擴充

ADOBE® FLASH® PROFESSIONAL CS5 及 CS5.5

上次更新 **2011/5/16**

法律聲明

目錄

第 47 章 Tools 物件

第 48 章 Vertex 物件

第 49 章 VideoItem 物件

第 50 章 XMLUI 物件

第 51 章 C 語言層次擴充

第 1 章 簡介

身為一名 Adobe® Flash® Professional CS5 或 CS5.5 的使用者，您可能已經熟悉使用 Adobe® ActionScript® 的方法，並建立可以在 Adobe® Flash® Player 執行階段中執行的指令碼。在此文件中說明的 Flash JavaScript 應用程式程式設計介面 (JavaScript API) 是免費的程式設計工具，可以用來建立在編寫環境中執行的指令碼。

本文件說明 JavaScript API 中可用的物件、方法和屬性。這裡假設您知道在編寫環境中工作時，該如何使用正式的命令。倘若您對於特定命令的功能有所疑問，請使用 「Flash 說明」中的其它文件 (例如，「使用 Flash」) 尋找該項資訊。

本文件假設您熟悉 JavaScript 或 ActionScript 語法和基本程式設計概念 (例如函數、參數與資料類型)。

使用 JavaScript API

Flash JavaScript API 可讓您撰寫指令碼，在 Flash 編寫環境中 (也就是使用者開啟 Flash 程式時) 執行數種動作。這項功能與 ActionScript 語言不同，後者可讓您撰寫在 Flash Player 環境中 (亦即 SWF 檔正在播放時) 執行動作的指令碼。這項功能也與 JavaScript 命令 (您可能會在網頁瀏覽器顯示的頁面中使用) 不同。

您可以使用 JavaScript API 來撰寫 Flash 應用程式指令碼，以便協助精簡編寫程序。例如，您可以撰寫自動化重複工作的指令碼，或將自訂工具新增至 「工具」面板。

Flash JavaScript API 的設計與 Adobe® Dreamweaver® 和 Adobe® Fireworks® JavaScript API (依據 Netscape JavaScript API 所設計) 相似。Flash JavaScript API 依據「文件物件模型」(Document Object Model，DOM) 設計，因此可以使用 JavaScript 物件存取 Flash 文件。Flash JavaScript API 包括 Netscape JavaScript API 的所有元素，再加上 Flash DOM。本文件將說明這些增加的物件與其方法和屬性。Flash 指令碼中可以使用原生 JavaScript 語言的任何元素，不過，只有在 Flash 文件內容中有意義的元素才有效。

此外，JavaScript API 還包含一些方法，讓您可以利用各種 JavaScript 及自訂的 C 程式碼組合來實作擴充程式。如需詳細資訊，請參閱：第 489 頁 「C 語言層次擴充」。

Flash 中的 JavaScript 解譯器為 1.6 版的 Mozilla SpiderMonkey 引擎，可在下列網站取得：www.mozilla.org/js/spidermonkey/。SpiderMonkey 為 Mozilla.org 開發的兩種 JavaScript 語言參考實作之一，與嵌入 Mozilla 瀏覽器的引擎相同。

SpiderMonkey 依照 ECMAScript (ECMA-262) Edition 3 的語言規格中的定義實作整個核心 JavaScript 語言，而且和這個規格完全相容。僅不支援不屬於 ECMA-262 規格的瀏覽器專用主機物件。同樣地，許多 JavaScript 參考指南會區分核心 JavaScript 與用戶端 (瀏覽器相關) JavaScript。只有核心 JavaScript 適用於 Flash JavaScript 解譯器。

建立 JSFL 檔

您可以使用 Adobe Flash Professional 或慣用的文字編輯器來撰寫和編輯 Flash JavaScript (JSFL) 檔案。如果您使用的是 Flash，這些檔案的預設副檔名是 .jsfl。若要撰寫指令碼，選取 「檔案 > 新增 > Flash JavaScript 檔案」。

您也可以選取 「操作記錄」面板中的命令，建立 JSFL 檔案。接著按一下 「操作記錄」面板中的 「儲存」按鈕，或是從面板選單中選取 「儲存成命令檔」。命令 (JSFL) 檔會儲存於 Commands 資料夾中 (請參閱第 2 頁 「儲存 JSFL 檔」)。您可以接著開啟檔案，並使用編輯其它指令碼檔案的相同方式加以編輯。

「操作記錄」面板還會提供其它有用的選項。您可以將選取的命令複製到 「剪貼簿」，也可以檢視在使用 Flash 時所產生的 JavaScript 命令。

若要將命令從 「操作記錄」面板複製到剪貼簿：

1 從 「操作記錄」面板選取一或多個命令。

2 執行下列其中一項操作：

- 按一下「複製」按鈕。
- 從面板選單選取「複製步驟」。

若要在「操作記錄」面板中檢視 **JavaScript** 命令：

- 從面板選單選取「面板」中的「檢視 > JavaScript」。

儲存 JSFL 檔

您可以在 Flash 編寫環境中提供 JSFL 指令碼，方法是將其儲存在 Configuration 資料夾內的數個資料夾之一。根據預設，Configuration 資料夾位於下列位置：

- Windows® 7™：

  < 開機磁碟 >\Users\< 使用者名稱 >\AppData\Local\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\

- Windows® Vista™：

  < 開機磁碟 >\Users\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\

- Windows XP：

  < 開機磁碟 >\Documents and Settings\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\

- Mac OS® X：

  Macintosh HD/Users/< 使用者名稱 >/Library/Application Support/Adobe/Flash CS5 或 CS5.5/< 語言 >/Configuration/

若要判斷 Configuration 資料夾的位置，請參閱 fl.configDirectory 或 fl.configURI，如下列範例所示：

```
// store directory to a variable
var configDir = fl.configDirectory;
// display directory in the Output panel
fl.trace(fl.configDirectory);
```

Configuration 資料夾內的下列資料夾可能會包含能在編寫環境中存取的指令碼：Behaviors (支援用於行為的使用者介面)、Commands (包含顯示於「命令」選單的程式碼)、JavaScript (包含由指令碼助理所使用的指令碼以填入使用者介面控制項)、Tools (包含「工具」面板中的可擴充工具) 以及 WindowSWF (包含顯示於 Windows 選單的面板)。本文件著重於用於命令和工具的指令碼。

如果編輯 Commands 資料夾中的指令碼，便可立即在 Flash 中使用新的指令碼。如果要編輯可擴充工具的指令碼，必須先關閉 Flash 然後再重新加以啟動，或者使用 fl.reloadTools() 命令。不過，如果是使用指令碼將可擴充工具新增至「工具」面板，之後又編輯過該指令碼，就必須從「工具」面板中移除該工具然後再將其加入，或者先關閉 Flash 然後再次啟動，修改過的工具才能使用。

有兩個位置可供您儲存命令和工具檔案，方便您在編寫環境中存取這些項目。

- 對於「命令」選單上顯示成項目的指令碼，請將 JSFL 檔儲存至下列位置的 Commands 資料夾內：

作業系統	位置
Windows 7	< 開機磁碟 >\Users\< 使用者名稱 >\AppData\Local\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\Commands
Windows Vista	< 開機磁碟 >\Users\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS5\< 語言 >\Configuration\Commands
Windows XP	< 開機磁碟 >\Documents and Settings\< 使用者 >\Local Settings\Application Data\Adobe\Flash CS5\< 語言 >\Configuration\Commands
Mac OS X	Macintosh HD/Users/< 使用者名稱 >/Library/Application Support/Adobe/Flash CS5/< 語言 >/Configuration/Commands

- 對於 「工具」面板中顯示成可擴充工具的指令碼，請將 JSFL 檔儲存至下列位置的 Tools 資料夾內：

作業系統	位置
Windows 7	< 開機磁碟 >\Users\< 使用者名稱 >\AppData\Local\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\Tools
Windows Vista	< 開機磁碟 >\Users\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS5\< 語言 >\Configuration\Tools
Windows XP	< 開機磁碟 >\Documents and Settings\user\Local Settings\Application Data\Adobe\Flash CS5\< 語言 >\Configuration\Tools
Mac OS X	Macintosh HD/Users/< 使用者名稱 >/Library/Application Support/Adobe/Flash CS5/< 語言 >/Configuration/Tools

如果 JSFL 檔有其它附加檔案，例如 XML 檔，請將這些檔案儲存到相同的目錄做為 JSFL 檔。

執行指令碼

您可以使用數種方式執行指令碼。本節將說明一些最常見的方式。

若要執行目前正在檢視或編輯的指令碼：

- 按一下右鍵 (Macintosh 中為 Command+ 按一下) 並選擇 「執行指令碼」。
- 在 「Script」視窗工具列上按一下 「執行指令碼」圖示。

此選項可讓您在儲存指令碼之前先執行它。此選項同時可在沒有開啟任何 FLA 檔的情況下讓您執行指令碼。

若要執行 **Commands** 資料夾中的指令碼，請執行下列步驟之一：

- 從編寫環境選取 「命令 > 指令碼名稱」。
- 使用已指定給指令碼的鍵盤快速鍵。若要指定鍵盤快速鍵，請使用 「編輯 > 鍵盤快速鍵」，然後從 「命令」彈出式選單選取 「繪圖」選單命令。展開選單樹狀結構中的 Commands 節點，以檢視可用指令碼的清單。

若要執行的命令指令碼不在 **Commands** 資料夾內，請執行下列步驟之一：

- 從編寫環境選取 「命令 > 執行命令」，然後選取要執行的指令碼。
- 在指令碼內使用 fl.runScript() 命令。
- 在檔案系統中按兩下指令碼檔。

若要將 **JSFL** 檔中實作的工具加入至 「工具」面板：

1 將工具的 JSFL 檔及其它相關檔案複製到 Tools 資料夾中 (請參閱第 2 頁 「儲存 JSFL 檔」)。

2 選取「編輯 > 自訂工具面板 (Windows) 或 Flash > 自訂工具面板 (Macintosh)」。

3 將工具增加至可用的工具清單。

4 按一下「確定」。

您可以使用 MMExecute() 函數 (內容記載於「ActionScript 3.0 語言和組件參考」中)，在 ActionScript 檔案中加入個別的 JavaScript API 命令。不過，MMExecute() 函數只有在自訂使用者介面元素內容 (例如組件「屬性」檢測器或編寫環境中的 SWF 面板) 中使用時才有作用。即使 JavaScript API 命令是從 ActionScript 呼叫，在 Flash Player 中或編寫環境外都不會有任何作用。

若要從 **ActionScript** 指令碼發出命令：

- 使用下列語法 (您可以將數個命令連結成一個字串)：

```
MMExecute(Javascript command string);
```

您也可以從命令列執行指令碼。

若要從 **Windows** 命令列執行指令碼：

- 使用下列語法 (依需要新增路徑資訊)：

```
"flash.exe" myTestFile.jsfl
```

若要從 **Macintosh** 的 **"Terminal"** 應用程式執行指令碼：

- 使用下列語法 (依需要新增路徑資訊)：

```
osascript -e 'tell application "flash" to open alias "Mac OS X:Users:user:myTestFile.jsfl" '
```

osascript 命令還可以在檔案中執行 AppleScript。例如，您可以將下列文字放在名為 myScript 的檔案中：

```
tell application "flash"
open alias "Mac OS X:Users:user:myTestFile.jsfl"
end tell
```

接著，使用下列命令執行指令碼：

```
osascript myScript
```

JavaScript API 的新增功能

Flash CS5 和 CS5.5 不只新增了一些物件、方法與屬性，並且同時移除了其它一些物件、方法與屬性。這些變更摘要如下。

如果您從未使用過 JavaScript API，可能需要略過本節並直接跳到第 6 頁「JavaScript API 物件」。

新的方法和屬性

Flash Pro CS5 中針對現有物件新增的方法和屬性如下：

- Document 物件
  - document.debugMovie()
  - document.loadCuepointXML()
- flash 物件 (fl)
  - fl.languageCode
  - fl.toggleBreakpoint

- Frame 物件
  - frame.convertMotionObjectTo2D()
  - frame.convertMotionObjectTo3D()
  - frame.getMotionObjectXML()
  - frame.hasMotionPath()
  - frame.isMotionObject()
  - frame.is3DMotionObject()
  - frame.selectMotionPath()
  - frame.setMotionObjectDuration()
  - frame.setMotionObjectXML()
  - frame.tweenInstanceName
- Timeline 物件
  - timeline.createMotionObject()
  - timeline.libraryItem
  - timeline.removeMotionObject()
  - timeline.startPlayback
  - timeline.stopPlayback

Flash Pro CS5.5 中針對現有物件新增的方法和屬性如下：

- SymbolInstance 物件
  - symbolInstance.bitmapRenderMode
  - symbolInstance.backgroundColor
  - symbolInstance.usesBackgroundColor
  - symbolInstance.visible
- Timeline 物件
  - timeline.copyLayers()
  - timeline.cutLayers()
  - timeline.duplicateLayers()
  - timeline.pasteLayers()
- flash 物件 (fl)
  - fl.getSwfPanel()
  - fl.installedPlayers()
  - fl.publishCacheEnabled
  - fl.publishCacheDiskSizeMax
  - fl.publishCacheMemorySizeMax
  - fl.publishCacheMemoryEntrySizeLimit
  - fl.clearPublishCache()

- swfPanel 物件
  - swfPanel.setFocus()

其它變更

Flash CS5 中已更新下列方法與屬性：

- fl.openScript()
- fl.publishDocument()
- fontItem.embedRanges
- fontItem.embeddedCharacters
- fontItem.embedVariantGlyphs

Flash CS5 中已不再提供下列物件與方法：

- Screen 物件
- ScreenOutline 物件
- document.canSaveAVersion()
- document.revertToLastVersion()
- document.saveAVersion()
- document.synchronizeWithHeadVersion()
- fl.downloadLatestVersion()
- fl.revertDocumentToLastVersion()
- fl.saveAVersionOfDocument()
- fl.synchronizeDocumentWithHeadVersion()

JavaScript API 物件

本節將提供 Flash JavaScript API 中可使用物件的摘要，並說明如何開始使用這些物件。使用 JavaScript API 時，也可以使用所有的標準 JavaScript 命令。

下表將簡要說明 JavaScript API 中的每個物件。物件會依英文字母順序列出。

物件	說明
actionsPanel 物件	actionsPanel 物件代表目前顯示的「動作」面板。
BitmapInstance 物件	BitmapInstance 物件為 Instance 物件的子類別，代表影格中的點陣圖。
BitmapItem 物件	BitmapItem 物件是指文件元件庫中的點陣圖。BitmapItem 物件為 Item 物件的子類別。
CompiledClipInstance 物件	CompiledClipInstance 物件為 Instance 物件的子類別。
compilerErrors 物件	compilerErrors 物件代表「編譯器錯誤」面板。它是 flash 物件 (fl.compilerErrors) 的屬性。
ComponentInstance 物件	ComponentInstance 物件為 SymbolInstance 物件的子類別，代表影格中的組件。
componentsPanel 物件	componentsPanel 物件 (代表「組件」面板) 為 flash 物件 (fl.componentsPanel) 的屬性。
Contour 物件	Contour 物件代表在形狀邊界上不完整邊緣的封閉路徑。

物件	說明
Document 物件	Document 物件代表「舞台」。
drawingLayer 物件	drawingLayer 物件可當做 flash 物件的子物件，從 JavaScript 存取。
Edge 物件	Edge 物件代表「舞台」上形狀的邊緣。
Element 物件	「舞台」上的所有物件類型皆為 Element。
Fill 物件	Fill 物件包含「工具」面板或選取形狀的填色顏色設定的所有屬性。
Filter 物件	Filter 物件包含所有濾鏡的所有屬性。
flash 物件 (fl)	flash 物件代表 Flash 應用程式。
FLfile 物件	FLfile 物件可以讓您編寫 Flash 擴充功能，以存取、修改和移除本機檔案系統中的檔案和資料夾。
folderItem 物件	folderItem 物件為 Item 物件的子類別。
fontItem 物件	fontItem 物件為 Item 物件的子類別。
Frame 物件	Frame 物件代表圖層中的影格。
HalfEdge 物件	Shape 物件邊緣的有向面 (Directed side)。
Instance 物件	Instance 物件為 Element 物件的子類別。
Item 物件	Item 物件為抽象的基底類別。
Layer 物件	Layer 物件代表時間軸內的圖層。
library 物件	Library 物件代表「元件庫」面板。
Math 物件	Math 物件可做為 flash 物件 (fl.Math) 的唯讀屬性使用。
Matrix 物件	Matrix 物件代表變形矩陣。
outputPanel 物件	OutputPanel 物件代表「輸出」面板，用來顯示語法錯誤等疑難排解資訊。它是 flash 物件 (fl.outputPanel) 的屬性。
Oval 物件	Oval 物件是使用「橢圓形」工具繪製的形狀。若要判斷某個項目是否為 Oval 物件，請使用 shape.isOvalObject。
Parameter 物件	Parameter 物件類型可由 screen.parameters 陣列 (對應 Flash 編寫工具中的螢幕「屬性」檢測器) 或由 componentInstance.parameters 陣列 (對應編寫工具中的組件「屬性」檢測器) 存取。
Path 物件	Path 物件定義連續線段 (直線、曲線或兩者)，通常用於建立可擴充工具。
presetItem 物件	presetItem 物件代表「移動預設效果」面板中的項目 (預設效果或資料夾)。
presetPanel 物件	presetPanel 物件代表「移動預設效果」面板 (「視窗 > 移動預設效果」)。它是 flash 物件 (fl.presetPanel) 的屬性。
Rectangle 物件	Rectangle 物件是使用「矩形」工具繪製的形狀。若要判斷某個項目是否為 Rectangle 物件，請使用 shape.isRectangleObject。
Screen 物件	Screen 物件代表幻燈片或表單文件中的單一螢幕。
ScreenOutline 物件	ScreenOutline 物件代表幻燈片或表單文件中的螢幕群組。
Shape 物件	Shape 物件為 Element 物件的子類別。相較於繪圖 API，Shape 物件對在「舞台」上處理或建立幾何的控制更精確。
SoundItem 物件	SoundItem 物件為 Item 物件的子類別。代表某個用來建立聲音的元件庫項目。
Stroke 物件	Stroke 物件包含筆畫的所有設定 (包括自訂設定)。

物件	說明
swfPanel 物件	swfPanel 物件代表 Windows 「SWF」面板。Windows 「SWF」面板是實作可從 Flash 編寫環境執行之應用程式的 SWF 檔。swfPanel 物件的陣列為 flash 物件 (fl.swfPanels) 的屬性。
SymbolInstance 物件	SymbolInstance 物件為 Instance 物件的子類別，代表影格中的元件。
SymbolItem 物件	SymbolItem 物件為 Item 物件的子類別。
Text 物件	Text 物件代表文件中的單一文字項目。
TextAttrs 物件	TextAttrs 物件包含所有可以套用於細部選取的文字屬性。此物件是 Text 物件的子類別。
TextRun 物件	TextRun 物件代表連續字元，所含特質與 TextAttrs 物件中的所有屬性皆相符。
Timeline 物件	Timeline 物件代表 Flash 時間軸，目前的文件可透過 fl.getDocumentDOM().getTimeline() 加以存取。
ToolObj 物件	ToolObj 物件代表 「工具」面板中的個別工具。
Tools 物件	Tools 物件可由 flash 物件 (fl.tools) 存取。
Vertex 物件	Vertex 物件為保留座標資料的部分形狀資料結構。
VideoItem 物件	VideoItem 物件為 Item 物件的子類別。
XMLUI 物件	XMLUI 物件可用來取得和設定 XMLUI 對話方塊的屬性，以及接受或取消特定屬性。

Flash 文件物件模型

Flash JavaScript API 的「Flash 文件物件模型」(DOM) 包含一組最上層函數 (請參閱第 13 頁「最上層函數和方法」) 以及兩個最上層物件 (FLfile 物件和 flash 物件 (fl))。Script 中保證這兩個物件都是可用的，因為當 Flash 編寫環境開啟時，它是一直存在的。如需詳細資訊，請參閱 FLfile 物件 和 flash 物件 (fl)。

在參照 Flash 物件時，可以使用 flash 或 fl。例如，若要關閉所有開啟的 FLA 檔，可以使用下列任何一個陳述式：

```
flash.closeAll();
fl.closeAll();
```

Flash 物件包含下列子物件：

物件	如何存取
actionsPanel 物件	請使用 fl.actionsPanel 來存取 actionsPanel 物件。此物件會對應至 Flash 編寫環境中的 「動作」面板。
compilerErrors 物件	請使用 fl.compilerErrors 來存取 compilerErrors 物件。此物件會對應至 Flash 編寫環境中的 「編譯器錯誤」面板。
componentsPanel 物件	請使用 fl.componentsPanel 來存取 componentsPanel 物件。此物件會對應至 Flash 編寫環境中的「組件」面板。
Document 物件	請使用 fl.documents 來擷取所有開啟文件的陣列；使用 fl.documents[index] 存取特定文件；使用 fl.getDocumentDOM() 存取目前文件 (焦點文件)。
drawingLayer 物件	請使用 fl.drawingLayer 來存取 drawingLayer 物件。
Math 物件	請使用 fl.Math 來存取 Math 物件。
outputPanel 物件	請使用 fl.outputPanel 來存取 outputPanel 物件。此物件會對應至 Flash 編寫環境中的 「輸出」面板。
presetPanel 物件	請使用 fl.presetPanel 來存取 presetPanel 物件。此物件會對應至「移動預設效果」面板 (「視窗 > 移動預設效果」)。

物件	如何存取
swfPanel 物件	請使用 fl.swfPanels 來存取 swfPanel 物件的陣列。這些物件會對應至 Windows 「SWF」面板。
Tools 物件	請使用 fl.tools 來存取 Tools 物件的陣列。
XMLUI 物件	請使用 fl.xmlui 來存取 XML 使用者介面 (XMLUI) 物件。XMLUI 物件可用來取得和設定 XMLUI 對話方塊的屬性。

Document 物件

fl.documents 屬性為最上層 flash 物件的重要屬性。此屬性包含 Document 物件的陣列，其中每個物件分別代表編寫環境中目前開啟的其中一個 FLA 檔。每個 Document 物件的屬性代表 FLA 檔可以包含的大多數元素。因此，大部分的 DOM 是由子物件和 Document 物件的屬性所組成。如需詳細資訊，請參閱 Document 物件。

例如，若要參照第一個開啟文件，請使用陳述式 flash.documents[0] 或 fl.documents[0]。第一個文件為編寫環境中，目前工作階段開啟的第一個 Flash 文件。在第一個開啟文件關閉時，其它開啟文件的索引將隨之遞減。

若要尋找特定文件的索引，請使用 flash.findDocumentIndex(nameOfDocument) 或 fl.findDocumentIndex(nameOfDocument)。請參閱 fl.findDocumentIndex()。

若要存取目前的焦點文件，請使用陳述式 flash.getDocumentDOM() 或 fl.getDocumentDOM()。請參閱 fl.getDocumentDOM()。後者為本文件中多數範例所使用的語法。

若要尋找 fl.documents 陣列中的特定文件，請在陣列中重複執行陳述式，並且測試每個文件的 document.name 屬性。請參閱 fl.documents 與 document.name。

前述表格中未列出的所有 DOM 物件 (請參閱：第 8 頁 「Flash 文件物件模型」)，則從 Document 物件存取。以存取文件的元件庫為例，您可以使用擷取文件庫物件的 document.library 屬性：

```
fl.getDocumentDOM().library
```

若要存取元件庫中的項目陣列，請使用 library.items 屬性：陣列中的每個元素都是一個 Item 物件：

```
fl.getDocumentDOM().library.items
```

若要存取元件庫中特定項目，請指定 library.items 陣列的成員：

```
fl.getDocumentDOM().library.items[0]
```

換句話說，Library 物件為 Document 物件的子物件，而 Item 物件又為 Library 物件的子物件。如需詳細資訊，請參閱 document.library、library 物件、library.itemslibrary.items，與 Item 物件。

指定動作的目標

除非特別指定，否則方法只會影響目前的焦點或選取範圍。例如，因為沒有特別指定物件，所以下列指令碼會將目前選取範圍的大小加倍：

```
fl.getDocumentDOM().scaleSelection(2, 2);
```

在某些情況下，您需要將動作目標特別設定為 Flash 文件中的目前選取項目。若要執行此項作業，請使用 document.selection 屬性所含的陣列 (請參閱 document.selection)。陣列中的第一個元素代表目前選取的項目，如以下範例所示：

```
var accDescription = fl.getDocumentDOM().selection[0].description;
```

以下指令碼將 「舞台」中儲存在元素陣列中的第一個元素 (而不是目前的選取範圍) 的大小加倍：

```
var element = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
if (element) {
    element.width = element.width*2;
    element.height = element.height*2;
}
```

您也可以重複「舞台」中的所有元素，並將寬度和高度增加指定長度，如以下範例所示：

```
var elementArray =
    fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
    for (var i=0; i < elementArray.length; i++) {
        var offset = 10;
        elementArray[i].width += offset;
        elementArray[i].height += offset;
    }
```

DOM 結構摘要

下列清單會以外框格式顯示 DOM 結構。在每一行開頭的編號代表物件的階層。例如，前面有「03」的物件為上一層「02」物件的子物件；以此類推，「02」物件則為再上一層「01」物件的子物件。

在某些情況下，物件可經由指定其父物件的屬性來取得。例如，document.timelines 屬性包含 Timeline 物件的陣列。下列綱要列有這些屬性的註解。

有些物件為其它物件的子類別，而不是其它物件的子系。屬於其它物件子類別的物件，除了具備父物件 (父類別) 的方法和屬性以外，本身也有方法和 / 或屬性。子類別在階層架構中的階層與其父類別相同。例如，Item 物件是 BitmapItem 物件的父類別。這些關係在下列綱要中說明：

```
01 Top-Level Functions and Methods
01 FLfile object
01 flash object (fl)
  02 compilerErrors object
  02 componentsPanel object
  02 Document object (fl.documents array)
    03 Filter object
    03 Matrix object
    03 Fill object
    03 Stroke object
    03 library object
      04 Item object (library.items array)
      04 BitmapItem object(subclass of Item object)
      04 folderItem object (subclass of Item object)
      04 fontItem object (subclass of Item object)
      04 SoundItem object (subclass of Item object)
      04 SymbolItem object (subclass of Item object)
      04 VideoItem object (subclass of Item object)
    03 Timeline object (document.timelines array)
      04 Layer object (timeline.layers array)
        05 Frame object (layer.frames array)
          06 Element object (frame.elements array)
            07 Matrix object (element.matrix)
          06 Instance object (abstract class, subclass of Element object)
          06 BitmapInstance object (subclass of Instance object)
          06 CompiledClipInstance object (subclass of Instance object)
          06 ComponentInstance object (subclass of SymbolInstance object)
            07 Parameter object (componentInstance.parameters array)
          06 SymbolInstance object (subclass of Instance object)
          06 Text object (subclass of Element object)
            07 TextRun object (text.textRuns array)
              08 TextAttrs object (textRun.textAttrs array)
          06 Shape object (subclass of Element object)
            07 Oval object
            07 Rectangle object
            07 Contour object (shape.contours array)
              08 HalfEdge object
                09 Vertex object
                09 Edge object
```

```
            07 Edge object (shape.edges array)
              08 HalfEdge object
                09 Vertex object
                09 Edge object
            07 Vertex object(shape.vertices array)
              08 HalfEdge object
                09 Vertex object
                09 Edge object
        05 Parameter object (screen.parameters array)
02 drawingLayer object
  03 Path object
    04 Contour object
02 Math object
02 outputPanel object
02 presetPanel object
  03 presetItem object(presetPanel.items array)
02 swfPanel object
02 Tools object (fl.tools array)
  03 ToolObj object (tools.toolObjs array)
02 XMLUI object
```

