藏變數名稱 71

顯示表格結果於 87

HTML 區塊

語法 28

說明 28

HTML 部份，巨集檔 2

HTML_INPUT 區塊 269

HTML_REPORT 區塊 269

I

IF 區塊

語法 31

說明 31

IN 關鍵字 20, 46, 123

INCLUDE 陳述式

語法 38

說明 38

INCLUDE_PATH 38

INCLUDE_URL 陳述式

語法 41

說明 41

INOUT 關鍵字 20, 46, 123

L

LIST 陳述式

語法 43

說明 43

LOCATION 103

LOGIN 104

M

MACRO_FUNCTION 區塊

語法 45

說明 45

MBCS 支援函數

字串函數 168

字組函數 185

MESSAGE 區塊

語法 49

說明 49

N

Net.Data 表格

上限 60

定義 60

NLIST 76

NULL_RPT_FIELD 105

NUM_COLUMNS 77

NUM_ROWS 78

Nn 75

O

OUT 關鍵字 20, 46, 123

P

PASSWORD 106

R

REPORT 區塊

表格變數 73

語法 54

說明 54

ALIGN 85

DTW_DEFAULT_REPORT 86

DTW_HTML_TABLE 87

NLIST 76

NUM_COLUMNS 77

NUM_ROWS 78

Nn 75

RPT_MAX_ROWS 88

START_ROW_NUM 89

TOTAL_ROWS 80

RETURNS 關鍵字 20

RETURN_CODE 122

ROW 區塊

語法 57

說明 57

NLIST 76

NUM_COLUMNS 77

NUM_ROWS 78

Nn 75

ROW_NUM 79

TOTAL_ROWS 80

Vn 82, 83

V_columnName 81

ROW_NUM 79

RPT_MAX_ROWS 88

Web 登記函數 (繼續)

DTWR_CLEARREG 250

DTWR_CLOSEREG 251

DTWR_CREATEREG 252

DTWR_DELENTY 253

DTWR_DELREG 254

DTWR_LISTREG 255

DTWR_LISTSUB 256

DTWR_OPENREG 257

DTWR_RTVENTRY 258

DTWR_UPDATEENTRY 260

Web 瀏覽器的替代文字 98

WHILE 區塊 62

語法 62

說明 62

S

SHOWSQL 107

SQL

設定指令字體 95

隱藏或顯示 107

SQL 區塊 269

SQL_CODE 271

SQL_MESSAGE 區塊 270

SQL_REPORT 區塊 271

SQL_STATE 108

START_ROW_NUM 89

T

TABLE 陳述式 73

語法 60

說明 60

TOTAL_ROWS 80

TRANSACTION_SCOPE 109

V

VLIST 82

Vn 83

V_columnName 81

W

Web 登記函數

DTWR_ADDENTRY 249

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
台灣地區郵政管理局 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