樣本實作

Adobe Flash Professional CS5 和 CS5.5 提供幾個範例 JSFL 實作。您可以檢閱並安裝這些檔案，藉此熟悉此 JavaScript API。這些樣本位於名為 Samples/ExtendingFlash 的資料夾中，此資料夾就在 www.adobe.com/go/learn_fl_samples_tw 的 Samples.zip 檔中。

樣本 **Shape** 命令

ExtendingFlash/Shape 資料夾內包含了名為 Shape.jsfl 的樣本 JavaScript API 指令碼 (請參閱上述的 「樣本實作」)。這個指令碼會在 「輸出」面板中顯示與形狀的輪廓相關的資訊。

安裝並執行 **Shape** 指令碼：

1 將 Shape.jsfl 檔複製到 Configuration/Commands 資料夾 (請參閱第 2 頁 「儲存 JSFL 檔」)。

2 在 Flash 文件 (FLA 檔) 中選取一個 Shape 物件。

3 選取 「命令 > 形狀」以執行這個指令碼。

取得樣本並設定濾鏡命令

ExtendingFlash/filtersGetSet 資料夾內包含了名為 filtersGetSet.jsfl 的樣本 JavaScript API 指令碼 (請參閱上述的 「樣本實作」)。這個指令碼會在選取的物件中加入濾鏡，並且顯示新增到 「輸出」面板中的濾鏡相關資訊。

安裝並執行 **filtersGetSet** 指令碼：

1 將 filtersGetSet.jsfl 檔複製到 Configuration/Commands 資料夾 (請參閱第 2 頁 「儲存 JSFL 檔」)。

2 在 Flash 文件 (FLA 檔) 中選取文字、影片片段或 button 物件。

3 選取 「命令 > filtersGetSet」以執行這個指令碼。

樣本 **PolyStar** 工具

ExtendingFlash/PolyStar 資料夾內包含了名為 PolyStar.jsfl 的樣本 JavaScript API 指令碼 (請參閱上述的 「樣本實作」)。

PolyStar.jsfl 會複製 PolyStar 工具；此工具位於 Flash 「工具」面板中。此指令碼會示範如何使用 JavaScript API 建立 PolyStar 工具，並且加入說明程式碼用途的詳細註解。請閱讀這個檔案，深入瞭解 JavaScript API 的應用方式。另外，也請閱讀 Tools 目錄中的 PolyStar.xml 檔，瞭解如何建立個人專屬的工具。

轉換成向量圖樣本面版

ExtendingFlash/TraceBitmapPanel 資料夾內包含了一組名為 TraceBitmap.fla 和 TraceBitmap.swf 的檔案 (請參閱上述的 「樣本實作」)。這些檔案說明如何設計及建立面板，以控制 Flash 的功能。它們也會說明如何使用 MMExecute() 函數，以從 ActionScript 指令碼呼叫 JavaScript 命令。

若要執行 **TraceBitmap** 樣本：

1. 如果 Flash 正在執行，請結束 Flash。
2. 將 TraceBitmap.swf 檔複製到 WindowSWF 資料夾，後者為 Configuration 資料夾的子目錄 (請參閱第 2 頁 「儲存 JSFL 檔」)。例如，在 Windows XP 中，資料夾位於 開機磁碟 \Documents and Settings\< 使用者 >\Local Settings\Application Data\Adobe\Flash CS5\< 語言 >\Configuration\WindowSWF 中。
3. 啟動 Flash。
4. 建立或開啟 Flash 文件 (FLA 檔)，然後將點陣圖或 JPEG 影像匯入此檔案。

   您可以使用 TraceBitmapPanel 資料夾中提供的 flower.jpg 檔或您所選擇的其它影像。
5. 選取好要匯入的影像後，請選取 「視窗 > 其它面板 > TraceBitmap」。
6. 按一下 「送出」。

   影像會轉換成一組形狀。

DLL 樣本

ExtendingFlash/dllSampleComputeSum 資料夾內包含了樣本 DLL 實作 (請參閱上述的 「樣本實作」)。如需有關建立 DLL 的詳細資訊，請參閱 第 489 頁 「C 語言層次擴充」。

第 2 章 最上層函數和方法

關於本節

本節將說明使用 Adobe Flash JavaScript 應用程式設計介面 (JavaScript API) 時可用的最上層函數和方法。如需有關在何處儲存 JavaScript API 檔的詳細資訊，請參閱第 2 頁 「儲存 JSFL 檔」。

全域方法

下列方法可從任何 JavaScript API 指令碼呼叫：

```
alert()
confirm()
prompt()
```

可擴充工具

下列函數僅可用在建立可擴充工具的指令碼中：

```
activate()
configureTool()
deactivate()
keyDown()
keyUp()
mouseDoubleClick()
mouseDown()
mouseMove()
mouseUp()
notifySettingsChanged()
setCursor()
```

activate()

適用版本

Flash MX 2004。

用法

```
function activate() {
    // statements
}
```

參數

無。

傳回值

無。

說明

函數：可擴充工具作用時 (也就是在 「工具」面板中選取工具時) 才可呼叫此函數。使用這個函數來執行工具所需求的任何初始化工作。

範例
當選取「工具」面板中的可擴充工具時，下列範例會設定 tools.activeTool 的值：

```
function activate() {
    var theTool = fl.tools.activeTool
}
```

請參閱
tools.activeTool

alert()

適用版本
Flash MX 2004。

用法
```
alert ( alertText )
```

參數
alertText 字串，指定「警告」對話方塊中的顯示訊息。

傳回值
無。

說明
方法：在強制回應「警告」對話方塊中顯示字串以及「確定」按鈕。

範例
以下範例在「警告」對話方塊中顯示「Process Complete」訊息：

```
alert("Process Complete");
```

請參閱
confirm(), prompt()

configureTool()

適用版本
Flash MX 2004。

用法
```
function configureTool() {
    // statements
}
```

參數
無。

傳回值
無。

說明
函數；開啟 Flash，並將可擴充工具載入至 「工具」面板時呼叫。請使用此函數設定 Flash 所需的任何工具相關資訊。

範例
以下示範兩種此函數可能的實作：

```
function configureTool() {
    theTool = fl.tools.activeTool;
    theTool.setToolName("myTool");
    theTool.setIcon("myTool.png");
    theTool.setMenuString("My Tool's menu string");
    theTool.setToolTip("my tool's tool tip");
    theTool.setOptionsFile( "mtTool.xml" );
}

function configureTool() {
    theTool = fl.tools.activeTool;
    theTool.setToolName("ellipse");
    theTool.setIcon("Ellipse.png");
    theTool.setMenuString("Ellipse");
    theTool.setToolTip("Ellipse");
    theTool.showTransformHandles( true );
}
```

confirm()

適用版本
Flash 8。

用法
```
confirm ( strAlert )
```

參數
strAlert 字串，指定 「警告」對話方塊中的顯示訊息。

傳回值
Boolean 值：如使用者按一下 「確定」，將傳回 true；如使用者按一下 「取消」，則傳回 false。

說明
方法：在強制回應 「警告」對話方塊中顯示字串以及 「確定」和 「取消」按鈕。

備註：如果沒有任何文件 (FLA 檔) 在開啟狀態，這個方法就會因錯誤條件而失敗。

範例
以下範例在 「警告」對話方塊中顯示 「Sort data?」訊息：

```
confirm("Sort data?");
```

請參閱
alert(), prompt()

deactivate()

適用版本
Flash MX 2004。

用法
```
function deactivate() {
    // statements
}
```

參數
無。

傳回值
無。

說明
函數：在停用可擴充工具時 (也就是作用中工具從此工具變更成另一個工具時) 呼叫。使用此函數執行工具所需進行的任何清理作業。

範例
下列範例會在工具停用時，於「輸出」面板中顯示訊息：

```
function deactivate() {
    fl.trace( "Tool is no longer active" );
}
```

keyDown()

適用版本
Flash MX 2004。

用法
```
function keyDown() {
    // statements
}
```

參數
無。

傳回值
無。

說明
函數：在可擴充工具有作用且使用者按下按鍵時呼叫。Script 應呼叫 tools.getKeyDown() 來判斷使用者按的是哪一個鍵。

範例
下列範例會在可擴充工具作用中且使用者按下按鍵時，顯示所按下的按鍵的相關資訊。

```
function keyDown() {
    fl.trace("key " + fl.tools.getKeyDown() + " was pressed");
}
```

請參閱
keyUp(), tools.getKeyDown()

keyUp()

適用版本
Flash MX 2004。

用法
```
function keyUp() {
    // statements
}
```

參數
無。

傳回值
無。

說明
函數：在可擴充工具有作用且使用者放開按鍵時呼叫。

範例
下列範例會在可擴充工具在作用中且使用者放開按鍵時，在「輸出」面板中顯示訊息。

```
function keyUp() {
    fl.trace("Key is released");
}
```

請參閱
keyDown()

mouseDoubleClick()

適用版本
Flash MX 2004。

用法

```
function mouseDoubleClick() {
    // statements
}
```

參數

無。

傳回值

無。

說明

函數；當可擴充工具作用中且使用者在 「舞台」上按兩下滑鼠按鈕時呼叫。

範例

下列範例會在可擴充工具在作用中且使用者按兩下滑鼠按鈕時，在 「輸出」面板中顯示訊息。

```
function mouseDoubleClick() {
    fl.trace("Mouse was double-clicked");
}
```

mouseDown()

適用版本

Flash MX 2004。

用法

```
function mouseDown( [ pt ] ) {
    // statements
}
```

參數

pt 點，指定按下按鈕時的滑鼠位置。它會在按下滑鼠按鈕時傳遞至函數。這個參數是選擇性參數。

傳回值

無。

說明

函數；在可擴充工具有作用且指標移到 「舞台」上並同時按下滑鼠按鈕時呼叫。

範例

下列範例顯示當可擴充工具在作用中時，應用這個函數的方法。第一個範例會在按下滑鼠按鈕時，在 「輸出」面板中顯示訊息。第二個範例顯示按下按鈕時，滑鼠位置的 x 座標和 y 座標。

```
function mouseDown() {
    fl.trace("Mouse button has been pressed");
}
function mouseDown(pt) {
    fl.trace("x = "+ pt.x+" :: y = "+pt.y);
}
```

mouseMove()

適用版本
Flash MX 2004。

用法
```
function mouseMove( [ pt ] ) {
    // statements
}
```

參數
pt 點，指定目前滑鼠位置。只要滑鼠移動，該點便會傳遞至函數，以便追蹤滑鼠的位置。若「舞台」為編輯中或使用原地編輯模式，點座標會與編輯物件相對應。否則，點座標會與「舞台」相對應。這個參數是選擇性參數。

傳回值
無。

說明
函數；每當可擴充工具在作用中而且滑鼠移到「舞台」上特定位置時呼叫。滑鼠按鈕可按下或放開。

範例
以下範例顯示如何使用此函數。第一個範例會在滑鼠移動時，在「輸出」面板中顯示訊息。第二個範例會在當滑鼠移動時，顯示其位置的 x 座標和 y 座標。

```
function mouseMove() {
    fl.trace("moving");
}

function mouseMove(pt) {
    fl.trace("x = "+ pt.x + " :: y = " + pt.y);
}
```

mouseUp()

適用版本
Flash MX 2004。

用法
```
function mouseUp() {
    // statements
}
```

參數
無。

傳回值
無。

說明
函數：當可擴充工具作用中且使用者在「舞台」上按下滑鼠按鈕而後放開時呼叫。

範例
下列範例會在可擴充工具在作用中且使用者放開滑鼠按鈕時，在「輸出」面板中顯示訊息。

```
function mouseUp() {
    fl.trace("mouse is up");
}
```

notifySettingsChanged()

適用版本
Flash MX 2004。

用法
```
function notifySettingsChanged() {
    // statements
}
```

參數
無。

傳回值
無。

說明
函數：在可擴充工具有作用，且使用者變更「屬性」檢測器中的選項時呼叫。您可利用 tools.activeTool 屬性來查詢此選項目前的值 (請參閱 tools.activeTool)。

範例
下列範例會在可擴充工具在作用中且使用者在「屬性」檢測器中變更選項時，在「輸出」面板中顯示訊息。

```
function notifySettingsChanged() {
    var theTool = fl.tools.activeTool;
    var newValue = theTool.myProp;
}
```

prompt()

適用版本
Flash MX 2004。

用法
```
prompt(promptMsg [,text])
```

參數
promptMsg　「提示」對話方塊中顯示的字串 (在 Mac OS X 中，限制為 256 個字元)。

text 選擇性字串，顯示為文字欄位預設值。

傳回值
若使用者按一下 「確定」，則傳回使用者輸入的字串；若使用者按一下 「取消」，則傳回 null。

說明
方法：在強制回應 「警告」對話方塊中顯示提示和選擇性文字，以及 「確定」和 「取消」按鈕。

範例
以下範例提示使用者輸入使用者名稱。若使用者輸入名稱並按一下 「確定」，則名稱會顯示於 「輸出」面板上。

```
var userName = prompt("Enter user name", "Type user name here");
fl.trace(userName);
```

請參閱
alert(), confirm()

setCursor()

適用版本
Flash MX 2004。

用法
```
function setCursor() {
    // statements
}
```

參數
無。

傳回值
無。

說明
函數：當可擴充工具在作用中且滑鼠移動以便讓指令碼設定自訂指標時呼叫。Script 應呼叫 tools.setCursor() 來指定要使用的指標。若需各指標對應整數值的清單，請參閱 tools.setCursor()。

範例
```
function setCursor() {
    fl.tools.setCursor( 1 );
}
```

第 3 章 actionsPanel 物件

適用版本
Flash CS3 Professional。

說明
actionsPanel 物件，代表目前顯示的 「動作」面板，為 flash 物件的屬性 (請參閱 fl.actionsPanel)。

方法摘要
下列方法可搭配 actionsPanel 物件使用：

方法	說明
actionsPanel.getClassForObject()	傳回指定變數的類別。
actionsPanel.getScriptAssistMode()	指定是否啟用指令碼助理模式。
actionsPanel.getSelectedText()	傳回目前在 「動作」面板中選取的文字。
actionsPanel.getText()	傳回 「動作」面板中的文字。
actionsPanel.hasSelection()	指定目前是否在 「動作」面板中選取任何文字。
actionsPanel.replaceSelectedText()	將目前選取的文字取代成指定的文字。
actionsPanel.setScriptAssistMode()	啟用或停用指令碼助理模式。
actionsPanel.setSelection()	在 「動作」面板中選取指定的字元組。
actionsPanel.setText()	清除 「動作」面板中的任何文字，然後加入指定的文字。

actionsPanel.getClassForObject()

適用版本
Flash CS3 Professional。

用法
```
actionsPanel.getClassForObject(ASvariableName)
```

參數
ASvariableName 代表 ActionScript 變數名稱的字串。

傳回值
字串，代表 ASvariableName 屬於其成員的類別。

說明
方法：傳回指定變數的類別，此變數必須在目前顯示的 「動作」面板中定義。此外，「動作」面板中的游標或選取文字必須位於變數定義之後。

範例

如果游標位於 「動作」面板中的陳述式 var myVar:ActivityEvent; 之後，下列範例將會顯示指派給 myVar 變數的類別。

```
// Place the following code in the Actions panel,
// and position the cursor somewhere after the end of the line
var myVar:ActivityEvent;
// Place the following code in the JSFL file
var theClass = fl.actionsPanel.getClassForObject("myVar");
fl.trace(theClass); // traces: "ActivityEvent"
```

actionsPanel.getScriptAssistMode()

適用版本

Flash CS3 Professional。

用法

```
actionsPanel.getScriptAssistMode()
```

參數

無。

傳回值

Boolean 值，指定指令碼助理模式為啟用 (true) 或停用 (false)。

說明

方法：指定是否啟用指令碼助理模式。

範例

如果指令碼助理模式沒有啟用，下列範例將會顯示一則訊息。

```
mAssist = fl.actionsPanel.getScriptAssistMode();
if (!mAssist) {
    alert("For more guidance when writing ActionScript code, try Script Assist mode");
}
```

請參閱

actionsPanel.setScriptAssistMode()

actionsPanel.getSelectedText()

適用版本

Flash CS3 Professional。

用法

```
actionsPanel.getSelectedText()
```

參數

無。

傳回值
字串，其中包含目前在 「動作」面板中選取的文字。

說明
方法；傳回目前在 「動作」面板中選取的文字。

範例
下列範例將會顯示目前在 「動作」面板中選取的文字。

```
var apText = fl.actionsPanel.getSelectedText();
fl.trace(apText);
```

請參閱
actionsPanel.getText()、actionsPanel.hasSelection()、actionsPanel.replaceSelectedText()、actionsPanel.setSelection()

actionsPanel.getText()

適用版本
Flash CS3 Professional。

用法

```
actionsPanel.getText()
```

參數
無。

傳回值
字串，其中包含 「動作」面板中的所有文字。

說明
方法；傳回 「動作」面板中的文字。

範例
下列範例將會顯示 「動作」面板中的文字。

```
var apText = fl.actionsPanel.getText();
fl.trace(apText);
```

請參閱
actionsPanel.getSelectedText()、actionsPanel.setText()

actionsPanel.hasSelection()

適用版本
Flash CS3 Professional。

用法
```
actionsPanel.hasSelection()
```

參數
無。

傳回值
Boolean 值，指定是 (true) 否 (false) 已在 「動作」面板中選取任何文字。

說明
方法：指定目前是否在 「動作」面板中選取任何文字。

範例
下列範例將會顯示目前在 「動作」面板中選取的文字。如果沒有選取任何文字，它就會顯示 「動作」面板中的所有文字。

```
if (fl.actionsPanel.hasSelection()) {
    var apText = fl.actionsPanel.getSelectedText();
}
else {
    var apText = fl.actionsPanel.getText();
}
fl.trace(apText);
```

請參閱
actionsPanel.getSelectedText()、actionsPanel.getText()、actionsPanel.replaceSelectedText()、actionsPanel.setSelection()

actionsPanel.replaceSelectedText()

適用版本
Flash CS3 Professional。

用法
```
actionsPanel.replaceSelectedText(replacementText)
```

參數
replacementText 字串，代表要取代 「動作」面板中選取文字的文字。

傳回值
如果找到 「動作」面板，則傳回 Boolean 值 true；否則會傳回 false。

說明
方法：將目前選取的文字取代成 replacementText 中指定的文字。如果 replacementText 包含的字元比選取的文字還多，則選取文字後面的字元就會接在 replacementText 後面；也就是說，這些字元不會被覆寫。

範例
下列範例將會取代目前在 「動作」面板中選取的文字。

```
if (fl.actionsPanel.hasSelection()) {
    fl.actionsPanel.replaceSelectedText("// © 2006 Adobe Inc.");
}
```

請參閱
actionsPanel.getSelectedText()、actionsPanel.hasSelection()、actionsPanel.setSelection()、actionsPanel.setText()

actionsPanel.setScriptAssistMode()

適用版本
Flash CS3 Professional。

用法
```
actionsPanel.setScriptAssistMode(bScriptAssist)
```

參數
bScriptAssist Boolean 值，指定要啟用或停用指令碼助理模式。

傳回值
Boolean 值，指定指令碼助理模式已成功啟用或停用。

說明
方法：啟用或停用指令碼助理模式。

範例
下列範例將會切換指令碼助理模式的狀態。

```
fl.trace(fl.actionsPanel.getScriptAssistMode());
if (fl.actionsPanel.getScriptAssistMode()){
    fl.actionsPanel.setScriptAssistMode(false);
}
else {
    fl.actionsPanel.setScriptAssistMode(true);
}
fl.trace(fl.actionsPanel.getScriptAssistMode());
```

請參閱
actionsPanel.getScriptAssistMode()

actionsPanel.setSelection()

適用版本
Flash CS3 Professional。

用法
```
actionsPanel.setSelection(startIndex, numberOfChars)
```

參數
startIndex 從零開始的整數，指定要選取的第一個字元。

numberOfChars 整數，指定要選取多少字元。

傳回值
Boolean 值，指定是 (true) 否 (false) 可以選取要求的字元。

說明
方法；在 「動作」面板中選取指定的字元組。

範例
下列範例會將 「動作」面板中的字元 「2006」取代成指定的文字。

```
// Type the following as the first line in the Actions panel
// 2006 - Addresses user request 40196
// Type the following in the JSFL file
fl.actionsPanel.setSelection(3,4);
fl.actionsPanel.replaceSelectedText("// Last updated: 2007");
```

請參閱
actionsPanel.getSelectedText()、actionsPanel.hasSelection()、actionsPanel.replaceSelectedText()

actionsPanel.setText()

適用版本
Flash CS3 Professional。

用法
```
actionsPanel.setText(replacementText)
```

參數
replacementText 字串，代表要在 「動作」面板中取代的文字。

傳回值
如果指定的文字已放置在 「動作」面板中，則傳回 Boolean 值 true；否則會傳回 false。

說明
方法；清除 「動作」面板中的任何文字，然後加入 replacementText 中指定的文字。

範例
下列範例會將目前在 「動作」面板中的文字取代成指定的文字。

```
fl.actionsPanel.setText("// Deleted this code - no longer needed");
```

請參閱
actionsPanel.getText()、actionsPanel.replaceSelectedText()

第 4 章 BitmapInstance 物件

繼承 Element 物件 > Instance 物件 > BitmapInstance 物件

適用版本
Flash MX 2004。

說明
BitmapInstance 物件為 Instance 物件的子類別，代表影格中的點陣圖 (請參閱 Instance 物件)。

方法摘要
除了 Instance 物件方法，BitmapInstance 物件還可以搭配下列方法使用：

方法	說明
bitmapInstance.getBits()	可讓您從點陣圖取出位元、處理位元，再將位元傳回 Flash 以建立點陣圖特效。
bitmapInstance.setBits()	設定現有點陣圖元素的位元。

屬性摘要
除了 Instance 物件屬性，BitmapInstance 物件還可以搭配下列屬性使用：

屬性	說明
bitmapInstance.hPixels	唯讀；代表點陣圖寬度的整數，以像素為單位。
bitmapInstance.vPixels	唯讀；代表點陣圖高度的整數，以像素為單位。

bitmapInstance.getBits()

適用版本
Flash MX 2004。

用法
```
bitmapInstance.getBits()
```

參數
無。

傳回值
物件，包含 width、height、depth、bits 及 cTab (點陣圖有色彩表時) 屬性。bits 元素為位元組陣列。cTab 元素為顏色數值陣列，格式為 "#RRGGBB"。陣列長度為色表長度。

位元組陣列僅在 DLL 或共享元件庫參考時才有意義。一般只會在建立可擴充工具或特效時才使用到。如需有關建立搭配 Flash JavaScript 使用的 DLL 的詳細資訊，請參閱：第 489 頁 「C 語言層次擴充」

說明
方法；可讓您從點陣圖取出位元、處理位元，再將點陣圖傳回 Flash 以建立點陣圖特效。

範例
下列程式碼將建立目前選取物件的參考，以測試物件是否為點陣圖，並追蹤點陣圖的高度、寬度和位元深度：

```
var isBitmap = fl.getDocumentDOM().selection[0].instanceType;
if(isBitmap == "bitmap"){
    var bits = fl.getDocumentDOM().selection[0].getBits();
    fl.trace("height = " + bits.height);
    fl.trace("width = " + bits.width);
    fl.trace("depth = " + bits.depth);
}
```

請參閱
bitmapInstance.setBits()

bitmapInstance.hPixels

適用版本
Flash MX 2004。

用法
```
bitmapInstance.hPixels
```

說明
唯讀屬性；代表點陣圖寬度的整數 — 也就是水平的像素數目。

範例
下列程式碼擷取以像素為單位的點陣圖寬度。

```
// Get the number of pixels in the horizontal dimension.
var bmObj = fl.getDocumentDOM().selection[0];
var isBitmap = bmObj.instanceType;
if(isBitmap == "bitmap"){
    var numHorizontalPixels = bmObj.hPixels;
}
```

請參閱
bitmapInstance.vPixels

bitmapInstance.setBits()

適用版本
Flash MX 2004。

用法
```
bitmapInstance.setBits(bitmap)
```

參數
bitmap 包含 height、width、depth、bits 和 cTab 屬性的物件。height、width 和 depth 屬性為整數。bits 屬性為位元組陣列。點陣圖位元深度不超過 8 時，才需要 cTab 屬性；這個字串代表顏色數值，格式為 "#RRGGBB"。

備註：位元組陣列僅在外部元件庫參考時才有意義。一般只會在建立可擴充工具或特效時才使用到。

傳回值
無。

說明
方法；設定現有點陣圖元素的位元。可讓您從點陣圖取出位元、處理位元，再將點陣圖傳回 Flash 建立點陣圖特效。

範例
下列程式碼會測試目前選取範圍是否為點陣圖，然後將點陣圖高度設定為 150 像素：

```
var isBitmap = fl.getDocumentDOM().selection[0].instanceType;
if(isBitmap == "bitmap"){
    var bits = fl.getDocumentDOM().selection[0].getBits();
    bits.height = 150;
    fl.getDocumentDOM().selection[0].setBits(bits);
}
```

請參閱
bitmapInstance.getBits()

bitmapInstance.vPixels

適用版本
Flash MX 2004。

用法
```
bitmapInstance.vPixels
```

說明
唯讀屬性；代表點陣圖高度的整數 — 也就是垂直的像素數目。

範例
下列程式碼將取得點陣圖的高度 (以像素為單位)。

```
// Get the number of pixels in the vertical dimension.
var bmObj = fl.getDocumentDOM().selection[0];
var isBitmap = bmObj.instanceType;
if(isBitmap == "bitmap"){
    var numVerticalPixels = bmObj.vPixels;
}
```

請參閱
bitmapInstance.hPixels

第 5 章 BitmapItem 物件

繼承　Item 物件 > BitmapItem 物件

適用版本
Flash MX 2004。

說明
BitmapItem 物件是指文件元件庫中的點陣圖。BitmapItem 物件為 Item 物件的子類別 (請參閱 Item 物件)。

屬性摘要
除了 Item 物件 屬性，BitmapItem 物件還有下列屬性：

屬性	說明
bitmapItem.allowSmoothing	Boolean 值，指定是否要讓點陣圖平滑化。
bitmapItem.compressionType	字串，決定套用至點陣圖的影像壓縮類型。
bitmapItem.fileLastModifiedDate	在 1970 年 1 月 1 日和原始檔案修改日期之間經過的秒數。
bitmapItem.originalCompressionType	指定項目是否已匯入為 JPEG 檔。
bitmapItem.sourceFileExists	指定匯入至元件庫的檔案是否仍然存在於匯入來源位置。
bitmapItem.sourceFileIsCurrent	指定元件庫項目的檔案修改日期是否與磁碟上已匯入之檔案的修改日期相同。
bitmapItem.sourceFilePath	已匯入至元件庫之檔案的路徑和名稱。
bitmapItem.useDeblocking	指定是否啟用消除馬賽克。
bitmapItem.useImportedJPEGQuality	Boolean 值，指定是否使用預設匯入的 JPEG 品質。

方法摘要
除了 Item 物件屬性，BitmapItem 物件還有下列方法：

方法	說明
bitmapItem.exportToFile()	將指定的項目匯出至 PNG 或 JPG 檔。

bitmapItem.allowSmoothing

適用版本
Flash MX 2004。

用法
```
bitmapItem.allowSmoothing
```

說明
屬性：指定是否允許 (true) 或不允許 (false) 點陣圖平滑化的 Boolean 值。

範例
下列程式碼將目前文件的元件庫中，第一個項目的 allowSmoothing 屬性設定為 true：

```
fl.getDocumentDOM().library.items[0].allowSmoothing = true;
alert(fl.getDocumentDOM().library.items[0].allowSmoothing);
```

bitmapItem.compressionType

適用版本
Flash MX 2004。

用法
```
bitmapItem.compressionType
```

說明
屬性；決定套用至點陣圖的影像壓縮類型的字串。可接受的值為 "photo" 或 "lossless"。若 bitmapItem.useImportedJPEGQuality 的值是 false，"photo" 對應的是 JPEG 格式 0 至 100 的品質；若 bitmapItem.useImportedJPEGQuality 是 true，"photo" 對應的則是使用預設的文件品質數值的 JPEG 格式。值 "lossless" 則對應至 GIF 或 PNG 格式 (請參閱 bitmapItem.useImportedJPEGQuality)。

範例
下列程式碼將目前文件的元件庫中，第一個項目的 compressionType 屬性設定為 "photo"：

```
fl.getDocumentDOM().library.items[0].compressionType = "photo";
alert(fl.getDocumentDOM().library.items[0].compressionType);
```

bitmapItem.exportToFile()

適用版本
Flash CS4 Professional。

用法
```
bitmapItem.exportToFile(fileURI)
```

參數
fileURI 字串；指定匯出檔案的路徑和名稱，表示為 file:/// URI。

傳回值
如果檔案成功匯出，就會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；將指定項目匯出至 PNG 或 JPG 檔。

範例
假設元件庫中的第一個項目是點陣圖項目，下列程式碼會將它匯出為 JPG 檔：

```
var imageFileURL = "file:///C|/exportTest/out.jpg";
var libItem = fl.getDocumentDOM().library.items[0];
libItem.exportToFile(imageFileURL);
```

bitmapItem.fileLastModifiedDate

適用版本
Flash CS4 Professional。

用法
```
bitmapItem.fileLastModifiedDate
```

說明
唯讀屬性：包含十六進位數字的字串，代表在 1970 年 1 月 1 日和匯入至元件庫的原始檔案修改日期之間經過的秒數。如果檔案已不存在，此值為「00000000」。

範例
假設元件庫中的第一個項目是點陣圖項目，下列程式碼會將它顯示為十六進位數字 (如上述)。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

請參閱
bitmapItem.sourceFileExists、bitmapItem.sourceFileIsCurrent、bitmapItem.sourceFilePath、FLfile.getModificationDate()

bitmapItem.originalCompressionType

適用版本
Flash CS4 Professional。

用法
```
bitmapItem.originalCompressionType
```

說明
唯讀屬性：字串，指定特定項目是否已匯入為 JPEG 檔。此屬性的可能值為 "photo" (JPEG 檔) 和 "lossless" (未壓縮的檔案類型，如 GIF 和 PNG)。

範例
假設元件庫中的第一個項目是點陣圖項目，下列程式碼會在檔案於元件庫中匯入為 JPEG 檔時顯示「photo」，否則顯示「lossless」：

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("Imported compression type = "+ libItem.originalCompressionType);
```

請參閱
bitmapItem.compressionType

bitmapItem.quality

適用版本
Flash MX 2004。

用法
```
bitmapItem.quality
```

說明
屬性；指定點陣圖品質的整數。若要使用預設的文件品質，請指定 -1；否則，請指定 0 到 100 的整數。僅適用於 JPEG 壓縮。

範例
下列程式碼將目前文件的元件庫中，第一個項目的 quality 屬性設定為 65：

```
fl.getDocumentDOM().library.items[0].quality = 65;
alert(fl.getDocumentDOM().library.items[0].quality);
```

bitmapItem.sourceFileExists

適用版本
Flash CS4 Professional。

用法
```
bitmapItem.sourceFileExists
```

說明
唯讀屬性；如果匯入至元件庫中的檔案仍然位於匯入來源位置，會傳回 Boolean 值 true，否則傳回 false。

範例
假設元件庫中的第一個項目是點陣圖項目，下列程式碼會在匯入至元件庫中的檔案仍然存在時顯示 "true"。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

請參閱
bitmapItem.sourceFileIsCurrent,

bitmapItem.sourceFilePath

bitmapItem.sourceFileIsCurrent

適用版本
Flash CS4 Professional。

用法
```
bitmapItem.sourceFileIsCurrent
```

說明
唯讀屬性：如果元件庫項目的檔案修改日期和磁碟上已匯入的原始檔案的修改日期相同，會傳回 Boolean 值 true，否則傳回 false。

範例
假設元件庫中的第一個項目是點陣圖項目，下列程式碼會在匯入的原始檔案自匯入後即未曾在磁碟上修改時顯示 "true"。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

請參閱
bitmapItem.fileLastModifiedDate、bitmapItem.sourceFilePath

bitmapItem.sourceFilePath

適用版本
Flash CS4 Professional。

用法
```
bitmapItem.sourceFilePath
```

說明
唯讀屬性：字串，代表已匯入至元件庫之檔案的路徑和名稱，表示方式為 file:/// URI。

範例
下列範例會顯示元件庫中任何類型為 "bitmap" 的項目之名稱和來源檔案路徑：

```
for (idx in fl.getDocumentDOM().library.items) {
if (fl.getDocumentDOM().library.items[idx].itemType == "bitmap") {
    var myItem = fl.getDocumentDOM().library.items[idx];
    fl.trace(myItem.name + " source is " + myItem.sourceFilePath);
    }
}
```

請參閱
bitmapItem.sourceFileExists

bitmapItem.useDeblocking

適用版本
Flash CS4 Professional。

用法
```
bitmapItem.useDeblocking
```

說明
屬性；Boolean 值，指定啟用 (true) 或停用 (false) 消除馬賽克。

範例

假設元件庫中的第一個項目是點陣圖項目，下列程式碼會為項目啟用消除馬賽克：

```
var libItem = fl.getDocumentDOM().library.items[0];
libItem.useDeblocking = true;
```

bitmapItem.useImportedJPEGQuality

適用版本

Flash MX 2004。

用法

```
bitmapItem.useImportedJPEGQuality
```

說明

屬性；Boolean 值，指定會使用 (true) 或不使用 (false) 預設匯入的 JPEG 品質。僅適用於 JPEG 壓縮。

範例

下列程式碼將目前文件的元件庫中，第一個項目的 useImportedJPEGQuality 屬性設定為 true：

```
fl.getDocumentDOM().library.items[0].useImportedJPEGQuality = true;
alert(fl.getDocumentDOM().library.items[0].useImportedJPEGQuality);
```

第 6 章 CompiledClipInstance 物件

繼承　Element 物件 > Instance 物件 > CompiledClipInstance 物件

適用版本
Flash MX 2004。

說明
CompiledClipInstance 物件為 Instance 物件的子類別。它基本上為影片片段的實體，而且已轉換成編譯後的影片片段元件庫項目 (請參閱 Instance 物件)。

屬性摘要
除了 Instance 物件的屬性，CompiledClipInstance 物件還有下列屬性：

屬性	說明
compiledClipInstance.accName	字串；等於 「輔助功能」面板中的 「名稱」欄位。
compiledClipInstance.actionScript	代表此實體 ActionScript 的字串；相等於 symbolInstance.actionScript。
compiledClipInstance.description	字串；等於 「輔助功能」面板的 「說明」欄位。
compiledClipInstance.forceSimple	啟用和停用可存取物件子系的 Boolean 值。
compiledClipInstance.shortcut	字串；等於 「輔助功能」面板的 「快速鍵」欄位。
compiledClipInstance.silent	Boolean 值；啟用和停用物件的存取；等於 「輔助功能」面板 「讓物件支援輔助功能」設定的反轉邏輯。
compiledClipInstance.tabIndex	整數；等於 「輔助功能」面板中的 「定位鍵索引」欄位。

compiledClipInstance.accName

適用版本
Flash MX 2004。

用法
```
compiledClipInstance.accName
```

說明
屬性；字串；等於 「輔助功能」面板中的 「名稱」欄位。螢幕朗讀程式將大聲唸出物件名稱來識別物件。

範例
下列範例取得並設定第一個選取物件的輔助功能名稱：

```
// Get the name of the object.
var theName = fl.getDocumentDOM().selection[0].accName;
// Set the name of the object.
fl.getDocumentDOM().selection[0].accName = 'Home Button';
```

compiledClipInstance.actionScript

適用版本
Flash MX 2004。

用法
```
compiledClipInstance.actionScript
```

說明
屬性；代表實體 ActionScript 的字串；相等於 symbolInstance.actionScript。

範例
下列程式碼將 ActionScript 指定給指定的元素：

```
// Assign some ActionScript to a specified Button compiled clip instance.
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0]
    .actionScript = "on(click) {trace('button is clicked');}";
// Assign some ActionScript to the currently selected Button compiled clip instance.
fl.getDocumentDOM().selection[0].actionScript =
    "on(click) {trace('button is clicked');}";
```

compiledClipInstance.description

適用版本
Flash MX 2004。

用法
```
compiledClipInstance.description
```

說明
屬性；字串；等於 「輔助功能」面板中的 「說明」欄位。這項說明是由螢幕朗讀程式唸出。

範例
下列範例示範如何取得和設定 description 屬性：

```
// Get the description of the current selection.
var theDescription = fl.getDocumentDOM().selection[0].description;
// Set the description of the current selection.
fl.getDocumentDOM().selection[0].description =
    "This is compiled clip number 1";
```

compiledClipInstance.forceSimple

適用版本
Flash MX 2004。

用法
```
compiledClipInstance.forceSimple
```

說明

屬性；啟用和停用可存取物件子系的 Boolean 值。等於 「輔助功能」面板 「讓子物件支援輔助功能」設定的反向邏輯。如果 forceSimple 為 true，則等於未選取 「讓子物件支援輔助功能」選項。如果 forceSimple 為 false，則等於選取 「讓子物件支援輔助功能」選項。

範例

下列範例示範如何取得和設定 forceSimple 屬性：

```
// Query if the children of the object are accessible.
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;
// Allow the children of the object to be accessible.
fl.getDocumentDOM().selection[0].forceSimple = false;
```

compiledClipInstance.shortcut

適用版本

Flash MX 2004。

用法

```
compiledClipInstance.shortcut
```

說明

屬性；字串；等於 「輔助功能」面板中的 「快速鍵」欄位。這項快速鍵是由螢幕朗讀程式唸出。動態文字欄位無法使用此屬性。

範例

下列範例示範如何取得和設定 shortcut 屬性：

```
// Get the shortcut key of the object.
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
// Set the shortcut key of the object.
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+I";
```

compiledClipInstance.silent

適用版本

Flash MX 2004。

用法

```
compiledClipInstance.silent
```

說明

屬性；啟用和停用物件存取的 Boolean 值；等於 「輔助功能」面板 「讓物件支援輔助功能」設定的反轉邏輯。也就是說，如果 silent 為 true，則未選取 「讓物件支援輔助功能」。如果 silent 為 false，則選取 「讓物件支援輔助功能」。

範例

下列範例示範如何取得和設定 silent 屬性：

```
// Query if the object is accessible.
var isSilent = fl.getDocumentDOM().selection[0].silent;
// Set the object to be accessible.
fl.getDocumentDOM().selection[0].silent = false;
```

compiledClipInstance.tabIndex

適用版本

Flash MX 2004。

用法

```
compiledClipInstance.tabIndex
```

說明

屬性；相等於「輔助功能」面板中「定位鍵索引」欄位的整數。建立在使用者按下 Tab 鍵時存取物件的定位鍵順序。

範例

下列範例示範如何取得和設定 tabIndex 屬性：

```
// Get the tabIndex of the object.
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;
// Set the tabIndex of the object.
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

第 7 章 compilerErrors 物件

適用版本
Flash CS3 Professional。

說明
compilerErrors 物件 (代表 「編譯器錯誤」面板) 為 flash 物件 (fl) 的屬性，可以由 fl.compilerErrors 存取 (請參閱 flash 物件 (fl))。

方法摘要
compilerErrors 物件可搭配使用以下方法：

方法	說明
compilerErrors.clear()	清除 「編譯器錯誤」面板的內容。
compilerErrors.save()	將 「編譯器錯誤」面板的內容儲存至本機的文字檔。

compilerErrors.clear()

適用版本
Flash CS3 Professional。

用法
```
compilerErrors.clear()
```

參數
無。

傳回值
無。

說明
方法：清除 「編譯器錯誤」面板的內容。

範例
下列範例會清除 「編譯器錯誤」面板的內容：

```
fl.compilerErrors.clear();
```

請參閱
compilerErrors.save()

compilerErrors.save()

適用版本
Flash CS3 Professional。

用法
```
compilerErrors.save(fileURI [, bAppendToFile [, bUseSystemEncoding]])
```

參數
fileURI 字串 (表示為 file:/// URI)，指定已儲存檔案的檔案名稱。如果 fileURI 已經存在，而且您尚未針對 bAppendToFile 指定 true 值，就會覆寫 fileURI 而不提出警告。

bAppendToFile 選擇性的 Boolean 值，指定 「編譯器錯誤」面板的內容是 (true) 否 (false) 應該附加至 fileURI。預設值為 false。

bUseSystemEncoding 選擇性 Boolean 值，指定是否要使用系統編碼來儲存 「編譯器錯誤」面板文字。如果這個值為 false (預設值)，系統就會使用 UTF-8 編碼來儲存 「編譯器錯誤」面板文字，而且 「位元組順序標記」字元會位於文字的開頭。預設值為 false。

傳回值
無。

說明
方法：將 「編譯器錯誤」面板的內容儲存至本機的文字檔。

範例
下列範例會將 「編譯器錯誤」面板的內容儲存至 C:\tests 資料夾中名為 errors.log 的檔案：

```
fl.compilerErrors.save("file:///c|/tests/errors.log");
```

請參閱
compilerErrors.clear()

第 8 章 ComponentInstance 物件

繼承　Element 物件 > Instance 物件 > SymbolInstance 物件 > ComponentInstance 物件

適用版本
Flash MX 2004。

說明
ComponentInstance 物件為 SymbolInstance 物件的子類別，代表影格中的組件 (請參閱 SymbolInstance 物件)。

屬性摘要
除了 SymbolInstance 物件的所有屬性，ComponentInstance 物件還具有下列屬性：

屬性	說明
componentInstance.parameters	唯讀；包含 ActionScript 2.0 屬性的陣列，可從組件 「屬性」檢測器存取。

componentInstance.parameters

適用版本
Flash MX 2004。

用法
```
componentInstance.parameters
```

說明
唯讀屬性；包含 ActionScript 2.0 屬性的陣列，可從組件 「屬性」檢測器存取。請參閱 Parameter 物件。

範例
下列範例示範如何取得和設定 parameters 屬性：

```
var parms = fl.getDocumentDOM().selection[0].parameters;
parms[0].value = "some value";
```

請參閱
Parameter 物件

第 9 章 componentsPanel 物件

適用版本
Flash MX 2004。

說明
componentsPanel 物件，代表「組件」面板，為 flash 物件 (fl) 的屬性，可以由 fl.componentsPanel 存取 (請參閱 flash 物件 (fl))。

方法摘要
componentsPanel 物件可以搭配下列方法使用：

方法	說明
componentsPanel.addItemToDocument()	增加指定組件至指定位置上的文件。
componentsPanel.reload()	重新整理 「組件」面板的組件清單。

componentsPanel.addItemToDocument()

適用版本
Flash MX 2004。

用法
```
componentsPanel.addItemToDocument(position, categoryName, componentName)
```

參數
position 指定增加組件的位置點 (例如，{x:0, y:100})。指定相對於組件中心點的 position，並非相對於組件的註冊點 (亦稱為「原點」或「零點」)。

categoryName 指定組件類別 (例如，"Data") 名稱的字串。有效的類別名稱列示於 「組件」面板中。

componentName 字串，指定特定類別 (例如，"WebServiceConnector") 中的組件名稱。有效的組件名稱列示於 「組件」面板中。

傳回值
無。

說明
增加指定組件至指定位置上的文件。

範例
下列範例示範此方法的一些使用方式：

```
fl.componentsPanel.addItemToDocument({x:0, y:0}, "User Interface", "CheckBox");
fl.componentsPanel.addItemToDocument({x:0, y:100}, "Data", "WebServiceConnector");
fl.componentsPanel.addItemToDocument({x:0, y:200}, "User Interface", "Button");
```

componentsPanel.reload()

適用版本
Flash 8。

用法
```
componentsPanel.reload()
```

參數
無。

傳回值
如果重新整理 「組件」面板清單，會顯示 true 的 Boolean 值，否則會顯示 false。

說明
方法；重新整理 「組件」面板的組件清單。

範例
下列範例會重新整理 「組件」面板：

```
fl.componentsPanel.reload();
```

第 10 章 Contour 物件

適用版本
Flash MX 2004。

說明
Contour 物件代表在形狀邊界上不完整邊緣的封閉路徑。

方法摘要
Contour 物件可以搭配下列方法使用：

方法	說明
contour.getHalfEdge()	傳回選取範圍的輪廓上的 HalfEdge 物件。

屬性摘要
Contour 物件可以搭配下列屬性使用：

屬性	說明
contour.fill	Fill 物件。
contour.interior	唯讀；如果輪廓包含區域，則值為 true，否則為 false。
contour.orientation	唯讀；指定輪廓方向的整數。

contour.fill

適用版本
Flash CS4 Professional。

用法
contour.fill

說明
屬性；為 Fill 物件 .

範例
假設您已選取有填色的輪廓，下列範例會在 「輸出」面板中顯示輪廓的填色顏色：

```
var insideContour = fl.getDocumentDOM().selection[0].contours[1];
var insideFill = insideContour.fill;
fl.trace(insideFill.color);
```

contour.getHalfEdge()

適用版本
Flash MX 2004。

用法
contour.getHalfEdge()

參數
無。

傳回值
HalfEdge 物件。

說明
方法：傳回選取範圍的輪廓上的 HalfEdge 物件。

範例
本範例會詳細檢查選取形狀的所有輪廓，並在「輸出」面板中顯示頂點的座標：

```
// with a shape selected

var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;
var contourCount = 0;
for (i=0;i<contourArray.length;i++)
{
    var contour = contourArray[i];
    contourCount++;
    var he = contour.getHalfEdge();

    var iStart = he.id;
    var id = 0;
    while (id != iStart)
    {
        // Get the next vertex.
        var vrt = he.getVertex();


        var x = vrt.x;
        var y = vrt.y;
        fl.trace("vrt: " + x + ", " + y);

        he = he.getNext();
        id = he.id;
    }
}
elt.endEdit();
```

contour.interior

適用版本
Flash MX 2004。

用法
contour.interior

說明
唯讀屬性：若輪廓包含區域，則值為 true；否則為 false。

範例
本範例會詳細檢查選取形狀的所有輪廓，並在 「輸出」面板中顯示每個輪廓的 interior 屬性值：

```
var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;

var contourCount = 0;
for (i=0;i<contourArray.length;i++) {
    var contour = contourArray[i];
    fl.trace("Next Contour, interior:" + contour.interior );
    contourCount++;
}
elt.endEdit();
```

contour.orientation

適用版本
Flash MX 2004。

用法
contour.orientation

說明
唯讀屬性：指定輪廓方向的整數。如為逆時針方向，則整數值為 -1；順時針則為 1；無區域的輪廓則為 0。

範例
下列範例將移至選取形狀的所有輪廓，並在 「輸出」面板上顯示每個輪廓的 orientation 屬性值：

```
var elt = fl.getDocumentDOM().selection[0];
elt.beginEdit();

var contourArray = elt.contours;

var contourCount = 0;
for (i=0;i<contourArray.length;i++) {
    var contour = contourArray[i];
    fl.trace("Next Contour, orientation:" + contour.orientation);
    contourCount++;
}
elt.endEdit();
```

第 11 章 Document 物件

適用版本
Flash MX 2004。

說明
Document 物件代表「舞台」。也就是說，只有 FLA 檔被視為文件。若要傳回目前文件的 Document 物件，請使用 fl.getDocumentDOM()。

方法摘要
Document 物件可以搭配下列方法使用：

方法	說明
document.addDataToDocument()	使用文件儲存指定資料。
document.addDataToSelection()	使用選取物件儲存指定資料。
document.addFilter()	套用濾鏡至選取物件。
document.addItem()	從任何開啟文件或元件庫加入項目至指定的 Document 物件。
document.addNewLine()	在兩點之間加入新的路徑。
document.addNewOval()	在指定的矩形邊界中加入新的 Oval 物件。
document.addNewPrimitiveOval()	加入符合指定邊界的新基本橢圓形。
document.addNewPrimitiveRectangle()	加入符合指定邊界的新基本矩形。
document.addNewPublishProfile()	加入新的發佈描述檔，並讓它成為目前的發佈描述檔。
document.addNewRectangle()	加入新的矩形或圓角矩形，讓它符合指定的邊界。
document.addNewScene()	加入新的場景 (Timeline 物件) 做為目前選取場景的下一個場景，並讓新場景成為目前選取的場景。
document.addNewText()	插入新的空白文字欄位。
document.align()	對齊選取範圍。
document.allowScreens()	在使用 第 123 頁「document.screenOutline」屬性之前使用此方法。
document.arrange()	安排「舞台」上的選取範圍。
document.breakApart()	在目前的選取範圍執行打散作業。
document.canEditSymbol()	指出是否已啟用「編輯元件」選單和功能。
document.canRevert()	判斷是否可以成功使用 document.revert() 或 fl.revertDocument() 方法。
document.canTestMovie()	判斷是否可以成功使用 document.testMovie() 方法。
document.canTestScene()	判斷是否可以成功使用 document.testScene() 方法。
document.changeFilterOrder()	變更「濾鏡」清單中的濾鏡索引。
document.clipCopy()	從文件將目前的選取範圍複製到「剪貼簿」。

方法	說明
document.clipCut()	從文件將目前的選取範圍剪下並寫入「剪貼簿」。
document.clipPaste()	將「剪貼簿」內容貼至文件。
document.close()	關閉指定的文件。
document.convertLinesToFills()	將選取物件上的線條轉換為填色。
document.convertToSymbol()	將選取的「舞台」項目轉換為新元件。
document.crop()	使用上方選取的繪圖物件裁切其下方所有選取的繪圖物件。
document.debugMovie()	使用文件起始除錯工作階段。
document.deleteEnvelope()	從選取的物件刪除封套 (包含一個或多個物件的範圍框)。
document.deletePublishProfile()	如果有一個以上的描述檔，則刪除目前作用中的描述檔。
document.deleteScene()	刪除目前場景 (Timeline 物件)；如果刪除的不是最後一個場景，則設定下一個場景為目前的 Timeline 物件。
document.deleteSelection()	刪除「舞台」上目前的選取範圍。
document.disableAllFilters()	停用選取物件中的所有濾鏡。
document.disableFilter()	停用「濾鏡」清單中的指定濾鏡。
document.disableOtherFilters()	停用除「濾鏡」清單中指定位置的濾鏡之外的所有濾鏡。
document.distribute()	分散選取範圍。
document.distributeToLayers()	在目前的選取範圍執行「分散至圖層」作業；等於選取「分散至圖層」。
document.documentHasData()	檢查文件是否有指定名稱的永續性資料。
document.duplicatePublishProfile()	重製目前作用中的描述檔，並且讓重製版本成為焦點。
document.duplicateScene()	複製目前選取的場景，為新場景取唯一的名稱，讓它成為目前的場景。
document.duplicateSelection()	重製「舞台」上的選取範圍。
document.editScene()	讓指定的場景成為目前的選取場景以進行編輯。
document.enableAllFilters()	針對選取的物件啟用「濾鏡」清單上的所有濾鏡。
document.enableFilter()	啟用選取物件的指定濾鏡。
document.enterEditMode()	切換編寫工具為參數指定的編輯模式。
document.exitEditMode()	結束元件編輯模式，使編輯模式的上一個階層成為焦點。
document.exportPNG()	將文件匯出為一個或多個 PNG 檔。
document.exportPublishProfile()	將目前作用中的描述檔匯出至 XML 檔。
document.exportPublishProfileString()	傳回以 XML 格式指定特定描述檔的字串。
document.exportSWF()	將文件以 Flash SWF 格式匯出。
document.getAlignToDocument()	等於擷取「對齊」面板的「對齊舞台」按鈕值。
document.getBlendMode()	傳回指定選取物件的混合模式的字串。
document.getCustomFill()	擷取選取形狀或指定「工具」面板以及「屬性」檢測器的 Fill 物件。

方法	說明
document.getCustomStroke()	傳回選取形狀或指定「工具」面板以及「屬性」檢測器的 Stroke 物件。
document.getDataFromDocument()	擷取指定資料的值。
document.getElementProperty()	取得目前選取範圍的指定 Element 屬性。
document.getElementTextAttr()	取得選取 Text 物件的指定 TextAttrs 屬性。
document.getFilters()	傳回包含套用於目前選取物件濾鏡清單的陣列。
document.getMetadata()	傳回包含與文件相關的 XML 中繼資料的字串。
document.getMobileSettings()	傳回傳遞給 document.setMobileSettings() 的字串。
document.getPlayerVersion()	傳回字串，代表指定文件的目標播放程式版本。
document.getSelectionRect()	取得目前選取範圍的矩形邊界。
document.getTextString()	取得目前選取的文字。
document.getTimeline()	擷取文件中目前的 Timeline 物件。
document.getTransformationPoint()	取得目前選取範圍的變形點位置。
document.group()	將目前的選取範圍轉換為群組。
document.importFile()	將檔案匯入文件。
document.importPublishProfile()	從檔案匯入描述檔。
document.importPublishProfileString()	匯入代表發佈描述檔的 XML 字串，並將它設定為目前的描述檔。
document.importSWF()	將 SWF 檔匯入文件。
document.intersect()	從所有選取的繪圖物件建立交會點繪圖物件。
document.loadCuepointXML()	載入提示點 XML 檔案。
document.match()	讓選取物件的大小相同。
document.mouseClick()	使用「選取」工具執行按一下滑鼠鈕。
document.mouseDblClk()	使用「選取」工具執行按兩下滑鼠鈕。
document.moveSelectedBezierPointsBy()	如果選取範圍至少包含一個路徑，且至少選取了一個貝茲控制點，此方法會依指定量移動所有選取路徑上所有選取的貝茲控制點。
document.moveSelectionBy()	將選取物件移動指定的距離。
document.optimizeCurves()	最佳化目前選取範圍的平滑化，允許多次平滑化動作 (如果有指定) 以取得最佳平滑化；等於選取「修改 > 形狀 > 最佳化」。
document.publish()	根據作用中的發佈設定 (「檔案 > 發佈設定」) 發佈文件；等於選取「檔案 > 發佈」。
document.punch()	使用上方選取的繪圖物件穿透其下方所有選取的繪圖物件。
document.removeAllFilters()	移除選取物件的所有濾鏡。
document.removeDataFromDocument()	移除文件附加的指定名稱永續性資料。
document.removeDataFromSelection()	移除選取範圍附加的指定名稱永續性資料。
document.removeFilter()	從選取物件的「濾鏡」清單移除指定的濾鏡。
document.renamePublishProfile()	重新命名目前的描述檔。

方法	說明
document.renameScene()	重新命名「場景」面板中目前選取的場景。
document.reorderScene()	將指定場景移動到另一個指定場景之前。
document.resetOvalObject()	將「屬性」檢測器中的所有值都設定為預設的 Oval 物件設定。
document.resetRectangleObject()	將「屬性」檢測器中的所有值都設定為預設的 Rectangle 物件設定。
document.resetTransformation()	重設變形矩陣；等於選取「修改 > 變形 > 移除變形」。
document.revert()	回復指定文件至先前儲存的版本；等於選取「檔案 > 回復」。
document.rotate3DSelection()	將 3D 旋轉套用至選取範圍。
document.rotateSelection()	依指定度數旋轉選取範圍。
document.save()	將文件儲存於其預設位置；等於選取「檔案 > 儲存檔案」。
document.saveAndCompact()	儲存並壓縮檔案；等於選取「檔案 > 儲存檔案並壓縮」。
document.scaleSelection()	將選取範圍縮放指定的量；等於使用「自由變形」工具縮放物件。
document.selectAll()	選取「舞台」上所有的項目；等於按下 Control+A (Windows)、Command+A (Macintosh) 或是選取「編輯 > 全選」。
document.selectNone()	取消選取任何的選取項目。
document.setAlignToDocument()	設定偏好設定，讓 document.align()、document.distribute()、document.match() 和 document.space() 於文件中產生作用；等於啟用「對齊」面板上的「對齊舞台」按鈕。
document.setBlendMode()	設定選取物件的混合模式。
document.setCustomFill()	設定「工具」面板、「屬性」檢測器和任何選取形狀的填色設定。
document.setCustomStroke()	設定「工具」面板、「屬性」檢測器和任何選取形狀的筆畫設定。
document.setElementProperty()	設定文件中選取物件的指定 Element 屬性。
document.setElementTextAttr()	將選取文字項目的指定 TextAttrs 屬性設定為指定值。
document.setFillColor()	將選取範圍的填色顏色變更為指定顏色。
document.setFilterProperty()	設定目前選取物件的指定濾鏡屬性。
document.setFilters()	套用濾鏡至選取物件。
document.setInstanceAlpha()	設定實體的不透明度。
document.setInstanceBrightness()	設定實體的亮度。
document.setInstanceTint()	設定實體的著色。
document.setMetadata()	設定指定文件的 XML 中繼資料，覆寫任何現有的中繼資料。
document.setMobileSettings()	在行動裝置 FLA 檔中設定 XML 設定字串的值。
document.setOvalObjectProperty()	指定基本 Oval 物件的指定屬性值。
document.setPlayerVersion()	設定指定文件的目標 Flash Player 版本。
document.setRectangleObjectProperty()	指定基本 Rectangle 物件的指定屬性值。
document.setSelectionBounds()	以單一作業移動選取範圍並調整其大小。

方法	說明
document.setSelectionRect()	使用指定座標繪製相對於 「舞台」的矩形選取圈選範圍。
document.setStageVanishingPoint()	指定檢視 3D 物件的消失點。
document.setStageViewAngle()	指定檢視 3D 物件的透視角度。
document.setStroke()	設定選取筆畫的顏色、寬度和樣式。
document.setStrokeColor()	將選取範圍的筆畫顏色變更為指定顏色。
document.setStrokeSize()	將選取範圍的筆畫大小變更為指定大小。
document.setStrokeStyle()	將選取範圍的筆畫樣式變更為指定樣式。
document.setTextRectangle()	將選取文字項目的矩形邊界變更為指定大小。
document.setTextSelection()	將目前選取文字欄位的文字選取範圍設定為 *startIndex* 和 *endIndex* 指定的值。
document.setTextString()	插入文字字串。
document.setTransformationPoint()	移動目前選取範圍的變形點。
document.skewSelection()	依指定量將選取範圍斜切。
document.smoothSelection()	讓每個選取的填色外框或曲線線段的曲線平滑化。
document.space()	平均調整選取範圍中物件的間隔。
document.straightenSelection()	將目前選取的筆畫直線化；等於使用 「工具」面板中的 「直線化」按鈕。
document.swapElement()	以指定的選取範圍替換目前的選取範圍。
document.swapStrokeAndFill()	替換筆畫和填色。
document.testMovie()	在文件中執行 「測試影片」作業。
document.testScene()	在文件目前的場景中執行 「測試場景」作業。
document.traceBitmap()	在目前的選取範圍中執行轉換成向量圖；等於選取 「修改 > 點陣圖 > 轉換成向量圖」。
document.transformSelection()	套用引數中指定的矩陣，在目前的選取範圍中執行一般變形。
document.translate3DCenter()	設定 XYZ 位置，選取範圍會環繞此位置來轉譯或旋轉。
document.translate3DSelection()	將 3D 轉譯套用至選取範圍。
document.unGroup()	解散目前選取範圍的群組。
document.union()	將所有選取的形狀結合成繪圖物件。
document.unlockAllElements()	解除目前選取影格上所有鎖定元素的鎖定。
document.xmlPanel()	顯示 XMLUI 對話方塊。

屬性摘要

Document 物件可以搭配以下屬性使用。

屬性	說明
document.accName	字串；等於「輔助功能」面板中的「名稱」欄位。
document.as3AutoDeclare	Boolean 值，描述放置於「舞台」上的實體是否會自動加入至使用者定義的時間軸類別。
document.as3Dialect	字串，其中描述指定文件中使用的 ActionScript 3.0「方言」
document.as3ExportFrame	整數，指定要用來匯出 ActionScript 3.0 類別的影格。
document.as3StrictMode	Boolean 值，指定 ActionScript 3.0 編譯器是否應該在「嚴謹模式」開啟或關閉的情況下進行編譯。
document.as3WarningsMode	Boolean 值，指定 ActionScript 3.0 編譯器是否應該在「警告模式」開啟或關閉的情況下進行編譯。
document.asVersion	整數，指定所指定檔案中使用的 ActionScript 版本。
document.autoLabel	Boolean 值；等於「輔助功能」面板的「自動標籤」核取方塊。
document.backgroundColor	字串，代表背景顏色的十六進位值或整數。
document.currentPublishProfile	字串，指定所指定文件的作用中發佈描述檔名稱。
document.currentTimeline	整數，指定作用中時間軸的索引。
document.description	字串；等於「輔助功能」面板的「說明」欄位。
document.docClass	指定與文件相關聯的最上層 ActionScript 3.0 類別。
document.externalLibraryPath	字串，其中包含文件 ActionScript 3.0 外部元件庫路徑中的項目清單，指定用來做為執行階段共享元件庫之 SWC 檔的位置。
document.forceSimple	Boolean 值，指定所指定物件子系是否可存取。
document.frameRate	浮點值，指定 SWF 檔播放時的每秒顯示影格數，預設值為 12。
document.height	整數，指定文件 (「舞台」) 的高度像素。
document.id	唯一的整數 (自動指定)，可在 Flash 工作階段期間識別文件。
document.library	唯讀；文件的 library 物件。
document.libraryPath	字串，其中包含文件 ActionScript 3.0 元件庫路徑中的項目清單，指定 SWC 檔的位置或包含 SWC 檔之資料夾的位置。
document.livePreview	Boolean 值，指定是否啟用「即時預覽」。
document.name	唯讀；字串，代表文件 (FLA 檔) 名稱。
document.path	唯讀；字串，代表文件的路徑 (使用平台專用格式)。
document.pathURI	唯讀；字串，代表文件的路徑，表示為 file:/// URI。
document.publishProfiles	唯讀；文件發佈描述檔名稱的陣列。
document.screenOutline	唯讀；文件目前的 ScreenOutline 物件。從 Flash Professional CS5 開始已不再支援 ScreenOutline 物件。
document.selection	文件中的選取物件陣列。
document.silent	Boolean 值；指定物件是否可存取。
document.sourcePath	字串，其中包含文件 ActionScript 3.0 來源路徑中的項目清單，指定 ActionScript 類別檔案的位置。

屬性	說明
document.timelines	唯讀：Timeline 物件陣列 (請參閱 Timeline 物件)。
document.viewMatrix	唯讀：Matrix 物件。
document.width	整數，指定文件 (「舞台」) 的寬度像素。
document.zoomFactor	指定編寫期間的 「舞台」縮放比例。

document.accName

適用版本
Flash MX 2004。

用法
```
document.accName
```

說明
屬性；字串；等於 「輔助功能」面板中的 「名稱」欄位。螢幕朗讀程式將大聲唸出物件名稱來識別物件。

範例
下列範例將文件的輔助功能名稱設定為 "Main Movie"：

```
fl.getDocumentDOM().accName = "Main Movie";
```

下列範例會取得文件的輔助功能名稱：

```
fl.trace(fl.getDocumentDOM().accName);
```

document.addDataToDocument()

適用版本
Flash MX 2004。

用法
```
document.addDataToDocument(name, type, data)
```

參數
name 字串；指定要增加的資料名稱。

type 字串，定義要增加的資料類型。可接受的值為 "integer"、"integerArray"、"double"、"doubleArray"、"string" 和 "byteArray"。

data 要增加的值。有效類型取決於 type 參數。

傳回值
無。

說明
方法；使用文件儲存指定資料。將資料寫入 FLA 檔，並可在檔案重新開啟時供 JavaScript 使用。

範例
下列範例會在目前文件中增加整數值 12：

```
fl.getDocumentDOM().addDataToDocument("myData", "integer", 12);
```

下列範例會傳回名為 "myData" 的資料值，並在 「輸出」面板上顯示結果：

```
fl.trace(fl.getDocumentDOM().getDataFromDocument("myData"));
```

請參閱
document.getDataFromDocument()、document.removeDataFromDocument()

document.addDataToSelection()

適用版本
Flash MX 2004。

用法

```
document.addDataToSelection(name, type, data)
```

參數
name 字串；指定永續性資料名稱。

type 定義資料類型。可接受的值為 "integer"、"integerArray"、"double"、"doubleArray"、"string" 和 "byteArray"。

data 要增加的值。有效類型取決於 type 參數。

傳回值
無。

說明
方法；使用選取物件儲存指定資料。將資料寫入 FLA 檔，並可在檔案重新開啟時供 JavaScript 使用。只有元件和點陣圖支援永續性資料。

範例
下列範例在選取物件中增加整數值 12：

```
fl.getDocumentDOM().addDataToSelection("myData", "integer", 12);
```

請參閱
document.removeDataFromSelection()

document.addFilter()

適用版本
Flash 8。

用法

```
document.addFilter(filterName)
```

參數
filterName 字串，指定要加入至 「濾鏡」清單以及要為選取物件啟用的濾鏡。可接受的值為 "adjustColorFilter"、"bevelFilter"、"blurFilter"、"dropShadowFilter"、"glowFilter"、"gradientBevelFilter" 和 "gradientGlowFilter"。

傳回值
無。

說明
方法：套用濾鏡至選取的物件，並且將濾鏡放置在 「濾鏡」清單的結尾。

範例
下列範例套用光暈濾鏡至選取物件：

```
fl.getDocumentDOM().addFilter("glowFilter");
```

請參閱
document.changeFilterOrder()、document.disableFilter()、document.enableFilter()、document.getFilters()、document.removeFilter()、document.setBlendMode()、document.setFilterProperty()

document.addItem()

適用版本
Flash MX 2004。

用法
```
document.addItem(position, item)
```

參數
position 點：指定要增加項目的 x 和 y 座標位置。使用元件的中心或是點陣圖或視訊的左上角。

item Item 物件：指定要增加的項目以及增加項目的來源元件庫 (請參閱 Item 物件)。

傳回值
Boolean 值：如果成功的話，會傳回 true；否則便傳回 false。

說明
方法：從任何開啟文件或元件庫增加項目至指定 Document 物件。

範例
下列範例將元件庫的第一個項目增加至選取物件、點陣圖或視訊指定位置的第一份文件：

```
var item = fl.documents[0].library.items[0];
fl.documents[0].addItem({x:0,y:0}, item);
```

下列範例會將目前文件之元件庫中的 myMovieClip 元件增加至目前文件：

```
var itemIndex = fl.getDocumentDOM().library.findItemIndex("myMovieClip");
var theItem = fl.getDocumentDOM().library.items[itemIndex];
fl.getDocumentDOM().addItem({x:0,y:0}, theItem);
```

下列範例會將文件陣列之第二份文件中的 myMovieClip 元件增加至文件陣列的第三份文件：

```
var itemIndex = fl.documents[1].library.findItemIndex("myMovieClip");
var theItem = fl.documents[1].library.items[itemIndex];
fl.documents[2].addItem({x:0,y:0}, theItem);
```

document.addNewLine()

適用版本
Flash MX 2004。

用法
```
document.addNewLine(startPoint, endpoint)
```

參數
startpoint 一組浮點數值；指定線段開始的 x 和 y 座標。

endpoint 一組浮點數值；指定線段結束的 x 和 y 座標。

傳回值
無。

說明
方法；在兩點之間增加新的路徑。此方法使用文件的目前筆畫特質並於目前影格和圖層增加路徑。此方法的運作方式，與按一下線段工具並繪製線段相同。

範例
下列範例在指定的開始點和結束點之間增加一條線段：

```
fl.getDocumentDOM().addNewLine({x:216.7, y:122.3}, {x:366.8, y:165.8});
```

document.addNewOval()

適用版本
Flash MX 2004。

用法
```
document.addNewOval(boundingRectangle [, bSuppressFill [, bSuppressStroke ]])
```

參數
boundingRectangle 矩形；指定增加的橢圓形邊界。如需有關 boundingRectangle 格式的詳細資訊，請參閱 document.addNewRectangle()。

bSuppressFill Boolean 值；如設定為 true，方法會建立無填色形狀。預設值為 false。這個參數是選擇性參數。

bSuppressStroke Boolean 值；如設定為 true，方法會建立無筆畫形狀。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法：在指定的矩形邊界中增加新的 Oval 物件。此方法執行的作業與 「橢圓形」工具相同。此方法使用文件的目前預設筆畫和填色特質，並於目前影格和圖層增加橢圓形。如果將 bSuppressFill 和 bSuppressStroke 都設定為 true，此方法不會有任何作用。

範例
下列範例在指定座標內增加新的橢圓形；它的寬度是 164 像素，而高度是 178 像素：

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228});
```

下列範例將繪製無填色的橢圓形：

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, true);
```

下列範例將繪製無筆畫的橢圓形：

```
fl.getDocumentDOM().addNewOval({left:72,top:50,right:236,bottom:228}, false, true);
```

請參閱
document.addNewPrimitiveOval()

document.addNewPrimitiveOval()

適用版本
Flash CS4 Professional。

用法

```
document.addNewPrimitiveOval( boundingRectangle [, bSpupressFill [, bSuppressStroke ]] ))
```

參數
boundingRectangle 矩形；指定在其中增加新基本橢圓形的邊界。如需有關 boundingRectangle 格式的詳細資訊，請參閱 document.addNewRectangle()。

bSuppressFill Boolean 值；如設定為 true，方法會建立無填色橢圓形。預設值為 false。這個參數是選擇性參數。

bSuppressStroke Boolean 值；如設定為 true，方法會建立無筆畫橢圓形。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法：增加符合指定邊界之新的基本橢圓形。此方法執行的作業與 「基本橢圓形」工具相同。基本橢圓形使用文件的目前預設筆畫和填色特質，並會增加在目前影格和圖層。如果將 bSuppressFill 和 bSuppressStroke 都設定為 true，此方法不會有任何作用。

範例
下列範例在指定座標內增加新的基本橢圓形，分別是具有填色和筆畫、無填色以及無筆畫的基本橢圓形：

```
// Add an oval primitive with fill and stroke
fl.getDocumentDOM().addNewPrimitiveOval({left:0,top:0,right:100,bottom:100});
// Add an oval primitive without a fill
fl.getDocumentDOM().addNewPrimitiveOval({left:100,top:100,right:200,bottom:200}, true);
// Add an oval primitive without a stroke
fl.getDocumentDOM().addNewPrimitiveOval({left:200,top:200,right:300,bottom:300},false,true);
```

請參閱
document.addNewOval()

document.addNewPrimitiveRectangle()

適用版本
Flash CS4 Professional。

用法
```
document.addNewPrimitiveRectangle( boundingRectangle, roundness, [, bSuppressFill [, bSuppressStroke ]] ))
```

參數
rect 矩形；指定在其中增加新基本矩形的邊界。如需有關 boundingRectangle 格式的詳細資訊，請參閱 document.addNewRectangle()。

roundness 介於 0 和 999 之間的整數，代表用來指定圓角大小的點數。

bSuppressFill Boolean 值；如設定為 true，方法會建立無填色矩形。預設值為 false。這個參數是選擇性參數。

bSuppressStroke Boolean 值；如設定為 true，方法會建立無筆畫矩形。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法；增加符合指定邊界之新的基本矩形。此方法執行的作業與「基本矩形」工具相同。基本矩形使用文件的目前預設筆畫和填色特質，並會增加在目前影格和圖層。如果將 bSuppressFill 和 bSuppressStroke 都設定為 true，此方法不會有任何作用。

範例
下列範例在指定座標內增加新的基本矩形，分別是具有填色和筆畫、無填色、無筆畫，而且有不同圓角的基本矩形：

```
// Add a rectangle primitive with fill and stroke
fl.getDocumentDOM().addNewPrimitiveRectangle({left:0,top:0,right:100,bottom:100}, 0);
// Add a rectangle primitive without a fill
fl.getDocumentDOM().addNewPrimitiveRectangle({left:100,top:100,right:200,bottom:200}, 20, true);
// Add a rectangle primitive without a stroke
fl.getDocumentDOM().addNewPrimitiveRectangle({left:200,top:200,right:300,bottom:300}, 50,false,true);
```

請參閱
document.addNewRectangle()

document.addNewPublishProfile()

適用版本
Flash MX 2004。

用法
```
document.addNewPublishProfile([profileName])
```

參數
profileName　新描述檔的唯一名稱。如果不指定名稱，則會提供預設名稱。這個參數是選擇性參數。

傳回值
為描述檔清單中新描述檔索引的整數。如果無法建立描述檔，則傳回 -1。

說明
方法；增加新的發佈描述檔，並讓它成為目前的發佈描述檔。

範例
下列範例以預設名稱增加新的發佈描述檔，然後在 「輸出」面板上顯示描述檔名稱：

```
fl.getDocumentDOM().addNewPublishProfile();
fl.outputPanel.trace(fl.getDocumentDOM().currentPublishProfile);
```

下列範例會以名稱 "my profile" 增加新的發佈描述檔：

```
fl.getDocumentDOM().addNewPublishProfile("my profile");
```

請參閱
document.deletePublishProfile()

document.addNewRectangle()

適用版本
Flash MX 2004。

用法

```
document.addNewRectangle(boundingRectangle, roundness
    [, bSuppressFill [, bSuppressStroke]])
```

參數
boundingRectangle 矩形；指定在其中增加新矩形的邊界，格式為 {left:value1,top:value2,right:value3,bottom:value4}。left 和 top 值會指定左上角的位置 (例如，left:0,top:0 代表 「舞台」的左上角)，而 right 和 bottom 值則指定右下角的位置。因此，矩形的寬度就是 left 和 right 值之間的差距，而矩形的高度就是 top 和 bottom 值之間的差距。

換句話說，矩形的邊界並不完全對應於 「屬性」檢測器中顯示的值。left 和 top 值會分別對應於 「屬性」檢測器中的 X 和 Y 值。不過，right 和 bottom 值並不會對應於 「屬性」檢測器中的 W 和 H 值。例如，假設有一矩形具有以下的邊界：

```
{left:10,top:10,right:50,bottom:100}
```

這個矩形在 「屬性」檢測器中會顯示下列值：

```
X = 10, Y = 10, W = 40, H = 90
```

roundness 整數；介於 0 和 999 之間，指定轉角使用的圓形。指定值為點數。值越大，則圓形越大。

bSuppressFill Boolean 值；如設定為 true，方法會建立無填色形狀。預設值為 false。這個參數是選擇性參數。

bSuppressStroke Boolean 值；如設定為 true，方法會建立無筆畫矩形。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法：增加新的矩形或圓角矩形，讓它符合指定的邊界。此方法執行的作業與「矩形」工具相同。此方法使用文件的目前預設筆畫和填色特質，並於目前影格和圖層增加矩形。如果將 bSuppressFill 和 bSuppressStroke 都設定為 true，此方法不會有任何作用。

範例
下列範例在指定座標內增加無圓角的新矩形；它的寬度和高度都是 100 像素：

```
fl.getDocumentDOM().addNewRectangle({left:0,top:0,right:100,bottom:100},0);
```

下列範例會增加無圓角和無填色的新矩形；它的寬度是 100 像素，高度是 200 像素：

```
fl.getDocumentDOM().addNewRectangle({left:10,top:10,right:110,bottom:210},0, true);
```

下列範例會增加無圓角和無筆畫的新矩形；它的寬是 200，高度是 100 像素：

```
fl.getDocumentDOM().addNewRectangle({left:20,top:20,right:220,bottom:120},0, false, true);
```

請參閱
document.addNewPrimitiveRectangle()

document.addNewScene()

適用版本
Flash MX 2004。

用法

```
document.addNewScene([name])
```

參數
name 指定場景的名稱。如果不指定名稱，則會產生新場景名稱。

傳回值
Boolean 值：如果成功增加場景的話，便傳回 true；否則傳回 false。

說明
方法：增加新的場景 (Timeline 物件) 做為目前選取場景的下一個場景，並讓新場景成為目前選取的場景。如果已存在指定場景名稱，則不會增加場景且方法會傳回錯誤。

範例
下列範例在目前文件的目前場景之後，增加名稱為 myScene 的新場景。在建立新場景後，變數 success 為 true；否則為 false。

```
var success = fl.getDocumentDOM().addNewScene("myScene");
```

下列範例會使用預設命名慣例增加新場景。如果已存在場景，則新建立的場景名稱為 "Scene 2"。

```
fl.getDocumentDOM().addNewScene();
```

document.addNewText()

適用版本
Flash MX 2004；Flash CS3 Professional 新增的選擇性 text 參數。

用法
```
document.addNewText(boundingRectangle [, text ])
```

參數
boundingRectangle 指定文字欄位的大小和位置。如需有關 boundingRectangle 格式的詳細資訊，請參閱 document.addNewRectangle()。

text 選擇性字串，其中指定要在欄位中放置的文字。如果您省略這個參數，「工具」面板中的選取範圍就會切換成 「文字」工具。因此，如果您不想讓選取的工具變更，請針對 text 傳遞值。

傳回值
無。

說明
方法；插入新的文字欄位，並選擇性地將文字置入欄位中。如果省略 text 參數，您可以呼叫 document.setTextString() 來填入文字欄位。

範例
下列範例會在 「舞台」左上角建立新的文字欄位，然後將文字字串設定為 "Hello World"：

```
fl.getDocumentDOM().addNewText({left:0, top:0, right:100, bottom:100} , "Hello World!" );
fl.getDocumentDOM().setTextString('Hello World!');
```

請參閱
document.setTextString()

document.align()

適用版本
Flash MX 2004。

用法
```
document.align(alignmode [, bUseDocumentBounds])
```

參數
alignmode 字串；指定如何對齊選取範圍。可接受的值為 "left"、"right"、"top"、"bottom"、"vertical center" 和 "horizontal center"。

bUseDocumentBounds Boolean 值；如設定為 true，方法會對齊文件邊界；否則，方法會使用選取物件的邊界。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法：對齊選取範圍。

範例
下列範例會將物件對齊左邊和 「舞台」。等於開啟 「對齊」面板中的 「對齊舞台」設定，並按一下 「靠左對齊」按鈕：

```
fl.getDocumentDOM().align("left", true);
```

請參閱
document.distribute()、document.getAlignToDocument()、document.setAlignToDocument()

document.allowScreens()

適用版本
Flash MX 2004。

用法
```
document.allowScreens()
```

參數
無。

傳回值
Boolean 值：如果可安全地使用 document.screenOutline，則為 true；否則為 false。

說明
方法：在使用 document.screenOutline 屬性之前使用。如果這個方法傳回的值為 true，則可安全地存取 document.screenOutline；如果存取無螢幕文件的 document.screenOutline，則 Flash 會顯示錯誤。

範例
下列範例決定目前文件是否能使用 screens 方法：

```
if(fl.getDocumentDOM().allowScreens()) {
    fl.trace("screen outline is available.");
}
else {
    fl.trace("whoops, no screens.");
}
```

請參閱
document.screenOutline

document.arrange()

適用版本
Flash MX 2004。

用法
```
document.arrange(arrangeMode)
```

參數
arrangeMode 指定移動選取範圍的方向。可接受的值為 "back"、"backward"、"forward" 和 "front"。它的功能與 「修改 > 排列」選單中的選項相同。

傳回值
無。

說明
方法：排列 「舞台」上的選取範圍。此方法只適用於無形狀物件。

範例
下列範例將目前選取範圍移動到前面：
```
fl.getDocumentDOM().arrange("front");
```

document.as3AutoDeclare

適用版本
Flash CS3 Professional。

用法
```
document.as3AutoDeclare
```

說明
屬性；Boolean 值，描述放置於 「舞台」上的實體是否會自動加入至使用者定義的時間軸類別。預設值為 true。

範例
下列範例會指定在目前文件中放置於 「舞台」上的實體必須手動加入至使用者定義的時間軸類別。
```
fl.getDocumentDOM().as3AutoDeclare=false;
```

document.as3Dialect

適用版本
Flash CS3 Professional。

用法
```
document.as3Dialect
```

說明
屬性；字串，其中描述指定文件中使用的 ActionScript 3.0 「方言」。預設值為 "AS3"。如果您想要允許原型類別，如舊版 ECMAScript 規格所允許，請將這個值設定為 "ES"。

範例
下列範例會指定目前文件中使用的方言是 ECMAScript：

```
fl.getDocumentDOM().as3Dialect="ES";
```

請參閱
document.asVersion

document.as3ExportFrame

適用版本
Flash CS3 Professional。

用法

```
document.as3ExportFrame
```

說明
屬性；整數，指定要用來匯出 ActionScript 3.0 類別的影格。依預設，類別會在影格 1 中匯出。

範例
下列範例會將用來匯出類別的影格從 1 (預設值) 變更為 5。

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("'Export classes in frame:' value before modification is " +
myDocument.as3ExportFrame);
myDocument.as3ExportFrame = 5;
fl.outputPanel.trace("'Export classes in frame:' value after modification is " + myDocument.as3ExportFrame);
```

document.as3StrictMode

適用版本
Flash CS3 Professional。

用法

```
document.as3StrictMode
```

說明
屬性；Boolean 值，指定 ActionScript 3.0 編譯器應該在「嚴謹模式」開啟 (true) 或關閉 (false) 的情況下進行編譯。「嚴謹模式」會將警告回報為錯誤，這表示如果有這些錯誤存在，就無法成功完成編譯。預設值為 true。

範例
下列範例會關閉 「嚴謹模式」編譯器選項。

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("Strict Mode value before modification is " + myDocument.as3StrictMode);
myDocument.as3StrictMode = false;
fl.outputPanel.trace("Strict Mode value after modification is " + myDocument.as3StrictMode);
```

請參閱
document.as3WarningsMode

document.as3WarningsMode

適用版本
Flash CS3 Professional。

用法
```
document.as3WarningsMode
```

說明
屬性；Boolean 值，指定 ActionScript 3.0 編譯器應該在「警告模式」開啟 (true) 或關閉 (false) 的情況下進行編譯。「警告模式」會導致系統回報額外警告，而這種模式在將 ActionScript 2.0 程式碼更新為 ActionScript 3.0 時，可用來探索不相容性。預設值為 true。

範例
下列範例會關閉 「警告模式」編譯器選項。

```
var myDocument = fl.getDocumentDOM();
fl.outputPanel.trace("Warnings Mode value before modification is " + myDocument.as3WarningsMode);
myDocument.as3WarningsMode = false;
fl.outputPanel.trace("Warnings Mode value after modification is " + myDocument.as3WarningsMode);
```

請參閱
document.as3StrictMode

document.asVersion

適用版本
Flash CS3 Professional。

用法
```
document.asVersion
```

說明
屬性；整數，指定所指定文件中使用的 ActionScript 版本。可接受的值包括 1、2 和 3。

若要判斷指定文件的目標播放程式版本，請使用 document.getPlayerVersion()；這個方法會傳回字串，所以 Flash® Lite™ 播放程式可以使用此方法。

範例
下列範例會將目前文件中的 ActionScript 版本設定為 ActionScript 2.0 (如果目前設定為 ActionScript 1.0 的話)。

```
if(fl.getDocumentDOM().asVersion == 1){
    fl.getDocumentDOM().asVersion = 2;
}
```

請參閱
document.as3Dialect、document.getPlayerVersion()

document.autoLabel

適用版本
Flash MX 2004。

用法
```
document.autoLabel
```

說明
屬性；Boolean 值；等於 「輔助功能」面板的 「自動標籤」核取方塊。可使用此屬性通知 Flash 自動為 「舞台」上的物件貼上與其相關的文字標籤。

範例
下列範例取得 autoLabel 屬性值，並在 「輸出」面板上顯示結果：

```
var isAutoLabel = fl.getDocumentDOM().autoLabel;
fl.trace(isAutoLabel);
```

下列範例將 autoLabel 屬性設定為 true，以通知 Flash 自動為 「舞台」上的物件貼上標籤：

```
fl.getDocumentDOM().autoLabel = true;
```

document.backgroundColor

適用版本
Flash MX 2004。

用法
```
document.backgroundColor
```

說明
屬性；背景的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

範例
下列範例將背景顏色設定為黑色：

```
fl.getDocumentDOM().backgroundColor = '#000000';
```

document.breakApart()

適用版本
Flash MX 2004。

用法
```
document.breakApart()
```

參數
無。

傳回值
無。

說明
方法；在目前的選取範圍執行打散作業。

範例
下列範例打散目前的選取範圍：

```
fl.getDocumentDOM().breakApart();
```

document.canEditSymbol()

適用版本
Flash MX 2004。

用法
```
document.canEditSymbol()
```

參數
無。

傳回值
Boolean 值：如果可以使用 「編輯元件」選單與功能則為 true；否則為 false。

說明
方法；指示是否已啟用 「編輯元件」選單和功能。與是否可以編輯選取範圍無關。不應該使用此方法測試是否允許 fl.getDocumentDOM().enterEditMode()。

範例
下列範例在 「輸出」面板上顯示 「編輯元件」選單與功能的狀態：

```
fl.trace("fl.getDocumentDOM().canEditSymbol() returns: " + fl.getDocumentDOM().canEditSymbol());
```

document.canRevert()

適用版本
Flash MX 2004。

用法
```
document.canRevert()
```

參數
無。

傳回值
Boolean 值：如果可以成功使用 document.revert() 或 fl.revertDocument() 方法，則為 true；否則為 false。

說明
方法；決定是否可以成功使用 document.revert() 或 fl.revertDocument() 方法。

範例
下列範例檢查目前的文件是否可以回復至先前儲存的版本。如果可以，則 fl.getDocumentDOM().revert() 會還原先前儲存的版本。

```
if(fl.getDocumentDOM().canRevert()){
    fl.getDocumentDOM().revert();
}
```

document.canTestMovie()

適用版本
Flash MX 2004。

用法
```
document.canTestMovie()
```

參數
無。

傳回值
Boolean 值：如果可以成功使用 document.testMovie() 方法，則為 true；否則為 false。

說明
方法；決定是否可以成功使用 document.testMovie() 方法。

範例
下列範例測試是否可以使用 fl.getDocumentDOM().testMovie()。如果可以，便會呼叫此方法。

```
if(fl.getDocumentDOM().canTestMovie()){
    fl.getDocumentDOM().testMovie();
}
```

請參閱
document.canTestScene()、document.testScene()

document.canTestScene()

適用版本
Flash MX 2004。

用法
```
document.canTestScene()
```

參數
無。

傳回值
Boolean 值：如果可以成功使用 document.testScene() 方法，則為 true；否則為 false。

說明
方法；決定是否可以成功使用 document.testScene() 方法。

範例
下列範例會先測試是否可以成功使用 fl.getDocumentDOM().testScene()。如果可以，便會呼叫此方法。

```
if(fl.getDocumentDOM().canTestScene()){
    fl.getDocumentDOM().testScene();
}
```

請參閱
document.canTestMovie()、document.testMovie()

document.changeFilterOrder()

適用版本
Flash 8。

用法
```
document.changeFilterOrder(oldIndex, newIndex)
```

參數
oldIndex 整數；代表「濾鏡」清單中要重新定位的濾鏡索引位置，從零開始。

newIndex 整數；代表清單中濾鏡的新索引位置。

傳回值
無。

說明

方法：變更「濾鏡」清單中的濾鏡索引。任何在 newIndex 之上或之下的濾鏡都會對應地向上或向下移動。例如，使用下列濾鏡時，如果您發佈命令 fl.getDocumentDOM().changeFilterOrder(3, 0)，濾鏡便重新排列如下：

變更前	變更後
blurFilterdropShadowFilterglowFiltergradientBevelFilter	gradientBevelFilterblurFilterdropShadowFilterglowFilter

如果您接著發佈命令 fl.getDocumentDOM().changeFilterOrder(0, 2)，濾鏡便重新排列如下：

變更前	變更後
gradientBevelFilterblurFilterdropShadowFilterglowFilter	blurFilterdropShadowFiltergradientBevelFilterglowFilter

範例

下列範例會將目前「濾鏡」清單中第二個位置的濾鏡移動至第一個位置。

```
fl.getDocumentDOM().changeFilterOrder(1,0);
```

請參閱

document.addFilter()、document.disableFilter()、document.enableFilter()、document.getFilters()、document.removeFilter()、Filter 物件

document.clipCopy()

適用版本

Flash MX 2004。

用法

```
document.clipCopy()
```

參數

無。

傳回值

無。

說明

方法：從文件複製目前的選取範圍到剪貼簿。

若要將字串複製到「剪貼簿」，請使用 fl.clipCopyString()。

範例

下列範例從文件複製目前的選取範圍到剪貼簿：

```
fl.getDocumentDOM().clipCopy();
```

請參閱

document.clipCut()、document.clipPaste()

document.clipCut()

適用版本
Flash MX 2004。

用法
```
document.clipCut()
```

參數
無。

傳回值
無。

說明
方法；從文件剪下目前的選取範圍並寫入剪貼簿。

範例
下列範例從文件剪下目前的選取範圍並寫入剪貼簿。

```
fl.getDocumentDOM().clipCut();
```

請參閱
document.clipCopy()、document.clipPaste()、fl.clipCopyString()

document.clipPaste()

適用版本
Flash MX 2004。

用法
```
document.clipPaste([bInPlace])
```

參數
bInPlace Boolean 值；設定為 true 時，方法會執行在原地貼上作業。預設值為 false，方法會執行在文件中心貼上作業。這個參數是選擇性參數。

傳回值
無。

說明
方法；將剪貼簿內容貼至文件。

範例
下列範例將剪貼簿內容貼至文件中心：

```
fl.getDocumentDOM().clipPaste();
```

下列範例會將剪貼簿內容原地貼至目前文件：

```
fl.getDocumentDOM().clipPaste(true);
```

請參閱
document.clipCopy()、document.clipCut()、fl.clipCopyString()

document.close()

適用版本
Flash MX 2004。

用法
```
document.close([bPromptToSaveChanges])
```

參數
bPromptToSaveChanges Boolean 值；設定為 true 且文件中有未儲存的變更時，方法會以對話方塊提示使用者。如果 bPromptToSaveChanges 設定為 false，則不會提示使用者儲存任何變更的文件。預設值為 true。這個參數是選擇性參數。

傳回值
無。

說明
方法；關閉指定的文件。

範例
下列範例關閉目前文件，並以對話方塊提示使用者儲存變更：

```
fl.getDocumentDOM().close();
```

下列範例會關閉目前文件而不儲存變更：

```
fl.getDocumentDOM().close(false);
```

document.convertLinesToFills()

適用版本
Flash MX 2004。

用法
```
document.convertLinesToFills()
```

參數
無。

傳回值
無。

說明
方法；將選取物件的線段轉換為填色。

範例
下列範例將目前的選取線段轉換為填色：

```
fl.getDocumentDOM().convertLinesToFills();
```

document.convertToSymbol()

適用版本
Flash MX 2004。

用法
```
document.convertToSymbol(type, name, registrationPoint)
```

參數
type 字串；指定建立元件的類型。可接受的值為 "movie clip"、"button" 和 "graphic"。

name 字串；指定新元件的名稱，必須是唯一名稱。您可以送出空字串，讓此方法建立唯一元件。

registration point 點；指定代表元件的 0,0 位置。可接受的值為："top left"、"top center"、"top right"、"center left"、"center"、"center right"、"bottom left"、"bottom center" 和 "bottom right"。

傳回值
新建元件的物件；若無法建立元件，則為 null。

說明
方法；將選取的 「舞台」項目轉換為新元件。如需有關定義連結和元件共享資源屬性的詳細資訊，請參閱 Item 物件。

範例
下列範例建立含有指定名稱的影片片段元件、含有指定名稱的按鈕元件，以及含有預設名稱的影片片段：

```
newMc = fl.getDocumentDOM().convertToSymbol("movie clip", "mcSymbolName", "top left");
newButton = fl.getDocumentDOM().convertToSymbol("button", "btnSymbolName", "bottom right");
newClipWithDefaultName = fl.getDocumentDOM().convertToSymbol("movie clip", "", "top left");
```

document.crop()

適用版本
Flash 8。

用法
```
document.crop()
```

參數
無。

傳回值
Boolean 值：如果成功的話，會傳回 true；否則便傳回 false。

說明
方法；使用上方選取的繪圖物件裁切其下方所有選取的繪圖物件。如果未選取繪圖物件，或任何選取項目不是繪圖物件，則此方法傳回 false。

範例
下列範例裁切目前選取的物件：

```
fl.getDocumentDOM().crop();
```

請參閱
document.deleteEnvelope()、document.intersect()、document.punch()、document.union()、shape.isDrawingObject

document.currentPublishProfile

適用版本
Flash MX 2004。

用法
```
document.currentPublishProfile
```

說明
屬性；字串；指定所指定文件作用中發佈描述檔的名稱。

範例
下列範例以預設名稱增加新的發佈描述檔，然後在「輸出」面板上顯示描述檔名稱：

```
fl.getDocumentDOM().addNewPublishProfile();
fl.outputPanel.trace(fl.getDocumentDOM().currentPublishProfile);
```

下列範例會將選取的發佈描述檔變更為 "Default"：

```
fl.getDocumentDOM().currentPublishProfile = "Default";
```

document.currentTimeline

適用版本
Flash MX 2004。

用法
```
document.currentTimeline
```

說明
屬性；整數；指定作用中時間軸的索引。您可以變更此屬性值來設定作用中的時間軸；特效幾乎等於呼叫 document.editScene()。唯一的差異為，如果時間軸索引無效，您不會取得錯誤訊息；只是因屬性沒有設定而造成無訊息失敗。

範例
下列範例會顯示目前時間軸的索引：

```
var myCurrentTL = fl.getDocumentDOM().currentTimeline;
fl.trace("The index of the current timeline is: "+ myCurrentTL);
```

下列範例會將作用中時間軸從主要時間軸變更成名為 "myScene" 的場景：

```
var i = 0;
var curTimelines = fl.getDocumentDOM().timelines;
while(i < fl.getDocumentDOM().timelines.length){
    if(curTimelines[i].name == "myScene"){
        fl.getDocumentDOM().currentTimeline = i;
    }
    ++i;
}
```

請參閱
document.getTimeline()

document.debugMovie()

適用版本
Flash Professional CS5。

用法
```
document.DebugMovie([Boolean abortIfErrorsExist])
```

說明
方法；會叫用文件上的「影片除錯」命令。

參數
abortIfErrorsExist Boolean 值；預設值是 false。如果設定為 true，將不會啟動除錯工作階段，而如果發生編譯器錯誤，將不會開啟 .swf 視窗。編譯器警告將不會中止命令。

範例
下列範例會在除錯模式下開啟目前的文件，但如果發生編譯器錯誤，就會中止作業：

```
fl.getDocumentDOM().debugMovie(1);
```

document.deleteEnvelope()

適用版本
Flash 8。

用法
```
document.deleteEnvelope()
```

參數
無。

傳回值
Boolean 值：如果成功的話，會傳回 true；否則便傳回 false。

說明
方法；從選取的物件刪除封套 (包含一個或多個物件的範圍框)。

範例
下列範例從選取物件刪除封套：

```
fl.getDocumentDOM().deleteEnvelope();
```

請參閱
document.crop()、document.intersect()、document.punch()、document.union()、shape.isDrawingObject

document.deletePublishProfile()

適用版本
Flash MX 2004。

用法
```
document.deletePublishProfile()
```

參數
無。

傳回值
整數；為新的目前描述檔索引。如果新的描述檔無法使用，方法會保留目前的描述檔，並且傳回其索引。

說明
方法；如果有一個以上的描述檔，則刪除目前作用中的描述檔。至少必須保留一個描述檔。

範例
下列範例刪除目前作用中描述檔 (有一個以上時)，並顯示新的作用中描述檔索引：

```
alert(fl.getDocumentDOM().deletePublishProfile());
```

請參閱
document.addNewPublishProfile()

document.deleteScene()

適用版本
Flash MX 2004。

用法
```
document.deleteScene()
```

參數
無。

傳回值
Boolean 值：如果成功刪除場景，會傳回 true；否則便傳回 false。

說明
方法；刪除目前場景 (Timeline 物件)；如果刪除的不是最後一個場景，則設定下一個場景為目前的 Timeline 物件。如果刪除的是最後一個場景，便將第一個物件設定為目前的 Timeline 物件。如果只有一個 Timeline 物件 (場景)，傳回值為 false。

範例
假設目前文件有三個場景 (Scene0、Scene1 和 Scene2)，下列範例將 Scene2 設為目前場景，然後再將其刪除：

```
fl.getDocumentDOM().editScene(2);
var success = fl.getDocumentDOM().deleteScene();
```

document.deleteSelection()

適用版本
Flash MX 2004。

用法
```
document.deleteSelection()
```

參數
無。

傳回值
無。

說明
方法；刪除 「舞台」上目前的選取範圍。如果沒有選取範圍，會顯示錯誤訊息。

範例
下列範例刪除文件中目前的選取範圍：

```
fl.getDocumentDOM().deleteSelection();
```

document.description

適用版本
Flash MX 2004。

用法
```
document.description
```

說明
屬性；字串；等於 「輔助功能」面板中的 「說明」欄位。這項說明是由螢幕朗讀程式唸出。

範例
下列範例設定文件說明：

```
fl.getDocumentDOM().description= "This is the main movie";
```

下列會範例取得文件說明並將顯示於 「輸出」面板：

```
fl.trace(fl.getDocumentDOM().description);
```

document.disableAllFilters()

適用版本
Flash 8。

用法

```
document.disableAllFilters()
```

參數
無。

傳回值
無。

說明
方法；停用選取物件中的所有濾鏡。

範例
下列範例停用選取物件中的所有濾鏡：

```
fl.getDocumentDOM().disableAllFilters();
```

請參閱
document.addFilter()、document.changeFilterOrder()、document.disableFilter()、document.disableOtherFilters()、document.enableAllFilters()、document.getFilters()、document.removeAllFilters()、Filter 物件

document.disableFilter()

適用版本
Flash 8。

用法

```
document.disableFilter(filterIndex)
```

參數
filterIndex 整數，代表 「濾鏡」清單中的濾鏡索引，從零開始。

傳回值
無。

說明
方法；停用 「濾鏡」清單中的指定濾鏡。

範例
下列範例停用選取物件上，「濾鏡」清單中的第一個和第三個濾鏡 (索引值為 0 和 2)：

```
fl.getDocumentDOM().disableFilter(0);
fl.getDocumentDOM().disableFilter(2);
```

請參閱
document.addFilter()、document.changeFilterOrder()、document.disableAllFilters()、document.disableOtherFilters()、document.enableFilter()、document.getFilters()、document.removeFilter()、Filter 物件

document.disableOtherFilters()

適用版本
Flash 8。

用法
```
document.disableOtherFilters(enabledFilterIndex)
```

參數
enabledFilterIndex 整數；代表停用其它濾鏡後仍應啟用的濾鏡索引，從零開始。

傳回值
無。

說明
方法；停用除 「濾鏡」清單中指定位置的濾鏡之外的所有濾鏡。

範例
下列範例停用所有的濾鏡，清單中的第二個濾鏡除外 (索引值為 1)：

```
fl.getDocumentDom().disableOtherFilters(1);
```

請參閱
document.addFilter()、document.changeFilterOrder()、document.disableAllFilters()、document.disableFilter()、document.enableFilter()、document.getFilters()、document.removeFilter()、Filter 物件

document.distribute()

適用版本
Flash MX 2004。

用法

```
document.distribute(distributemode [, bUseDocumentBounds])
```

參數

distributemode 字串，指定分散選取物件的位置。可接受的值為 "left edge"、"horizontal center"、"right edge"、"top edge"、"vertical center" 和 "bottom edge"。

bUseDocumentBounds Boolean 值；如設定為 true，會使用文件邊界分散選取物件。否則，方法會使用選取物件的邊界。預設值為 false。

傳回值

無。

說明

方法；分散選取範圍。

範例

下列範例會依據上邊緣分散選取物件：

```
fl.getDocumentDOM().distribute("top edge");
```

下列範例會依據上邊緣分散選取物件，並且明確設定 bUseDcoumentBounds 參數：

```
fl.getDocumentDOM().distribute("top edge", false);
```

下列範例會使用文件邊界依上邊緣分散選取物件：

```
fl.getDocumentDOM().distribute("top edge", true);
```

請參閱

document.getAlignToDocument()、document.setAlignToDocument()

document.distributeToLayers()

適用版本

Flash MX 2004。

用法

```
document.distributeToLayers()
```

參數

無。

傳回值

無。

說明

方法；在目前的選取範圍執行 「分散至圖層」作業；等於選取 「分散至圖層」。如果沒有選取範圍，此方法會顯示錯誤訊息。

範例

下列範例分散目前的選取範圍至圖層：

```
fl.getDocumentDOM().distributeToLayers();
```

document.docClass

適用版本
Flash CS3 Professional。

用法
```
document.docClass
```

說明
屬性；字串，指定與文件相關的最上層 ActionScript 3.0 類別。如果文件並非設定為使用 ActionScript 3.0，就會忽略這個屬性。

範例
下列範例會指定與文件相關聯的 ActionScript 3.0 類別是 com.mycompany.ManagerClass (定義於 com/mycompany/ManagerClass.as)：

```
var myDocument = fl.getDocumentDOM();
// set the property
myDocument.docClass = "com.mycompany.ManagerClass";
// get the property
fl.outputPanel.trace("document.docClass has been set to " + myDocument.docClass);
```

請參閱
item.linkageBaseClass

document.documentHasData()

適用版本
Flash MX 2004。

用法
```
document.documentHasData(name)
```

參數
name 字串；指定要檢查的資料名稱。

傳回值
Boolean 值；如果文件有永續性資料，會傳回 true；否則便傳回 false。

說明
方法；檢查文件是否有指定名稱的永續性資料。

範例
下列範例檢查文件是否有名稱為 "myData" 的永續性資料：

```
var hasData = fl.getDocumentDOM().documentHasData("myData");
```

請參閱
document.addDataToDocument()、document.getDataFromDocument()、document.removeDataFromDocument()

document.duplicatePublishProfile()

適用版本
Flash MX 2004。

用法
```
document.duplicatePublishProfile([profileName])
```

參數
profileName 字串：指定複製描述檔的唯一名稱。如果沒有指定名稱，方法會使用預設名稱。這個參數是選擇性參數。

傳回值
整數：為描述檔清單中的新描述檔索引。如果無法複製描述檔，則傳回 -1。

說明
方法：複製目前作用中的描述檔，並且讓複製版本成為焦點。

範例
下列範例重製目前作用中的描述檔，並且在「輸出」面板上顯示新的描述檔索引：

```
fl.trace(fl.getDocumentDOM().duplicatePublishProfile("dup profile"));
```

document.duplicateScene()

適用版本
Flash MX 2004。

用法
```
document.duplicateScene()
```

參數
無。

傳回值
Boolean 值：如果成功複製場景的話，便傳回 true；否則便傳回 false。

說明
方法：複製目前選取的場景，替新場景取唯一的名稱，讓它成為目前的場景。

範例
下列範例複製目前文件中的第二個場景：

```
fl.getDocumentDOM().editScene(1); //Set the middle scene to current scene.
var success = fl.getDocumentDOM().duplicateScene();
```

document.duplicateSelection()

適用版本
Flash MX 2004。

用法
```
document.duplicateSelection()
```

參數
無。

傳回值
無。

說明
方法：複製「舞台」上的選取範圍。

範例
下列範例複製目前的選取範圍 (類似按住 Alt 並拖曳項目)：

```
fl.getDocumentDOM().duplicateSelection();
```

document.editScene()

適用版本
Flash MX 2004。

用法
```
document.editScene(index)
```

參數
index 整數；指定編輯的場景，從零開始。

傳回值
無。

說明
方法：讓指定的場景成為目前的選取場景以編輯。

範例
假設目前文件有三個場景 (Scene0、Scene1 和 Scene2)，下列範例將 Scene2 設為目前場景，然後再將其刪除：

```
fl.getDocumentDOM().editScene(2);
fl.getDocumentDOM().deleteScene();
```

document.enableAllFilters()

適用版本
Flash 8。

用法
```
document.enableAllFilters()
```

參數
無。

傳回值
無。

說明
方法；啟用 「濾鏡」清單上選取物件的所有濾鏡。

範例
下列範例啟用 「濾鏡」清單上選取物件的所有濾鏡：

```
fl.getDocumentDOM().enableAllFilters();
```

請參閱
document.addFilter()、document.changeFilterOrder()、document.disableAllFilters()、document.enableFilter()、document.getFilters()、document.removeAllFilters()、Filter 物件

document.enableFilter()

適用版本
Flash 8。

用法
```
document.enableFilter(filterIndex)
```

參數
filterIndex 整數；指定 「濾鏡」清單中啟用濾鏡的索引，從零開始。

傳回值
無。

說明
方法；啟用選取物件的指定濾鏡。

範例
下列範例啟用選取物件的第二個濾鏡：

```
fl.getDocumentDOM().enableFilter(1);
```

請參閱
document.addFilter()、document.changeFilterOrder()、document.disableFilter()、document.enableAllFilters()、document.getFilters()、document.removeFilter()、Filter 物件

document.enterEditMode()

適用版本
Flash MX 2004。

用法
```
document.enterEditMode([editMode])
```

參數
editMode 字串；指定編輯的模式。可接受的值為 "inPlace" 或 "newWindow"。如果未指定參數，預設為元件編輯模式。這個參數是選擇性參數。

傳回值
無。

說明
方法；切換編寫工具為參數指定的編輯模式。如果未指定參數，方法預設為元件編輯模式，結果與在元件上按一下滑鼠右鍵來叫用快顯選單並選取編輯相同。

範例
下列範例為目前選取元件將 Flash 設為在原地編輯模式：

```
fl.getDocumentDOM().enterEditMode('inPlace');
```

下列範例會為目前選取元件將 Flash 設為在新視窗中編輯模式：

```
fl.getDocumentDOM().enterEditMode('newWindow');
```

請參閱
document.exitEditMode()

document.exitEditMode()

適用版本
Flash MX 2004。

用法
```
document.exitEditMode()
```

參數
無。

傳回值
無。

說明
方法；結束元件編輯模式，使編輯模式的上一個階層成為焦點。例如，如果編輯的元件在另一個元件中，此方法可讓您上升一個階層，由編輯的元件到父元件。

範例
下列範例結束元件編輯模式：

```
fl.getDocumentDOM().exitEditMode();
```

請參閱
document.enterEditMode()

document.exportPNG()

適用版本
Flash 8。

用法
```
document.exportPNG([fileURI [, bCurrentPNGSettings [, bCurrentFrame]]])
```

參數
fileURI 字串；指定匯出檔案的名稱，表示為 file:/// URI。如果 fileURI 為空白字串或沒有指定，則 Flash 會顯示 「匯出影片」對話方塊。

bCurrentPNGSettings Boolean 值；指定要使用目前的 PNG 發佈設定 (true) 或顯示 「匯出 PNG」對話方塊 (false)。這個參數是選擇性參數。預設值為 false。

bCurrentFrame Boolean 值，指定只要匯出目前的影格 (true) 或要匯出所有的影格，每個影格都具有單獨的 PNG 檔 (false)。這個參數是選擇性參數。預設值為 false。

傳回值
如果成功地將檔案匯出為 PNG 檔，會傳回 Boolean 值 true；否則便傳回 false。

說明
方法；以一個或多個 PNG 檔匯出文件。如果有指定 fileURI 且檔案已經存在，就會直接覆寫檔案而不提供警告。

範例
下列範例會使用目前的 PNG 發佈設定，將目前文件中的目前影格匯出至 myFile.png：

```
fl.getDocumentDOM().exportPNG("file:///C|/myProject/myFile.png", true, true);
```

document.exportPublishProfile()

適用版本
Flash MX 2004。

用法
```
document.exportPublishProfile(fileURI)
```

參數
fileURI 字串；指定匯出描述檔的 XML 檔路徑，表示為 file:/// URI。

傳回值
無。

說明
方法；將目前作用中的描述檔匯至 XML 檔。

範例
下列範例將目前作用中描述檔匯出至 C 磁碟機的 /Documents and Settings/username/Desktop 資料夾中，名稱為 profile.xml 的檔案：

```
fl.getDocumentDOM().exportPublishProfile('file:///C|/Documents and
Settings/username/Desktop/profile.xml');
```

請參閱
document.exportPublishProfileString()、document.importPublishProfile()

document.exportPublishProfileString()

適用版本
Flash CS4 Professional。

用法
```
document.exportPublishProfileString( [profileName] )
```

參數
profileName 字串，指定要匯出為 XML 字串之描述檔的名稱。這個參數是選擇性參數。

傳回值
XML 字串。

說明
方法；傳回以 XML 格式指定特定描述檔的字串。若未傳遞值給 profileName，則會匯出目前描述檔。

範例
下列範例會將表示目前描述檔的 XML 字串儲存在名為 profileXML 的變數中，然後在「輸出」面板中顯示該字串：

```
var profileXML=fl.getDocumentDOM().exportPublishProfileString();
fl.trace(profileXML);
```

請參閱
document.exportPublishProfile()、document.importPublishProfileString()

document.exportSWF()

適用版本
Flash MX 2004。

用法
```
document.exportSWF([fileURI [, bCurrentSettings]])
```

參數
fileURI 字串；指定匯出檔案的名稱，表示為 file:/// URI。如果 fileURI 為空白或沒有指定，則 Flash 會顯示 「匯出影片」對話方塊。這個參數是選擇性參數。

bCurrentSettings Boolean 值；設定為 true 時，Flash 會使用目前的 SWF 發佈設定。否則，Flash 會顯示 「匯出 Flash Player」對話方塊。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法；將文件以 Flash SWF 格式匯出。

範例
下列範例根據目前的發佈設定將文件匯出至指定的檔案位置：

```
fl.getDocumentDOM().exportSWF("file:///C|/Documents and Settings/joe_user/Desktop/qwerty.swf");
```

下列範例會顯示 「匯出影片」對話方塊與 「匯出 Flash Player」對話方塊，然後根據指定設定匯出文件：

```
fl.getDocumentDOM().exportSWF("", true);
```

下列範例會顯示 「匯出影片」對話方塊，然後根據指定設定匯出文件：

```
fl.getDocumentDOM().exportSWF();
```

document.externalLibraryPath

適用版本
Flash CS4 Professional。

用法
```
document.externalLibraryPath
```

說明
屬性；字串，其中包含文件 ActionScript 3.0 外部元件庫路徑中的項目清單，指定用來做為執行階段共享元件庫之 SWC 檔的位置。此字串中的項目會以分號隔開。在編寫工具中，藉由選擇 「檔案 > 發佈設定」，然後選擇 「Flash」索引標籤上的 「ActionScript 3.0 指令碼設定」來指定項目。

範例
下列範例會將文件的外部元件庫路徑設定為 "." 和 "../mySWCLibrary"：

```
var myDocument = fl.getDocumentDOM();
myDocument.externalLibraryPath = ".;../mySWCLibrary";
fl.trace(myDocument.externalLibraryPath);
```

請參閱

document.libraryPath、document.sourcePath、fl.externalLibraryPath

document.forceSimple

適用版本

Flash MX 2004。

用法

```
document.forceSimple
```

說明

屬性；Boolean 值；指定所指定物件子系是否可存取。等於 「輔助功能」面板 「讓子物件支援輔助功能」設定的反向邏輯。也就是說，如果 forceSimple 為 true，則等於未選取 「讓子物件支援輔助功能」選項。如果 forceSimple 為 false，則等於選取「讓子物件支援輔助功能」選項。

範例

下列範例會將 areChildrenAccessible 變數設定為 forceSimple 屬性的值。false 值表示子系是可存取的。

```
var areChildrenAccessible = fl.getDocumentDOM().forceSimple;
```

下列範例會設定 forceSimple 屬性，以便允許存取文件的子系：

```
fl.getDocumentDOM().forceSimple = false;
```

document.frameRate

適用版本

Flash MX 2004。

用法

```
document.frameRate
```

說明

屬性；浮點值；指示 SWF 檔播放時的每秒顯示影格數；預設值為 12。設定這個屬性與在 「文件屬性」對話方塊 (「修改 > 文件」) 設定影格速率具有相同的功能。

範例

下列範例設定影格速率為每秒 25.5 影格：

```
fl.getDocumentDOM().frameRate = 25.5;
```

document.getAlignToDocument()

適用版本
Flash MX 2004。

用法
```
document.getAlignToDocument()
```

參數
無。

傳回值
Boolean 值：如果將偏好設定設定為物件對齊 「舞台」，則為 true；否則為 false。

說明
方法；等於擷取 「對齊」面板的 「對齊舞台」按鈕值。取得偏好設定，可用於文件上的 document.align()、document.distribute()、document.match() 和 document.space() 方法。

範例
下列範例擷取 「對齊」面板上的 「對齊舞台」值。如果傳回值為 true，則 「對齊舞台」按鈕為作用中；否則為停用。

```
var isAlignToDoc = fl.getDocumentDOM().getAlignToDocument();
fl.getDocumentDOM().align("left", isAlignToDoc);
```

請參閱
document.setAlignToDocument()

document.getBlendMode()

適用版本
Flash 8。

用法
```
document.getBlendMode()
```

參數
無。

傳回值
指定選取物件的混合模式的字串。若選取一個以上的物件，而且這些物件具有不同的混合模式，字串會反映出具最長深度物件的混合模式。

備註：如果選取範圍包含不支援混合模式的物件，或者包含 "normal" 混合模式值，傳回的值就無法預測。

說明
方法；傳回指定選取物件的混合模式的字串。

範例
下列範例將顯示 「輸出」面板中的混合模式名稱：

```
fl.trace(fl.getDocumentDom().getBlendMode());
```

document.getCustomFill()

適用版本
Flash MX 2004。

用法
```
document.getCustomFill([objectToFill])
```

參數
objectToFill 字串；指定 Fill 物件的位置。以下為有效值：

- "toolbar" 傳回 「工具」面板與 「屬性」檢測器的 Fill 物件。
- "selection" 傳回選取範圍的 Fill 物件。

  如忽略參數，則預設值為 "selection"。如果沒有任何的選取範圍，方法會傳回 undefined。這個參數是選擇性參數。

傳回值
如果成功，會傳回 objectToFill 參數指定的 Fill 物件；否則會傳回 undefined。

說明
方法；擷取選取形狀或指定 「工具」面板以及 「屬性」檢測器的 Fill 物件。

範例
下列範例取得選取範圍的 Fill 物件，然後將選取範圍的顏色變更為白色：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.color = '#FFFFFF';
fill.style = "solid";
fl.getDocumentDOM().setCustomFill(fill);
```

下列範例會傳回 「工具」面板和 「屬性」檢測器的 Fill 物件，然後將顏色色票變更為線性漸層：

```
var fill = fl.getDocumentDOM().getCustomFill("toolbar");
fill.style = "linearGradient";
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];
fill.posArray = [0, 100, 200];
fl.getDocumentDOM().setCustomFill( fill );
```

請參閱
document.setCustomFill()

document.getCustomStroke()

適用版本
Flash MX 2004。

用法

```
document.getCustomStroke([locationOfStroke])
```

參數

locationOfStroke 字串；指定 Stroke 物件的位置。以下為有效值：

- 如設定 "toolbar"，會傳回 「工具」面板和 「屬性」檢測器的 Stroke 物件。
- 如設定 "selection"，會傳回選取範圍的 Stroke 物件。

  如忽略參數，則預設值為 "selection"。如果沒有任何的選取範圍，則傳回 undefined。這個參數是選擇性參數。

傳回值

如果成功，會傳回 locationOfStroke 參數指定的 Stroke 物件；否則會傳回 undefined。

說明

傳回選取形狀或指定 「工具」面板以及 「屬性」檢測器的 Stroke 物件。

範例

下列範例傳回選取範圍目前的筆畫設定，並變更筆畫粗細為 2：

```
var stroke = fl.getDocumentDOM().getCustomStroke("selection");
stroke.thickness = 2;
fl.getDocumentDOM().setCustomStroke(stroke);
```

下列範例會傳回 「工具」面板和 「屬性」檢測器目前的筆畫設定，並將筆畫顏色設定為紅色：

```
var stroke = fl.getDocumentDOM().getCustomStroke("toolbar");
stroke.color = "#FF0000";
fl.getDocumentDOM().setCustomStroke(stroke);
```

請參閱

document.setCustomStroke()

document.getDataFromDocument()

適用版本

Flash MX 2004。

用法

```
document.getDataFromDocument(name)
```

參數

name 字串；指定要傳回的資料名稱。

傳回值

指定資料。

說明

方法；擷取指定資料的值。傳回的類型取決於儲存的資料類型。

範例
下列範例於目前文件中增加整數值 12，並使用此方法在「輸出」面板上顯示值：

```
fl.getDocumentDOM().addDataToDocument("myData", "integer", 12);
fl.trace(fl.getDocumentDOM().getDataFromDocument("myData"));
```

請參閱
document.addDataToDocument()、document.documentHasData()、document.removeDataFromDocument()

document.getElementProperty()

適用版本
Flash MX 2004。

用法
```
document.getElementProperty(propertyName)
```

參數
propertyName 字串，指定要擷取其值的 Element 屬性名稱。

傳回值
指定屬性的值。如果屬性為不確定狀態 (同時有多個元素選取不同的屬性值)，則傳回 null。如果屬性不是選取元素的有效屬性，則傳回 undefined。

說明
方法：取得目前選取範圍的指定 Element 屬性。如需可接受值的清單，請參閱 Element 物件的屬性摘要表。

範例
下列範例取得目前選取範圍 Element 屬性的 name：

```
// elementName = the instance name of the selected object.
var elementName = fl.getDocumentDOM().getElementProperty("name");
```

請參閱
document.setElementProperty()

document.getElementTextAttr()

適用版本
Flash MX 2004。

用法
```
document.getElementTextAttr(attrName [, startIndex [, endIndex]])
```

參數
attrName 字串：指定傳回的 TextAttrs 屬性名稱。如需屬性名稱和期望值的清單，請參閱 TextAttrs 物件的屬性摘要表。

startIndex 整數：指定第一個字元的索引，0 (零) 指定第一個位置。這個參數是選擇性參數。

endIndex 整數；指定最後一個字元的索引。這個參數是選擇性參數。

傳回值
如果選取一個文字欄位，而且文字內只使用一個值，則傳回此屬性。如果文字欄位內使用數個值，則傳回 undefined。如果選取數個文字欄位，而且所有的文字對齊值相同，則方法傳回此值。如果選取數個文字欄位，但是所有的文字對齊值都不相同，則方法傳回 undefined。如果沒有傳遞選擇性引數，則這些規則會套用到目前選取的文字範圍；如果目前未編輯文字，則套用到整個文字欄位。如果僅傳遞 startIndex，則在所有選取 Text 物件符合值時，才會傳回索引右邊的字元屬性。如果傳遞 startIndex 和 endIndex，則傳回值反映 startIndex 到 endIndex 的所有字元範圍 (但不包括 endIndex)。

說明
方法；取得選取 Text 物件的指定 TextAttrs 屬性。會忽略不為文字欄位的選取物件。如需屬性名稱和期望值的清單，請參閱 TextAttrs 物件的屬性摘要表。請參閱 document.setElementTextAttr()。

範例
下列範例取得選取文字欄位的大小：

```
fl.getDocumentDOM().getElementTextAttr("size");
```

下列範例會取得選取文字欄位中，索引 3 的字元顏色：

```
fl.getDocumentDOM().getElementTextAttr("fillColor", 3);
```

下列範例會取得選取文字欄位中，索引 2 到索引 10 的文字字體名稱 (但不包括索引 10)：

```
fl.getDocumentDOM().getElementTextAttr("face", 2, 10);
```

document.getFilters()

適用版本
Flash 8。

用法
```
document.getFilters()
```

參數
無。

傳回值
陣列，包含套用於目前選取物件的濾鏡清單。

說明
方法；傳回包含套用於目前選取物件濾鏡清單的陣列。如果選取多個物件，而這些物件沒有相似的濾鏡，此方法會傳回套用到第一個選取物件的濾鏡清單。

範例
請參閱 document.setFilters()。

請參閱
document.addFilter()、document.changeFilterOrder()、document.setFilters()、Filter 物件

document.getMetadata()

適用版本
Flash 8。

用法
```
document.getMetadata()
```

參數
無。

傳回值
包含文件相關 XML 中繼資料的字串；如果沒有中繼資料，則為空字串。

說明
方法；傳回包含文件相關 XML 中繼資料的字串；若無中繼資料，則傳回空字串。

範例
下列範例在「輸出」面板上顯示目前文件的 XML 中繼資料：

```
fl.trace("XML Metadata is :" + fl.getDocumentDOM().getMetadata());
```

請參閱
document.setMetadata()

document.getMobileSettings()

適用版本
Flash CS3 Professional。

用法
```
document.getMobileSettings()
```

參數
無。

傳回值
字串，代表文件的 XML 設定。如果尚未設定任何值，就會傳回空字串。

說明
方法；傳回文件的行動裝置 XML 設定。

範例
下列範例會顯示目前文件的 XML 設定字串：

```
fl.trace(fl.getDocumentDOM().getMobileSettings());
//traces a string like the following"<? xml version="1.0" encoding="UTF-16" standalone="no"
?><mobileSettings> <contentType id="standalonePlayer" name="Standalone Player"/> <testDevices> <testDevice
id="1170" name="Generic Phone" selected="yes"/> </testDevices> <outputMsgFiltering info="no" trace="yes"
warning="yes"/> <testWindowState height="496" splitterClosed="No" splitterXPos="400" width="907"/>
</mobileSettings>"
```

請參閱
document.setMobileSettings()

document.getPlayerVersion()

適用版本
Flash CS3 Professional。

用法
```
document.getPlayerVersion()
```

參數
無。

傳回值
字串，代表透過使用 document.setPlayerVersion() 指定的 Flash Player 版本。如果尚未設定任何值，就會傳回在 「發佈設定」對話方塊中指定的值。

說明
方法；傳回字串，代表指定文件的目標播放程式版本。如需此方法可傳回值的清單，請參閱 document.setPlayerVersion()。

若要決定指定檔案中的目標 ActionScript 版本，請使用 document.asVersion。

範例
下列範例會說明如何指定目前文件的目標播放程式版本，然後擷取這些值：

```
fl.getDocumentDOM().setPlayerVersion("6");
var version = fl.getDocumentDOM().getPlayerVersion();
fl.trace(version) // displays "6"
fl.getDocumentDOM().setPlayerVersion("FlashPlayer10");
var version = fl.getDocumentDOM().getPlayerVersion();
fl.trace(version) // displays ""FlashPlayer10""
```

請參閱
document.setPlayerVersion()

document.getSelectionRect()

適用版本
Flash MX 2004。

用法
```
document.getSelectionRect()
```

參數
無。

傳回值
目前選取範圍的矩形邊界；若無選取，則為 0。如需有關傳回值格式的詳細資訊，請參閱 document.addNewRectangle()。

說明
方法：取得目前選取範圍的矩形邊界。如果選取範圍非矩形，則傳回包含整個選取範圍的最小矩形。此矩形依據文件空間或是編輯模式中的編輯元件註冊點 (亦稱為「原點」或「零點」)。

範例
下列範例取得目前選取範圍的矩形邊界，然後顯示其屬性：

```
var newRect = fl.getDocumentDOM().getSelectionRect();
var outputStr = "left: " + newRect.left + " top: " + newRect.top + " right: " + newRect.right + " bottom:
" + newRect.bottom;
alert(outputStr);
```

請參閱
document.selection、document.setSelectionRect()

document.getTextString()

適用版本
Flash MX 2004。

用法
```
document.getTextString([startIndex [, endIndex]])
```

參數
startIndex 整數；第一個取得字元的索引。這個參數是選擇性參數。

endIndex 整數；最後一個取得字元的索引。這個參數是選擇性參數。

傳回值
包含選取文字的字串。

說明
方法：取得目前選取的文字。如未傳遞選擇性參數，則會使用目前文字選取範圍。如果目前未開啟文字編輯，則會傳回整個文字字串。如果僅傳遞 startIndex，則傳回開始於該索引且結束於欄位結尾的字串。如果傳遞 startIndex 和 endIndex，則傳回自 startIndex 開始到 endIndex 的字串 (但不包括 endIndex)。

如果選取數個文字欄位，則傳回所有的字串連接。

範例
下列範例取得選取文字欄位的字串：

```
fl.getDocumentDOM().getTextString();
```

下列範例會取得選取文字欄位中字元索引 5 的字串：

```
fl.getDocumentDOM().getTextString(5);
```

下列範例會取得從字元索引 2 到字元索引 10 的字串 (但不包括字元索引 10)：

```
fl.getDocumentDOM().getTextString(2, 10);
```

請參閱
document.setTextString()

document.getTimeline()

適用版本
Flash MX 2004。

用法

```
document.getTimeline()
```

參數
無。

傳回值
目前 Timeline 物件。

說明
方法；擷取文件中目前的 Timeline 物件。目前的時間軸可以是目前場景、目前編輯字元或目前螢幕。

範例
下列範例取得 Timeline 物件，並傳回最長圖層中的影格數目：

```
var longestLayer = fl.getDocumentDOM().getTimeline().frameCount;
fl.trace("The longest layer has" + longestLayer + "frames");
```

下列範例會針對「舞台」上的選取元件進入原地編輯模式，並在元件的時間軸上插入一個影格。

```
fl.getDocumentDOM().enterEditMode("inPlace");
fl.getDocumentDOM().getTimeline().insertFrames();
```

下列範例會取得 Timeline 物件並顯示其名稱：

```
var timeline = fl.getDocumentDOM().getTimeline();
alert(timeline.name);
```

請參閱
document.currentTimeline、document.timelines、symbolItem.timeline

document.getTransformationPoint()

適用版本
Flash MX 2004。

用法
```
document.getTransformationPoint()
```

參數
無。

傳回值
指定在所選元素座標系統中變形點 (亦稱為原點或零點) 位置的點 (例如 {x:10, y:20}，其中 x 和 y 是浮點數字)。

說明
方法：取得目前選取範圍的變形點位置。可以使用變形點進行換向 (Commutation)，如旋轉和斜切。

備註：根據選取項目的類型而定，變形點會相對於不同的位置。如需詳細資訊，請參閱 document.setTransformationPoint()。

範例
下列範例取得目前選取範圍的變形點。transPoint.x 屬性會提供變形點的 x 座標。transPoint.y 屬性會提供變形點的 y 座標。

```
var transPoint = fl.getDocumentDOM().getTransformationPoint();
```

請參閱
document.setTransformationPoint()、element.getTransformationPoint()

document.group()

適用版本
Flash MX 2004。

用法
```
document.group()
```

參數
無。

傳回值
無。

說明
方法：將目前的選取範圍轉換成群組。

範例
下列範例將目前選取範圍的物件轉換為群組。

```
fl.getDocumentDOM().group();
```

請參閱
document.unGroup()

document.height

適用版本
Flash MX 2004。

用法
```
document.height
```

說明
屬性；整數；指定文件 (「舞台」) 的高度像素。

範例
下列範例設定 「舞台」高度為 400 像素：

```
fl.getDocumentDOM().height = 400;
```

請參閱
document.width

document.id

適用版本
Flash CS3 Professional。

用法
```
document.id
```

說明
唯讀屬性；唯一的整數 (自動指定)，可在 Flash 工作階段期間識別文件。與 fl.findDocumentDOM() 搭配使用此屬性，以指定動作的特定文件。

範例
下列範例會顯示目前文件的文件 ID：

```
fl.trace("Current doc's internal ID is: " + fl.getDocumentDOM().id);
```

請參閱
fl.findDocumentDOM()

document.importFile()

適用版本
Flash 8。

用法
```
document.importFile(fileURI [, importToLibrary])
```

參數
fileURI 字串；指定要匯入之檔案的路徑，表示為 file:/// URI。

importToLibrary Boolean 值；指定只要將檔案匯入文件的元件庫 (true) 或是還要在 「舞台」上放置檔案副本 (false)。預設值為 false。

傳回值
無。

說明
方法；將檔案匯入文件。這個方法執行的作業與 「匯入至元件庫」或 「匯入至舞台」選單命令相同。若要匯入發佈描述檔，請使用 document.importPublishProfile()。

範例
下列範例會讓使用者瀏覽要匯入 「舞台」的檔案：

```
var dom = fl.getDocumentDOM();
var URI = fl.browseForFileURL("select", "Import File");
dom.importFile(URI);
```

請參閱
document.importSWF()、fl.browseForFileURL()

document.importPublishProfile()

適用版本
Flash MX 2004。

用法
```
document.importPublishProfile( fileURI )
```

參數
fileURI 字串；指定定義匯入描述檔的 XML 檔路徑，表示為 file:/// URI。

傳回值
為描述檔清單中匯入描述檔索引的整數。如果無法匯入描述檔，則傳回 -1。

說明
方法；從檔案匯入描述檔。

範例
下列範例匯入 profile.xml 檔中包含的描述檔，並在描述檔清單中顯示其索引：

```
alert(fl.getDocumentDOM().importPublishProfile('file:///C|/Documents and
Settings/janeUser/Desktop/profile.xml'));
```

document.importPublishProfileString()

適用版本
Flash CS4 Professional。

用法
```
document.importPublishProfileString(xmlString)
```

參數
xmlString 字串，包含要匯入做為目前描述檔的 XML 資料。

傳回值
如果成功匯入字串，就會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；匯入代表發佈描述檔的 XML 字串，並將它設定為目前的描述檔。若要產生要匯入的 XML 字串，請先使用 document.exportPublishProfileString()，然後再使用此方法。

範例
在下列範例中，預設描述檔會匯出為 XML 字串。標準 JavaScript replace 命令會用來修改此 XML 字串。接著會匯入字串，並將預設 ActionScript 3 輸出設定設為 ActionScript 1。

```
var profileXML=fl.getDocumentDOM().exportPublishProfileString('Default');
fl.trace(profileXML);
var newProfileXML = profileXML.replace("<ActionScriptVersion>3</ActionScriptVersion>",
"<ActionScriptVersion>1</ActionScriptVersion>");
fl.getDocumentDOM().importPublishProfileString(newProfileXML);
```

document.importSWF()

適用版本
Flash MX 2004。

用法
```
document.importSWF(fileURI)
```

參數
fileURI 字串；指定 SWF 檔匯入的檔案，表示為 file:/// URI。

傳回值
無。

說明
方法：將 SWF 檔匯入文件。這個方法所執行的作業與使用「匯入」選單命令指定 SWF 檔相同。在 Flash 8 和更新版本中，還可以使用 document.importFile() 匯入 SWF 檔 (以及其它的檔案類型)。

範例
下列範例從 Flash Configuration 資料夾匯入 "mySwf.swf" 檔：

```
fl.getDocumentDOM().importSWF(fl.configURI+"mySwf.swf");
```

請參閱
document.importFile()

document.intersect()

適用版本
Flash 8。

用法
```
document.intersect()
```

參數
無。

傳回值
Boolean 值：如果成功的話，會傳回 true；否則便傳回 false。

說明
方法：從所有選取的繪圖物件建立交會點繪圖物件。如果未選取繪圖物件，或任何選取項目不是繪圖物件，則此方法傳回 false。

範例
下列範例從所有選取的繪圖物件建立交會點繪圖物件：

```
fl.getDocumentDOM().intersect();
```

請參閱
document.crop()、document.deleteEnvelope()、document.punch()、document.union()、shape.isDrawingObject

document.library

適用版本
Flash MX 2004。

用法
```
document.library
```

說明
唯讀屬性：文件的 library 物件。

範例
下列範例取得目前焦點文件的元件庫：

```
var myCurrentLib = fl.getDocumentDOM().library;
```

假設目前焦點文件不是 fl.documents[1]，則下列範例會取得非焦點元件庫的元件庫，或使用 「檔案 > 開啟」開啟為外部元件庫的元件庫：

```
var externalLib = fl.documents[1].library;
```

document.libraryPath

適用版本
Flash CS4 Professional。

用法
```
document.libraryPath
```

說明
屬性：字串，其中包含文件 ActionScript 3.0 元件庫路徑中的項目清單，指定 SWC 檔的位置或包含 SWC 檔之資料夾的位置。此字串中的項目會以分號隔開。在編寫工具中，藉由選擇 「檔案 > 發佈設定」，然後選擇 「Flash」索引標籤上的 「ActionScript 3.0 指令碼設定」來指定項目。

範例
下列範例會將 ../Files 資料夾加入至文件的元件庫路徑，然後在 「輸出」面板中顯示該元件庫路徑：

```
var myDoc = fl.getDocumentDOM()
fl.trace(myDoc.libraryPath);
myDoc.libraryPath = "../Files;" + myDoc.libraryPath;
fl.trace(myDoc.libraryPath);
```

請參閱
document.externalLibraryPath、document.sourcePath、fl.libraryPath

document.livePreview

適用版本
Flash MX 2004。

用法
```
document.livePreview
```

說明
屬性：Boolean 值，指定是否啟用 「即時預覽」。如果設定為 true，組件會依發佈的 Flash 內容中的顯示方式顯示在 「舞台」上 (包括組件的概略大小)。如果設定為 false，組件就只顯示外框。預設值為 true。

範例
下列範例將 「即時預覽」設定為 false：

```
fl.getDocumentDOM().livePreview = false;
```

document.loadCuepointXML()

適用版本
Flash Professional CS5

用法

```
document.loadCuepointXML(String URI)
```

參數
URI 字串；提示點 XML 檔案的絕對路徑。

說明
方法；會載入提示點 XML 檔案。XML 檔案的格式和 DTD 與 「提示點屬性」檢測器匯入和匯出的檔案相同。傳回值與包含 FLVPlayback 元件實體之物件的 「提示點」屬性中所序列化的字串相同。

範例
下列提示點 XML 範例檔案位於 C:\\testCuePoints.xml：

```
var cuePoints = fl.getDocumentDOM().LoadCuepointXML("c:\\testCuePoints.xml");
```

document.match()

適用版本
Flash MX 2004。

用法

```
document.match(bWidth, bHeight [, bUseDocumentBounds])
```

參數
bWidth Boolean 值；如果設定為 true，方法會讓選取項目的寬度相同。

bHeight Boolean 值；如果設定為 true，方法會讓選取項目的高度相同。

bUseDocumentBounds Boolean 值；如設定為 true，方法會讓物件大小符合文件邊界。否則，方法會使用最大物件的邊界。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法；讓選取物件的大小相同。

範例
下列範例只符合選取物件的寬度：

```
fl.getDocumentDOM().match(true,false);
```

下列範例只符合高度：

```
fl.getDocumentDOM().match(false,true);
```

下列範例只符合文件邊界的寬度：

```
fl.getDocumentDOM().match(true,false,true);
```

請參閱
document.getAlignToDocument()、document.setAlignToDocument()

document.mouseClick()

適用版本
Flash MX 2004。

用法

```
document.mouseClick(position, bToggleSel, bShiftSel)
```

參數
position 一組浮點值，指定按一下動作的 x 和 y 座標，以像素為單位。

bToggleSel Boolean 值：指定 Shift 鍵的狀態，若為按下則傳回 true，否則傳回 false。

bShiftSel Boolean 值：指定應用程式偏好設定的 Shift 選擇狀態，若為開啟則為 true，否則為 false。

傳回值
無。

說明
方法：使用 「選取」工具執行按一下滑鼠鈕。

範例
下列範例在指定位置執行按一下滑鼠鈕：

```
fl.getDocumentDOM().mouseClick({x:300, y:200}, false, false);
```

請參閱
document.mouseDblClk()

document.mouseDblClk()

適用版本
Flash MX 2004。

用法
```
document.mouseDblClk(position, bAltDown, bShiftDown, bShiftSelect)
```

參數
position 一組浮點值，指定按一下動作的 x 和 y 座標，以像素為單位。

bAltdown Boolean 值：記錄事件發生時是否按下 Alt 鍵，若為按下則傳回 true，否則為 false。

bShiftDown Boolean 值：記錄在事件發生時是否按下 Shift 鍵，若為按下則傳回 true，否則為 false。

bShiftSelect Boolean 值：指出應用程式偏好設定的 Shift 選擇狀態，若為開啟則為 true，否則為 false。

傳回值
無。

說明
方法：使用 「選取」工具執行按兩下滑鼠鈕。

範例
下列範例在指定位置執行按兩下滑鼠鈕：
```
fl.getDocumentDOM().mouseDblClk({x:392.9, y:73}, false, false, true);
```

請參閱
document.mouseClick()

document.moveSelectedBezierPointsBy()

適用版本
Flash MX 2004。

用法
```
document.moveSelectedBezierPointsBy(delta)
```

參數
delta 一組浮點值；指定選取之貝茲控制點移動時所依據的 x 和 y 座標，以像素為單位。例如，傳遞 ({x:1,y:2}) 指定的位置為：目前位置向右一個像素，向下兩個像素。

傳回值
無。

說明
方法：如果選取範圍至少包含一個路徑，且至少選取了一個貝茲控制點，則依指定量移動所有選取路徑上所有的選取貝茲控制點。

範例
下列範例將選取的貝茲控制點向右移動 10 像素，向下移動 5 像素：
```
fl.getDocumentDOM().moveSelectedBezierPointsBy({x:10, y:5});
```

document.moveSelectionBy()

適用版本
Flash MX 2004。

用法
```
document.moveSelectionBy(distanceToMove)
```

參數
distanceToMove 一組浮點值；指定方法移動選取範圍時所依據的 x 和 y 座標值。例如，傳遞 ({x:1,y:2}) 指定的位置為：目前位置向右一個像素，向下兩個像素。

傳回值
無。

說明
方法；將選取物件移動指定的距離。

備註：當使用者使用方向鍵移動項目時，「操作記錄」面板會將所有按下的方向鍵結合為一個移動步驟。使用者重複按下方向鍵，而不是在「操作記錄」面板中採取多個步驟時，方法會執行一個步驟，且引數會更新反映重複的方向鍵。

如需有關選擇的詳細資訊，請參閱 document.setSelectionRect()、document.mouseClick()、document.mouseDblClk() 和 Element 物件。

範例
下列範例將選取項目向右移動 62 像素，向下移動 84 像素：

```
fl.getDocumentDOM().moveSelectionBy({x:62, y:84});
```

document.name

適用版本
Flash MX 2004。

用法
```
document.name
```

說明
唯讀屬性；字串；代表文件 (FLA 檔) 名稱。

範例
下列範例將變數 fileName 設定為文件陣列中第一份文件的檔案名稱：

```
var fileName = flash.documents[0].name;
```

下列範例會在「輸出」面板中顯示所有開啟文件的名稱：

```
var openDocs = fl.documents;
for(var i=0;i < openDocs.length; i++){
    fl.trace(i + " " + openDocs[i].name +"\n");
}
```

document.optimizeCurves()

適用版本
Flash MX 2004。

用法
```
document.optimizeCurves(smoothing, bUseMultiplePasses)
```

參數
smoothing 介於 0 到 100 之間的整數，0 指定無平滑化，100 指定最大平滑化。

bUseMultiplePasses Boolean 值；設定為 true 時，指示方法應使用多次平滑化動作，速度雖然比較慢，但產生的結果較佳。此參數的特效與按一下「最佳化曲線」對話方塊中的「使用多次平滑化動作」按鈕相同。

傳回值
無。

說明
方法；最佳化目前選取範圍的平滑化，允許多次平滑化動作 (如果有指定) 以取得最佳平滑化；此方法等於選取「修改 > 形狀 > 最佳化」。

範例
下列範例使用多次平滑化動作，將目前選取範圍的曲線平滑化程度最佳化為 50 度：

```
fl.getDocumentDOM().optimizeCurves(50, true);
```

document.path

適用版本
Flash MX 2004。

用法
```
document.path
```

說明
唯讀屬性；字串；代表文件的路徑。如果文件未儲存，則屬性為 undefined。

範例
下列範例在「輸出」面板中顯示文件陣列中第一份文件的路徑。您必須先儲存文件，再執行此指令碼。在範例中，檔案名稱為 test.fla，儲存在 Windows 電腦上的 My Documents 資料夾。

```
var filePath = flash.documents[0].path;
fl.trace(filePath);
// displays C:\Documents and Settings\<user name>\My Documents\test.fla
```

請參閱
document.pathURI

document.pathURI

適用版本
Flash CS4 Professional。

用法
```
document.pathURI
```

說明
唯讀屬性：字串，代表文件的路徑，表示為 file:/// URI。如果文件未儲存，則屬性為 undefined。

範例
下列範例在「輸出」面板中將文件陣列中第一份文件的路徑顯示為 file:/// URI 字串。您必須先儲存文件，再執行此指令碼。在範例中，檔案名稱為 test.fla，儲存在 Windows 電腦上的 My Documents 資料夾。

```
var filePathURI = flash.documents[0].pathURI;
fl.trace(filePathURI);
// displays file:///C|/Documents%20and%20Settings/<userName>/My%20Documents/test.fla
```

請參閱
document.path

document.publish()

適用版本
Flash MX 2004。

用法
```
document.publish()
```

參數
無。

傳回值
無。

說明
方法：根據作用中的發佈設定發佈文件 (「檔案 > 發佈設定」)。此方法等於選取「檔案 > 發佈」。

範例
下列範例發佈目前的文件：

```
fl.getDocumentDOM().publish();
```

document.publishProfiles

適用版本
Flash MX 2004。

用法
```
document.publishProfiles
```

說明
唯讀屬性；文件的發佈描述檔名稱陣列。

範例
下列範例顯示文件發佈描述檔的名稱：

```
var myPubProfiles = fl.getDocumentDOM().publishProfiles;
for (var i=0; i < myPubProfiles.length; i++){
     fl.trace(myPubProfiles[i]);
}
```

document.punch()

適用版本
Flash 8。

用法
```
document.punch()
```

參數
無。

傳回值
Boolean 值：如果成功的話，會傳回 true；否則便傳回 false。

說明
方法：使用上方選取的繪圖物件穿透其下方所有選取的繪圖物件。如果未選取繪圖物件，或任何選取項目不是繪圖物件，則此方法傳回 false。

範例
下列範例會穿透選取繪圖物件下方的繪圖物件：

```
fl.getDocumentDOM().punch();
```

請參閱
document.crop()、document.deleteEnvelope()、document.intersect()、document.union()、shape.isDrawingObject

document.removeAllFilters()

適用版本
Flash 8。

用法
```
document.removeAllFilters()
```

參數
無。

傳回值
無。

說明
方法；移除選取物件的所有濾鏡。

範例
下列範例移除選取物件的所有濾鏡：

```
fl.getDocumentDOM().removeAllFilters();
```

請參閱
document.addFilter()、document.changeFilterOrder()、document.disableAllFilters()、document.getFilters()、document.removeFilter()、Filter 物件

document.removeDataFromDocument()

適用版本
Flash MX 2004。

用法
```
document.removeDataFromDocument(name)
```

參數
name 字串，指定要移除的資料名稱。

傳回值
無。

說明
方法；移除文件附加的指定名稱永續性資料。

範例
下列範例從文件移除名稱為 "myData" 的永續性資料：

```
fl.getDocumentDOM().removeDataFromDocument("myData");
```

請參閱
document.addDataToDocument()、document.documentHasData()、document.getDataFromDocument()

document.removeDataFromSelection()

適用版本
Flash MX 2004。

用法
```
document.removeDataFromSelection(name)
```

參數
name 字串；指定要移除的永續性資料名稱。

傳回值
無。

說明
方法；移除選取範圍附加的指定名稱永續性資料。

範例
下列範例從選取範圍移除名稱為 "myData" 的永續性資料：

```
fl.getDocumentDOM().removeDataFromSelection("myData");
```

請參閱
document.addDataToSelection()

document.removeFilter()

適用版本
Flash 8。

用法
```
document.removeFilter(filterIndex)
```

參數
filterIndex 整數；指定選取從物件移除的濾鏡索引，從零開始。

傳回值
無。

說明
方法；從選取物件的 「濾鏡」清單移除第一個濾鏡。

範例
下列範例從選取物件的「濾鏡」清單移除第一個濾鏡 (索引值 0)：

```
fl.getDocumentDOM().removeFilter(0);
```

請參閱
document.addFilter()、document.changeFilterOrder()、document.disableFilter()、document.getFilters()、document.removeAllFilters()、Filter 物件

document.renamePublishProfile()

適用版本
Flash MX 2004。

用法

```
document.renamePublishProfile([profileNewName])
```

參數
profileNewName 選擇性參數；指定描述檔的新名稱。新名稱必須是唯一的。如果不指定名稱，則會提供預設名稱。

傳回值
Boolean 值：如果成功變更名稱的話，便傳回 true；否則傳回 false。

說明
方法；重新命名目前的描述檔。

範例
下列範例將目前的描述檔重新命名為預設名稱並顯示：

```
alert(fl.getDocumentDOM().renamePublishProfile());
```

document.renameScene()

適用版本
Flash MX 2004。

用法

```
document.renameScene(name)
```

參數
name 字串；指定場景的新名稱。

傳回值
Boolean 值：如果成功變更名稱的話，便傳回 true；否則傳回 false。例如，如果新名稱不為唯一的，則方法會傳回 false。

說明
方法；重新命名「場景」面板中目前選取的場景。選取場景的新名稱必須是唯一的。

範例
下列範例將目前場景重新命名為 "new name"：

```
var success = fl.getDocumentDOM().renameScene("new name");
```

document.reorderScene()

適用版本
Flash MX 2004。

用法

```
document.reorderScene(sceneToMove, sceneToPutItBefore)
```

參數
sceneToMove 整數；指定移動的場景，第一個場景為 0 (零)。

sceneToPutItBefore 整數；指定將 sceneToMove 所指定的場景移動至此場景之前。第一個場景請指定為 0 (零)。例如，如果指定 sceneToMove 為 1，sceneToPutItBefore 為 0，則第二個場景會放置在第一個場景之前。移動場景至結尾請指定 -1。

傳回值
無。

說明
方法；將指定場景移動到另一個指定場景之前。

範例
下列範例移動第二個場景至第一個場景之前：

```
fl.getDocumentDOM().reorderScene(1, 0);
```

document.resetOvalObject()

適用版本
Flash CS3 Professional。

用法

```
document.resetOvalObject()
```

參數
無。

傳回值
無。

說明
方法；將 「屬性」檢測器中的所有值都設定為預設的 Oval 物件設定。如果選取了任何 Oval 物件，其屬性也會一併重設為預設值。

範例
下列範例會將目前文件中的 Oval 物件屬性重設為預設值：

```
fl.getDocumentDOM().resetOvalObject();
```

請參閱
document.resetRectangleObject()

document.resetRectangleObject()

適用版本
Flash CS3 Professional。

用法

```
document.resetRectangleObject()
```

參數
無。

傳回值
無。

說明
方法；將「屬性」檢測器中的所有值都設定為預設的 Rectangle 物件設定。如果選取了任何 Rectangle 物件，其屬性也會一併重設為預設值。

範例
下列範例會將目前文件中的 Rectangle 物件屬性重設為預設值：

```
fl.getDocumentDOM().resetRectangleObject();
```

請參閱
document.resetOvalObject()

document.resetTransformation()

適用版本
Flash MX 2004。

用法

```
document.resetTransformation()
```

參數
無。

傳回值
無。

說明
方法：重設變形矩陣。此方法等於選取「修改 > 變形 > 移除變形」。

範例
下列範例重設目前選取範圍的變形矩陣：

```
fl.getDocumentDOM().resetTransformation();
```

document.revert()

適用版本
Flash MX 2004。

用法
```
document.revert()
```

參數
無。

傳回值
無。

說明
方法：將指定文件回復至先前儲存的版本。此方法等於選取「檔案 > 回復」。

範例
下列範例將指定文件回復至先前儲存的版本：

```
fl.getDocumentDOM().revert();
```

請參閱
document.canRevert()、fl.revertDocument()

document.rotate3DSelection()

適用版本
Flash CS4 Professional。

用法
```
document.rotate3DSelection(xyzCoordinate, bGlobalTransform)
```

參數
xyzCoordinate XYZ 座標點，指定 3D 旋轉的座標軸。

bGlobalTransform Boolean 值，指定變形模式應該是整體 (true) 或各自 (false) 的型態。

傳回值
無。

說明
方法：將 3D 旋轉套用至選取範圍。這個方法只能用於影片片段。

範例
在下列範例中，選取範圍先相對於舞台 (整體) 旋轉，然後相對於它本身 (各自) 旋轉。

```
var myDocument = fl.getDocumentDOM();
myDocument.rotate3DSelection({x:52.0, y:0, z:0}, true);
myDocument.rotate3DSelection({x:52.0, y:0, z:-55.2}, false);
```

document.rotateSelection()

適用版本
Flash MX 2004。

用法
```
document.rotateSelection(angle [, rotationPoint])
```

參數
angle 浮點值：指定旋轉的角度。

rotationPoint 字串：指定範圍框的旋轉邊。可接受的值為 "top right"、"top left"、"bottom right"、"bottom left"、"top center"、"right center"、"bottom center" 和 "left center"。如未指定，方法會使用變形點。這個參數是選擇性參數。

傳回值
無。

說明
方法：依指定度數旋轉選取範圍。特效等於使用 「自由變形」工具旋轉物件。

範例
下列範例會繞著變形點旋轉選取範圍 45 度：

```
fl.getDocumentDOM().rotateSelection(45);
```

下列範例會繞著左下角旋轉選取範圍 45 度：

```
fl.getDocumentDOM().rotateSelection(45, "bottom left");
```

document.save()

適用版本
Flash MX 2004。

用法

```
document.save([bOkToSaveAs])
```

參數

bOkToSaveAs 選擇性參數；如果為 true 或省略此參數，而且檔案尚未儲存，則開啟 「另存新檔」對話方塊。如果為 false 且檔案尚未儲存，則不儲存檔案。

傳回值

Boolean 值：如果成功完成儲存作業，則為 true，否則為 false。

說明

方法：將文件儲存至預設位置。此方法等於選取 「檔案 > 儲存檔案」。

若要指定檔案名稱 (而不是以相同名稱儲存)，請使用 fl.saveDocument()。

備註：如果檔案是新的，而且沒有修改或儲存，或是檔案自上次儲存後沒有修改，則此方法無效，且會傳回 false。若要針對未儲存或未修改的檔案進行儲存，請使用 document.saveAndCompact() 或 fl.saveDocumentAs()。

範例

下列範例將目前檔案儲存至預設位置：

```
fl.getDocumentDOM().save();
```

請參閱

document.saveAndCompact()、fl.saveAll()、fl.saveDocument()、fl.saveDocumentAs()

document.saveAndCompact()

適用版本

Flash MX 2004。

用法

```
document.saveAndCompact([bOkToSaveAs])
```

參數

bOkToSaveAs 選擇性參數；如果為 true 或忽略參數，而且檔案尚未儲存，則開啟 「另存新檔」對話方塊。如果為 false 且檔案尚未儲存，則不儲存檔案。預設值為 true。

傳回值

Boolean 值：如果成功完成儲存並壓縮作業，則為 true，否則為 false。

說明

方法：儲存並壓縮檔案。此方法等於選取 「檔案 > 儲存並壓縮」。

備註：如果檔案從未儲存，即使使用者取消 「另存新檔」對話方塊，這個方法仍會傳回 true。若要準確地確認檔案是否儲存，請使用 fl.saveDocumentAs()。

範例

下列範例儲存並壓縮目前的文件：

```
fl.getDocumentDOM().saveAndCompact();
```

請參閱
document.save()、fl.saveDocumentAs()、fl.saveDocument()、fl.saveAll()

document.scaleSelection()

適用版本
Flash MX 2004。

用法
```
document.scaleSelection(xScale, yScale [, whichCorner])
```

參數
xScale 浮點值；指定 x 的縮放量。

yScale 浮點值；指定 y 的縮放量。

whichCorner 字串值；指定變形發生的邊緣。如忽略，將以變形點為中心進行縮放。可接受的值為："bottom left"、"bottom right"、"top right"、"top left"、"top center"、"right center"、"bottom center" 和 "left center"。這個參數是選擇性參數。

傳回值
無。

說明
方法；將選取範圍縮放指定的量。此方法等於使用 「自由變形」縮放物件。

範例
下列範例將目前選取範圍的寬度擴展為原寬度的兩倍，並將高度縮減一半：

```
fl.getDocumentDOM().scaleSelection(2.0, 0.5);
```

下列範例會垂直翻轉選取範圍：

```
fl.getDocumentDOM().scaleSelection(1, -1);
```

下列範例會水平翻轉選取範圍：

```
fl.getDocumentDOM().scaleSelection(-1, 1);
```

下列範例會由上中將選取範圍垂直縮放 1.9 倍：

```
fl.getDocumentDOM().scaleSelection(1, 1.90, 'top center');
```

document.screenOutline

適用版本
Flash MX 2004。

用法
```
document.screenOutline
```

說明
唯讀屬性：文件目前的 ScreenOutline 物件。第一次存取物件之前，務必使用 document.allowScreens() 判斷屬性是否存在。

範例
下列範例在 screenOutline 屬性中顯示值陣列：

```
var myArray = new Array();
for(var i in fl.getDocumentDOM().screenOutline) {
    myArray.push(" "+i+" : "+fl.getDocumentDOM().screenOutline[i]) ;
}
fl.trace("Here is the property dump for screenOutline: "+myArray);
```

請參閱
document.allowScreens()

document.selectAll()

適用版本
Flash MX 2004。

用法

```
document.selectAll()
```

參數
無。

傳回值
無。

說明
方法：選取「舞台」上的所有項目。此方法等於按下 Control+A (Windows) 或 Command+A (Macintosh)，或是選取「編輯 > 全選」。

範例
下列範例選取目前使用者可見的所有項目：

```
fl.getDocumentDOM().selectAll();
```

請參閱
document.selection、document.selectNone()

document.selection

適用版本
Flash MX 2004。

用法

```
document.selection
```

說明

屬性：文件中的選取物件陣列。如果未選取，則傳回長度零的陣列。如果未開啟文件，則傳回 null。

若要增加物件至陣列，必須以下列任何一種方法選取：

- 手動選取「舞台」上的物件。
- 使用其中一種選取範圍方法，例如 document.setSelectionRect()、document.setSelectionBounds()、document.mouseClick()、document.mouseDblClk() 或 document.selectAll()。
- 手動選取一個或多個影格。
- 使用 Timeline 物件的其中一個方法選取一個或多個影格，例如 timeline.getSelectedFrames()、timeline.setSelectedFrames() 或 timeline.selectAllFrames()。
- 在特定影格中指定所有元素 (請參閱 Element 物件)。請參閱下面第一則範例。
- 建立一個或多個元素的陣列，然後將該陣列指定給 document.selection 陣列。請參閱下面第三則範例。

範例

下列範例將「影格 11」上的所有元素指定給目前的選取範圍 (請記住，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().currentFrame = 10;
fl.getDocumentDOM().selection = fl.getDocumentDOM().getTimeline().layers[0].frames[10].elements;
```

下列範例會在「舞台」左上角建立矩形，並在矩形下方建立文字字串。然後使用 document.setSelectionRect() 選取兩個物件，並將它們增加至 document.selection 陣列。最後在「輸出」面板上顯示 document.selection 內容。

```
fl.getDocumentDOM().addNewRectangle({left:0, top:0, right:99, bottom:99}, 0);
fl.getDocumentDOM().addNewText({left:-1, top:117.3, right:9.2, bottom:134.6});
fl.getDocumentDOM().setTextString('Hello World');
fl.getDocumentDOM().setSelectionRect({left:-28, top:-22, right:156.0, bottom:163});

var theSelectionArray = fl.getDocumentDOM().selection;

for(var i=0;i<theSelectionArray.length;i++){
fl.trace("fl.getDocumentDOM().selection["+i+"] = " + theSelectionArray[i]);
}
```

下列範例為進階範例，顯示如何重複圖層陣列和元素陣列，以找出並選取特定元件的實體。您可以延伸此範例，重複多個影格或場景。此範例將第一個影格影片片段 myMovieClip 的所有實體指定至目前的選取範圍：

```
// Assigns the layers array to the variable "theLayers".
var theLayers = fl.getDocumentDOM().getTimeline().layers;
// Creates an array to hold all the elements
// that are instances of "myMovieClip".
var myArray = new Array();
// Counter variable
var x = 0;
// Begin loop through all the layers.
for (var i = 0; i < theLayers.length; i++) {
    // Gets the array of elements in Frame 1
    // and assigns it to the array "theElems".
    var theElems = theLayers[i].frames[0].elements;
    // Begin loop through the elements on a layer.
    for (var c = 0; c < theElems.length; c++) {
        // Checks to see if the element is of type "instance".
        if (theElems[c].elementType == "instance") {
            // If the element is an instance, it checks
            // if it is an instance of "myMovieClip".
            if (theElems[c].libraryItem.name == "myMovieClip") {
                // Assigns elements that are instances of "myMovieClip" to "myArray".
                myArray[x] = theElems[c];
                // Increments counter variable.
                x++;
            }
        }
    }
}
// Now that you have assigned all the instances of "myMovieClip"
// to "myArray", you then set the document.selection array
// equal to myArray. This selects the objects on the Stage.
fl.getDocumentDOM().selection = myArray;
```

document.selectNone()

適用版本

Flash MX 2004。

用法

```
document.selectNone()
```

參數

無。

傳回值

無。

說明

方法：取消全選任何的選取項目。

範例

下列範例取消全選任何的選取項目：

```
fl.getDocumentDOM().selectNone();
```

請參閱
document.selectAll()、document.selection

document.setAlignToDocument()

適用版本
Flash MX 2004。

用法
```
document.setAlignToDocument(bToStage)
```

參數
bToStage Boolean 值，如果設定為 true，物件會對齊 「舞台」。如果設定為 false，則不作用。

傳回值
無。

說明
方法；設定 document.align()、document.distribute()、document.match() 和 document.space() 在文件中作用的偏好設定。此方法等於啟用 「對齊」面板中的 「對齊舞台」按鈕。

範例
下列範例啟用 「對齊」面板中的 「對齊舞台」按鈕，讓物件對齊 「舞台」：

```
fl.getDocumentDOM().setAlignToDocument(true);
```

請參閱
document.getAlignToDocument()

document.setBlendMode()

適用版本
Flash 8。

用法
```
document.setBlendMode(mode)
```

參數
mode 字串；代表選取物件所需的混合模式。可接受的值為 "normal"、"layer"、"multiply"、"screen"、"overlay"、"hardlight"、"lighten"、"darken"、"difference"、"add"、"subtract"、"invert"、"alpha" 和 "erase"。

傳回值
無。

說明
方法；設定選取物件的混合模式。

範例
下列範例設定選取物件的混合模式為 "add"。

```
fl.getDocumentDOM().setBlendMode("add");
```

請參閱
document.addFilter()、document.setFilterProperty()、symbolInstance.blendMode

document.setCustomFill()

適用版本
Flash MX 2004。

用法

```
document.setCustomFill(fill)
```

參數
fill 指定要使用的填色設定的 Fill 物件。請參閱 Fill 物件。

傳回值
無。

說明
方法：設定 「工具」面板、「屬性」檢測器和任何選取形狀的填色設定。可讓指令碼在繪製物件之前設定填色設定；而不是繪製物件、選取物件，然後再變更填色設定。也可以讓指令碼變更 「工具」面板和 「屬性」檢測器的填色設定。

範例
下列範例將 「工具」面板、「屬性」檢測器和任何選取形狀中的填色色票顏色變更為白色：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.color = '#FFFFFF';
fill.style = "solid";
fl.getDocumentDOM().setCustomFill(fill);
```

請參閱
document.getCustomFill()

document.setCustomStroke()

適用版本
Flash MX 2004。

用法

```
document.setCustomStroke(stroke)
```

參數
stroke Stroke 物件。

傳回值
無。

說明
方法；設定 「工具」面板、「屬性」檢測器和任何選取形狀的筆畫設定。可讓指令碼在繪製物件之前設定筆畫設定；而不是繪製物件、選取物件，然後再變更筆畫設定。也可讓指令碼變更 「工具」面板和 「屬性」檢測器的筆畫設定。

範例
下列範例變更 「工具」面板、「屬性」檢測器和任何選取形狀的筆畫粗細設定：

```
var stroke = fl.getDocumentDOM().getCustomStroke();
stroke.thickness += 2;
fl.getDocumentDOM().setCustomStroke(stroke);
```

請參閱
document.getCustomStroke()

document.setElementProperty()

適用版本
Flash MX 2004。

用法
```
document.setElementProperty(property, value)
```

參數
property 字串，指定要設定之 Element 屬性的名稱。如需屬性和值的完整清單，請參閱 Element 物件的屬性摘要表。

您無法使用此方法設定唯讀屬性的值，例如 element.elementType、element.top 或 element.left。

value 整數；指定所指定 Element 屬性中設定的值。

傳回值
無。

說明
方法；設定文件中選取物件的指定 Element 屬性。如果沒有選取範圍，則此方法不會執行任何動作。

範例
下列範例將所有選取物件的寬度設定為 100，高度設定為 50：

```
fl.getDocumentDOM().setElementProperty("width", 100);
fl.getDocumentDOM().setElementProperty("height", 50);
```

document.setElementTextAttr()

適用版本
Flash MX 2004。

用法

```
document.setElementTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

參數

attrName 字串；指定要變更的 TextAttrs 屬性名稱。

attrValue 設定 TextAttrs 屬性的值。如需屬性名稱和期望值的清單，請參閱 TextAttrs 物件的屬性摘要表。

startIndex 整數值；指定受到影響的第一個字元索引。這個參數是選擇性參數。

endIndex 整數值；指定受到影響的最後一個字元索引。這個參數是選擇性參數。

傳回值

Boolean 值：如果至少有一個變更的文字特質屬性，則傳回 true，否則為 false。

說明

方法；將選取文字項目的指定 textAttrs 屬性設定為指定值。如需屬性名稱和允許值的清單，請參閱 TextAttrs 物件的屬性摘要表。如果未傳遞選擇性參數，方法會設定目前選取文字範圍的樣式；如果未選取任何的文字，則設定整個文字欄位的樣式。如果只傳遞 startIndex，方法會設定字元的特質。如果傳遞 startIndex 和 endIndex，方法會設定從 startIndex 開始到 endIndex (但不包括 endIndex) 的字元特質。如果指定段落樣式，則落在範圍內的段落都會受到影響。

範例

下列範例設定選取文字項目的 fillColor、italic 和 bold 文字特質：

```
var success = fl.getDocumentDOM().setElementTextAttr("fillColor", "#00ff00");
var pass = fl.getDocumentDOM().setElementTextAttr("italic", true, 10);
var ok = fl.getDocumentDOM().setElementTextAttr("bold", true, 5, 15);
```

document.setFillColor()

適用版本

Flash MX 2004。

用法

```
document.setFillColor(color)
```

參數

color 填色的顏色，格式為下列其中一種：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

  如果設定為 null，則不會設定填色顏色，等於將使用者介面中的 「填色」色票設定為無填色。

傳回值

無。

說明

方法；變更選取範圍的填色顏色為指定顏色。如需有關變更 「工具」面板和 「屬性」檢測器填色顏色的詳細資訊，請參閱 document.setCustomFill()。

範例
在下列範例中，前三個陳述式使用每種不同的指定顏色格式來設定填色顏色。第四個陳述式將填色設定為無填色。

```
fl.getDocumentDOM().setFillColor("#cc00cc");
fl.getDocumentDOM().setFillColor(0xcc00cc);
fl.getDocumentDOM().setFillColor(120000);
fl.getDocumentDOM().setFillColor(null);
```

document.setFilterProperty()

適用版本
Flash 8。

用法
```
document.setFilterProperty(property, filterIndex, value)
```

參數
property 字串；指定設定的屬性。可接受的值為 "blurX"、"blurY"、"quality"、"angle"、"distance"、"strength"、"knockout"、"inner"、"bevelType"、"color"、"shadowColor" 和 "highlightColor"。

filterIndex 整數；指定 「濾鏡」清單中的濾鏡索引，從零開始。

value 數字或字串；指定所指定濾鏡屬性的設定值。可接受的值取決於設定的屬性和濾鏡。

傳回值
無。

說明
方法；設定目前選取物件的指定濾鏡屬性 (假設此物件支援指定濾鏡)。

範例
下列範例會將選取物件 「濾鏡」清單中的第二個濾鏡 (索引值為 1) 的 quality 屬性設定為 2，然後設定選取物件上 「濾鏡」清單中第一個濾鏡的 shadowColor 屬性：

```
fl.getDocumentDOM().setFilterProperty("quality", 1, 2);
fl.getDocumentDOM().setFilterProperty("shadowColor", 0, "#FF00FF");
```

請參閱
document.addFilter()、document.getFilters()、document.setBlendMode()、document.setFilters()、Filter 物件

document.setFilters()

適用版本
Flash 8。

用法
```
document.setFilters(filterArray)
```

參數
filterArray　目前指定的濾鏡陣列。

傳回值
無。

說明
方法；套用濾鏡至選取物件。在呼叫 document.getFilters() 並對濾鏡進行想要的變更之後，使用這個方法。

範例
下列範例取得選取物件中的濾鏡，並將所有 Blur 濾鏡的 blurX 屬性設定為 50：

```
var myFilters = fl.getDocumentDOM().getFilters();
for (i=0; i < myFilters.length; i++) {
    if (myFilters[i].name == "blurFilter"){
        myFilters[i].blurX = 50;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.addFilter()、document.getFilters()、document.setFilterProperty()、Filter 物件

document.setInstanceAlpha()

適用版本
Flash MX 2004。

用法
```
document.setInstanceAlpha(opacity)
```

參數
opacity　整數；介於 0 (透明) 和 100 (完全飽和) 之間，用來調整實體透明度。

傳回值
無。

說明
方法；設定實體的不透明度。

範例
下列範例將著色的不透明度值設定為 50：

```
fl.getDocumentDOM().setInstanceAlpha(50);
```

document.setInstanceBrightness()

適用版本
Flash MX 2004。

用法
```
document.setInstanceBrightness(brightness)
```

參數
brightness 整數，指定亮度值為 -100 (黑色) 至 100 (白色)。

傳回值
無。

說明
方法；設定實體的亮度。

範例
下列範例將實體的亮度值設定為 50：
```
fl.getDocumentDOM().setInstanceBrightness(50);
```

document.setInstanceTint()

適用版本
Flash MX 2004。

用法
```
document.setInstanceTint( color, strength )
```

參數
color 色調的顏色，格式為下列其中一種：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

strength 整數；介於 0 和 100 之間，指定著色的不透明度。

傳回值
無。

說明
方法；設定實體的著色。

範例
下列範例將選取實體的著色設定為紅色，不透明度值為 50：

```
fl.getDocumentDOM().setInstanceTint(0xff0000, 50);
```

document.setMetadata()

適用版本
Flash 8。

用法
```
document.setMetadata(strMetadata)
```

參數
strMetadata 字串；包含文件相關的 XML 中繼資料。如需詳細資訊，請參閱下列說明。

傳回值
Boolean 值：如果成功的話，會傳回 true；否則便傳回 false。

說明
方法：設定指定文件的 XML 中繼資料，覆寫任何現有的中繼資料。會驗證以 strMetadata 傳遞的 XML，可能於儲存前重新寫入。如果無法驗證為合法的 XML，或者違反特定規則，則不會設定 XML 中繼資料，且會傳回 false (如果傳回 false，則無法取得更詳細的錯誤資訊)。

備註：即使傳回 true，XML 設定也未必完全等於傳入的字串。若要取得與設定的 XML 完全相同的值，請使用 document.getMetadata()。

中繼資料的格式是與 XMP 規格相容的 RDF。如需有關 RDF 和 XMP 的詳細資訊，請參閱下列來源：

- RDF 入門書，網址為 www.w3.org/TR/rdf-primer/
- RDF 規格，網址為 www.w3.org/TR/1999/REC-rdf-syntax-19990222/
- XMP 首頁，網址為 www.adobe.com/products/xmp/

範例
下列範例顯示幾個呈現相同資料的不同合法方式。在這所有的案例中 (第二個案例除外)，如果資料傳送到 Document.setMetadata()，就不會重新寫入 (除了移除斷行符號之外)。

在第一個範例中，中繼資料位於標籤內，其中不同的資料結構分放在不同的 rdf:Description 標籤內：

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/'>
<dc:title>Simple title</dc:title>
<dc:description>Simple description</dc:description>
</rdf:Description>
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'>
<xmp:CreateDate>2004-10-12T10:29-07:00</xmp:CreateDate>
<xmp:CreatorTool>Flash Authoring WIN 8,0,0,215</xmp:CreatorTool>
</rdf:Description>
</rdf:RDF>
```

在第二個範例中，中繼資料也在標籤內，不過所有不同的結構描述都在一個 rdf:Description 標籤內。這個範例也包括註解，Document.setMetadata() 會忽略並捨棄註解：

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <!-- This is before the first rdf:Description tag -->
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/'>
<dc:title>Simple title</dc:title>
<dc:description>Simple description</dc:description>
</rdf:Description>
    <!-- This is between the two rdf:Description tags -->
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'>
<xmp:CreateDate>2004-10-12T10:29-07:00</xmp:CreateDate>
<xmp:CreatorTool>Flash Authoring WIN 8,0,0,215</xmp:CreatorTool>
</rdf:Description>
    <!-- This is after the second rdf:Description tag -->
</rdf:RDF>
```

在第三個範例中，中繼資料在特質內，而且所有不同的結構描述都在一個 rdf:Description 標籤內：

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<rdf:Description rdf:about='' xmlns:dc='http://purl.org/dc/1.1/' dc:title='Simple title'
dc:description='Simple description' />
<rdf:Description rdf:about='' xmlns:xmp='http://ns.adobe.com/xap/1.0/'
xmp:CreateDate='2004-10-12T10:29-07:00' xmp:CreatorTool='Flash Authoring WIN 8,0,0,215' />
</rdf:RDF>
```

請參閱
document.getMetadata()

document.setMobileSettings()

適用版本
Flash CS3 Professional。

用法

```
document.setMobileSettings(xmlString)
```

參數
xmlString 字串，描述行動裝置 FLA 檔中的 XML 設定。

傳回值
如果已成功設定，就會傳回 true，否則便傳回 false。

說明
方法：在行動裝置 FLA 檔中設定 XML 設定字串的值 (大部分行動裝置 FLA 檔都具有描述文件設定的 XML 字串)。

範例
下列範例會設定行動裝置 FLA 檔的 XML 設定字串。請注意，下列範例代表單行程式碼。

```
fl.getDocumentDOM().setMobileSettings("<? xml version="1.0" encoding="UTF-16" standalone="no" ?>
<mobileSettings> <contentType id="standalonePlayer" name="Standalone Player"/> <testDevices> <testDevice
id="1170" name="Generic Phone" selected="yes"/> </testDevices> <outputMsgFiltering info="no" trace="yes"
warning="yes"/> <testWindowState height="496" splitterClosed="No" splitterXPos="400" width="907"/>
</mobileSettings>");
```

請參閱
document.getMobileSettings()

document.setOvalObjectProperty()

適用版本
Flash CS3 Professional。

用法
```
document.setOvalObjectProperty(propertyName, value)
```

參數
propertyName 字串，指定要設定的屬性。如需可接受的值，請參閱 Oval 物件的屬性摘要表。

value 指定給屬性的值。可接受的值取決於您在 propertyName 中指定的屬性。

傳回值
無。

說明
方法；指定基本 Oval 物件的指定屬性值。

範例
請參閱 Oval 物件中的個別屬性以取得範例。

請參閱
Oval 物件、shape.isOvalObject

document.setPlayerVersion()

適用版本
Flash CS3 Professional。

用法
```
document.setPlayerVersion(version)
```

參數
version 字串，代表指定文件的目標 Flash Player 版本。可接受的值為 "FlashLite"、"FlashLite11"、"FlashLite20"、"FlashLite30"、"1"、"2"、"3"、"4"、"5"、"6"、"7"、"8"、"9"、"FlashPlayer10" 和 "AdobeAIR1_1"。

傳回值
如果已成功設定播放程式版本，就會傳回 true 值，否則就會傳回 false。

說明
方法；設定指定文件的目標 Flash Player 版本。這個值與 「發佈設定」對話方塊中設定的值相同。

範例
下列範例會將 Flash Player 6 指定為目前文件的目標播放程式版本：

```
fl.getDocumentDOM().setPlayerVersion("6");
```

請參閱
document.getPlayerVersion()

document.setRectangleObjectProperty()

適用版本
Flash CS3 Professional。

用法
```
document.setRectangleObjectProperty(propertyName, value)
```

參數
propertyName 字串，指定要設定的屬性。如需可接受的值，請參閱 Rectangle 物件的屬性摘要表。

value 指定給屬性的值。可接受的值取決於您在 propertyName 中指定的屬性。

傳回值
無。

說明
方法；指定基本 Rectangle 物件的指定屬性值。

範例
請參閱 Rectangle 物件中的個別屬性以取得範例。

請參閱
Rectangle 物件、shape.isRectangleObject

document.setSelectionBounds()

適用版本
Flash MX 2004；在 Flash 8 中已加入 bContactSensitiveSelection 參數。

用法
```
document.setSelectionBounds(boundingRectangle [, bContactSensitiveSelection])
```

參數
boundingRectangle 矩形；指定選取範圍的新位置和大小。如需有關 boundingRectangle 格式的詳細資訊，請參閱 document.addNewRectangle()。

bContactSensitiveSelection Boolean 值；指定在選取物件時是否啟用接觸感應選取模式 (true) 或加以停用 (false)。預設值為 false。

傳回值
無。

說明

方法：移動單一作業的選取範圍並調整其大小。

如果傳遞 bContactSensitiveSelection 的值，則只對這個方法有效，而不會影響文件的接觸感應選取模式 (請參閱 fl.contactSensitiveSelection)。

範例

下列範例將目前選取範圍移動到 10, 20，並將大小調整為 100, 200：

```
var l = 10;
var t = 20;
fl.getDocumentDOM().setSelectionBounds({left:l, top:t, right:(100+l), bottom:(200+t)});
```

請參閱

document.selection、document.setSelectionRect()

document.setSelectionRect()

適用版本

Flash MX 2004；在 Flash 8 中已加入 bContactSensitiveSelection 參數。

用法

```
document.setSelectionRect(rect [, bReplaceCurrentSelection [, bContactSensitiveSelection]])
```

參數

rect 設定為選取的矩形物件。如需有關 rect 格式的詳細資訊，請參閱 document.addNewRectangle()。

bReplaceCurrentSelection Boolean 值，指定方法是否取代目前的選取範圍 (true) 或加入目前的選取範圍 (false)。預設值為 true。

bContactSensitiveSelection Boolean 值；指定在選取物件時是否啟用接觸感應選取模式 (true) 或加以停用 (false)。預設值為 false。

傳回值

無。

說明

方法：使用指定座標繪製相對於 「舞台」的矩形選取圈選範圍，與 document.getSelectionRect() 不同；它的矩形是相對於被編輯的物件。

此方法等於使用 「選取」工具拖曳矩形。實體必須完全在選取的矩形內。

如果傳遞 bContactSensitiveSelection 的值，則只對這個方法有效，而不會影響文件的接觸感應選取模式 (請參閱 fl.contactSensitiveSelection)。

備註：使用 「操作記錄」面板或選單項目重複 setSelectionRect()，會重複 setSelectionRect() 作業之前的步驟。

範例

在下列範例中，第二個選取範圍會取代第一個選取範圍：

```
fl.getDocumentDOM().setSelectionRect({left:1, top:1, right:200, bottom:200});
fl.getDocumentDOM().setSelectionRect({left:364.0, top:203.0, right:508.0, bottom:434.0}, true);
```

在下列範例中，第二個選取範圍會增加到第一個選取範圍。這與按住 Shift 並選取第二個物件的手動操作相同。

```
fl.getDocumentDOM().setSelectionRect({left:1, top:1, right:200, bottom:200});
fl.getDocumentDOM().setSelectionRect({left:364.0, top:203.0, right:508.0, bottom:434.0}, false);
```

請參閱

document.getSelectionRect()、document.selection、document.setSelectionBounds()

document.setStageVanishingPoint()

適用版本

Flash CS4 Professional。

用法

```
document.setStageVanishingPoint(point)
```

參數

point 點，指定要設定檢視 3D 物件之消失點的 x 和 y 座標位置。

傳回值

無。

說明

指定檢視 3D 物件的消失點。

範例

下列範例會設定 「舞台」消失點：

```
fl.getDocumentDOM().setStageVanishingPoint({x:45, y:45});
```

document.setStageViewAngle()

適用版本

Flash CS4 Professional。

用法

```
document.setStageViewAngle(angle)
```

參數

angle 介於 0.0 和 179.0 之間的浮點值。

傳回值

無。

說明

指定檢視 3D 物件的透視角度。

範例
下列範利會將 「舞台」透視角度設定為 70 度：

```
fl.getDocumentDOM().setStageViewAngle(70);
```

document.setStroke()

適用版本
Flash MX 2004。

用法

```
document.setStroke(color, size, strokeType)
```

參數
color 筆畫的顏色，格式為下列其中一種：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

size 浮點值；指定選取範圍的新筆畫大小。

strokeType 字串；指定選取範圍的新筆畫類型。可接受的值為 "hairline"、"solid"、"dashed"、"dotted"、"ragged"、"stipple" 和 "hatched"。

傳回值
無。

說明
方法；設定選取筆畫的顏色、寬度和樣式。如需有關在 「工具」面板和 「屬性」檢測器中變更筆畫的詳細資訊，請參閱 document.setCustomStroke()。

範例
下列範例將筆畫顏色設定為紅色，大小設定為 3.25，類型設定為虛線：

```
fl.getDocumentDOM().setStroke("#ff0000", 3.25, "dashed");
```

document.setStrokeColor()

適用版本
Flash MX 2004。

用法

```
document.setStrokeColor(color)
```

參數
color 筆畫的顏色，格式為下列其中一種：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串

- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

傳回值
無。

說明
方法：變更選取範圍的筆畫顏色為指定顏色。如需有關在 「工具」面板和 「屬性」檢測器中變更筆畫的詳細資訊，請參閱 document.setCustomStroke()。

範例
在下列範例中，三個陳述式使用指定顏色的格式來設定筆畫顏色。

```
fl.getDocumentDOM().setStrokeColor("#cc00cc");
fl.getDocumentDOM().setStrokeColor(0xcc00cc);
fl.getDocumentDOM().setStrokeColor(120000);
```

document.setStrokeSize()

適用版本
Flash MX 2004。

用法
```
document.setStrokeSize(size)
```

參數
size 浮點值；介於 0.25 至 10 之間，用來指定筆畫大小。方法會忽略兩位數以上的小數位數。

傳回值
無。

說明
方法：變更選取範圍的筆畫大小為指定大小。如需有關在 「工具」面板和 「屬性」檢測器中變更筆畫的詳細資訊，請參閱 document.setCustomStroke()。

範例
下列範例將選取範圍的筆畫大小變更為 5：

```
fl.getDocumentDOM().setStrokeSize(5);
```

document.setStrokeStyle()

適用版本
Flash MX 2004。

用法
```
document.setStrokeStyle(strokeType)
```

參數
strokeType 字串；指定目前選取範圍的筆畫類型。可接受的值為 "hairline"、"solid"、"dashed"、"dotted"、"ragged"、"stipple" 和 "hatched"。

傳回值
無。

說明
方法：變更選取範圍的筆畫樣式為指定樣式。如需有關在 「工具」面板和 「屬性」檢測器中變更筆畫的詳細資訊，請參閱 document.setCustomStroke()。

範例
下列範例將選取範圍的筆畫樣式變更為 "dashed"：

```
fl.getDocumentDOM().setStrokeStyle("dashed");
```

document.setTextRectangle()

適用版本
Flash MX 2004。

用法
```
document.setTextRectangle(boundingRectangle)
```

參數
boundingRectangle 矩形，指定文字項目排列範圍的新大小。如需有關 boundingRectangle 格式的詳細資訊，請參閱 document.addNewRectangle()。

傳回值
Boolean 值：如果至少有一個文字欄位的大小變更，則傳回 true，否則為 false。

說明
方法：變更選取文字項目的矩形邊界為指定大小。此方法讓文字在新矩形內重新排列；文字項目不會縮放或變形。傳入 boundingRectangle 的值應用如下：

- 如果文字為水平且靜態，則方法僅考慮傳入 boundingRectangle 的寬度值；系統會自動計算高度來符合所有的文字。
- 如果文字為垂直 (因而是靜態的)，則方法僅考慮傳入 boundingRectangle 的高度值；系統會自動計算寬度來符合所有的文字。
- 如果文字是動態或輸入的，方法會將傳入 boundingRectangle 的寬度和高度值都列入考慮，產生的矩形可能會大於所需大小以符合所有文字。不過，如果參數指定的矩形大小太小以致於無法容納所有文字，此方法就只會考慮傳入 boundingRectangle 的寬度值 (而會自動計算高度來符合所有的文字)。

範例
下列範例將文字矩形邊界的大小變更為指定尺寸：

```
fl.getDocumentDOM().setTextRectangle({left:0, top:0, right:50, bottom:200})
```

document.setTextSelection()

適用版本
Flash MX 2004。

用法
```
document.setTextSelection(startIndex, endIndex)
```

參數
startIndex 整數；指定第一個字元的選取位置。第一個字元位置為 0 (零)。

endIndex 整數；指定選取範圍的結束位置，最大為 endIndex，但不包括 endIndex。第一個字元位置為 0 (零)。

傳回值
Boolean 值：如果方法可以成功設定文字選取範圍，則為 true，否則為 false。

說明
方法；將目前選取文字欄位的文字選取範圍設定為 startIndex 和 endIndex 指定的值。如果文字編輯尚未啟用，便會啟用。

範例
下列範例從第六個字元至到第二十五個字元之間選取文字：

```
fl.document.setTextSelection(5, 25);
```

document.setTextString()

適用版本
Flash MX 2004。

用法
```
document.setTextString(text [, startIndex [, endIndex]])
```

參數
text 插入文字欄位的字元字串。

startIndex 整數，指定第一個取代的字元。第一個字元位置為 0 (零)。這個參數是選擇性參數。

endIndex 整數；指定最後一個取代的字元。這個參數是選擇性參數。

傳回值
Boolean 值：如果至少設定一個文字字串的文字，則為 true，否則為 false。

說明
方法；插入文字字串。如果未傳遞選擇性參數，則會取代現有的文字選取範圍；如果目前未編輯 Text 物件，則會取代整個文字字串。如果僅傳遞 startIndex，則會在此位置插入傳遞的字串。如果傳遞 startIndex 和 endIndex，傳遞的字串會取代從 startIndex 到 endIndex 的文字片段，但不包括 endIndex。

範例
下列範例以 "Hello World" 取代目前的文字選取範圍：

```
var success = fl.getDocumentDOM().setTextString("Hello World!");
```

下列範例會在目前文字選取範圍中的位置 6 插入 "hello"：

```
var pass = fl.getDocumentDOM().setTextString("hello", 6);
```

下列範例會在目前文字選取範圍中的位置 2 到位置 7 之間插入 "Howdy"，但不包括位置 7：

```
var ok = fl.getDocumentDOM().setTextString("Howdy", 2, 7);
```

請參閱
document.getTextString()

document.setTransformationPoint()

適用版本
Flash MX 2004。

用法

```
document.setTransformationPoint( transformationPoint )
```

參數
transformationPoint 點 (例如，{x:10, y:20}，其中 x 和 y 是浮點數)，指定下列每個元素之變形點的值：

- 形狀：transformationPoint 的設定是相對於文件 (0,0 為「舞台」的左上角)。
- 元件：transformationPoint 的設定是相對於元件的註冊點 (0,0 位於註冊點)。
- 文字：transformationPoint 的設定是相對於文字欄位 (0,0 為文字欄位的左上角)。
- 點陣圖 / 視訊：transformationPoint 的設定是相對於點陣圖 / 視訊 (0,0 代表點陣圖或視訊的左上角)。
- 繪圖物件、基本橢圓形和矩形以及群組：transformationPoint 的設定是相對於文件 (0,0 為「舞台」的左上角)。若要設定 transformationPoint 使其相對於物件、基本物件或群組中心點，請使用 element.setTransformationPoint()。

傳回值
無。

說明
方法；設定目前選取範圍之變形點的位置。

範例
下列範例將目前選取範圍的變形點設定為 100, 200：

```
fl.getDocumentDOM().setTransformationPoint({x:100, y:200});
```

請參閱
document.getTransformationPoint()、element.setTransformationPoint()

document.silent

適用版本
Flash MX 2004。

用法
```
document.silent
```

說明
屬性；Boolean 值；指定物件是否可存取。等於 「輔助功能」面板 「讓影片支援輔助功能」設定的反轉邏輯。也就是說，如果 document.silent 為 true，則等於未選取 「讓影片支援輔助功能」選項。如果為 false，則等於選取 「讓影片支援輔助功能」選項。

範例
下列範例將 isSilent 變數設定為 silent 屬性的值：

```
var isSilent = fl.getDocumentDOM().silent;
```

下列範例會將 silent 屬性設定為 false，表示文件可存取：

```
fl.getDocumentDOM().silent = false;
```

document.skewSelection()

適用版本
Flash MX 2004。

用法
```
document.skewSelection(xSkew, ySkew [, whichEdge])
```

參數
xSkew 浮點數值；指定 x 的傾斜量，以度為單位。

ySkew 浮點數值；指定 y 的傾斜量，以度為單位。

whichEdge 字串；指定變形的邊緣，如忽略，會在變形點斜切。可接受的值為 "top center"、"right center"、"bottom center" 和 "left center"。這個參數是選擇性參數。

傳回值
無。

說明
方法；將選取範圍斜切指定量。特效等於使用 「自由變形」工具斜切物件。

範例
下列範例的選取物件將垂直斜切 2.0 度、水平斜切 1.5 度。第二個範例的物件在上中邊緣變形：

```
fl.getDocumentDOM().skewSelection(2.0, 1.5);
fl.getDocumentDOM().skewSelection(2.0, 1.5, "top center");
```

document.smoothSelection()

適用版本
Flash MX 2004。

用法
```
document.smoothSelection()
```

參數
無。

傳回值
無。

說明
方法：讓每個選取的填色外框或曲線線段的曲線平滑化。此方法執行的動作與 「工具」面板中 「平滑化」按鈕相同。

範例
下列範例會讓目前選取範圍的曲線平滑化：

```
fl.getDocumentDOM().smoothSelection();
```

document.sourcePath

適用版本
Flash CS4 Professional。

用法
```
document.sourcePath
```

說明
屬性：字串，其中包含文件 ActionScript 3.0 來源路徑中的項目清單，指定 ActionScript 類別檔案的位置。此字串中的項目會以分號隔開。在編寫工具中，藉由選擇 「檔案 > 發佈設定」，然後選擇 「Flash」索引標籤上的 「ActionScript 3.0 指令碼設定」來指定項目。

範例
下列範例會將 ./Class files 資料夾加入至文件的來源路徑：

```
var myDoc = fl.getDocumentDOM();
fl.trace(myDoc.sourcePath);
myDoc.sourcePath = "./Class files;" + myDoc.sourcePath;
fl.trace(myDoc.sourcePath);
```

請參閱
document.externalLibraryPath、document.libraryPath、fl.sourcePath

document.space()

適用版本
Flash MX 2004。

用法
```
document.space(direction [, bUseDocumentBounds])
```

參數
direction 字串；指定選取範圍中的物件分散方向。可接受的值為 "horizontal" 或 "vertical"。

bUseDocumentBounds Boolean 值；如果設定為 true，則依文件邊界分散物件。否則，方法會使用選取物件的邊界。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法；平均選取範圍中的物件的間隔。

範例
下列範例依「舞台」相對位置水平分散物件：

```
fl.getDocumentDOM().space("horizontal",true);
```

下列範例會依物件彼此的相對位置水平分散物件：

```
fl.getDocumentDOM().space("horizontal");
```

下列範例會依物件彼此的相對位置水平分散物件，並將 bUseDcoumentBounds 明確設定為 false：

```
fl.getDocumentDOM().space("horizontal",false);
```

請參閱
document.getAlignToDocument()、document.setAlignToDocument()

document.straightenSelection()

適用版本
Flash MX 2004。

用法
```
document.straightenSelection()
```

參數
無。

傳回值
無。

說明
方法；直線化目前選取的筆畫。此方法等於使用 「工具」面板的 「直線化」按鈕。

範例
下列範例會讓目前選取範圍的曲線直線化：

```
fl.getDocumentDOM().straightenSelection();
```

document.swapElement()

適用版本
Flash MX 2004。

用法

```
document.swapElement(name)
```

參數
name 字串；指定使用的元件庫項目名稱。

傳回值
無。

說明
方法；以指定的選取範圍替換目前的選取範圍。選取範圍必須包含圖像、按鈕、影片片段、視訊或點陣圖。若未選取任何物件或找不到指定物件，則此方法會顯示錯誤訊息。

範例
下列範例將目前選取範圍替換為文件庫的 Symbol 1：

```
fl.getDocumentDOM().swapElement('Symbol 1');
```

document.swapStrokeAndFill()

適用版本
Flash 8。

用法

```
document.swapStrokeAndFill()
```

參數
無。

傳回值
無。

說明
方法；替換筆畫和填色。

範例
下列範例會替換目前文件中的筆劃和填色。

```
fl.getDocumentDOM().swapStrokeAndFill();
```

document.testMovie()

適用版本
Flash MX 2004。

用法
```
document.testMovie([Boolean abortIfErrorsExist])
```

參數
abortIfErrorsExist Boolean 值；預設值是 false。如果設定為 true，將不會啟動測試影片工作階段，而如果發生編譯器錯誤，將不會開啟 .swf 視窗。編譯器警告將不會中止命令。在 Flash Professional CS5 中已加入此參數。

傳回值
無。

說明
方法；在文件中執行 「測試影片」作業。

範例
下列範例會測試目前文件的影片，但如果發生編譯器錯誤，就會中止測試影片：

```
fl.getDocumentDOM().testMovie(1);
```

請參閱
document.canTestMovie()、document.testScene()

document.testScene()

適用版本
Flash MX 2004。

用法
```
document.testScene()
```

參數
無。

傳回值
無。

說明
方法；在文件的目前場景中執行 「測試場景」作業。

範例
下列範例測試文件中目前的選取場景：

```
fl.getDocumentDOM().testScene();
```

請參閱
document.canTestScene()、document.testMovie()

document.timelines

適用版本
Flash MX 2004。

用法
```
document.timelines
```

說明
唯讀屬性；Timeline 物件陣列 (請參閱 Timeline 物件)。

範例
下列範例取得作用中文件的目前時間軸陣列，並在 「輸出」面板上顯示其名稱：

```
var i = 0;
var curTimelines = fl.getDocumentDOM().timelines;
while(i < fl.getDocumentDOM().timelines.length){
    alert(curTimelines[i].name);
    ++i;
}
```

請參閱
document.currentTimeline、document.getTimeline()

document.traceBitmap()

適用版本
Flash MX 2004。

用法
```
document.traceBitmap(threshold, minimumArea, curveFit, cornerThreshold)
```

參數
threshold 整數；控制轉換成向量圖中的顏色數目。可接受的值是介於 0 到 500 之間的整數。

minimumArea 整數；指定半徑，以像素為單位。可接受的值是介於 1 到 1000 之間的整數。

curveFit 字串；指定繪製外框的平滑度。可接受的值為 "pixels"、"very tight"、"tight"、"normal"、"smooth" 和 "very smooth"。

cornerThreshold 類似 curveFit 的字串，但適用於點陣圖影像的轉角。可接受的值為 "many corners"、"normal" 和 "few corners"。

傳回值
無。

說明
方法；在目前選取範圍執行轉換成向量圖。此方法等於選取 「修改 > 點陣圖 > 轉換成向量圖」。

範例
下列範例使用指定參數追蹤選取的點陣圖：

```
fl.getDocumentDOM().traceBitmap(0, 500, 'normal', 'normal');
```

document.translate3DCenter()

適用版本
Flash CS4 Professional。

用法

```
document.translate3DCenter(xyzCoordinate)
```

參數
xyzCoordinate XYZ 座標點，指定 3D 旋轉或轉譯的中心點。

傳回值
無。

說明
方法；設定選取範圍繞著此 XYZ 位置來轉譯或旋轉。這個方法只能用於影片片段。

範例
下列範例會指定 3D 轉譯的 XYZ 座標軸：

```
fl.getDocumentDOM().translate3DCenter({x:180, y:18,z:-30});
```

document.translate3DSelection()

適用版本
Flash CS4 Professional。

用法

```
document.translate3DSelection(xyzCoordinate, bGlobalTransform)
```

參數
xyzCoordinate XYZ 座標，指定 3D 轉譯的座標軸。
bGlobalTransform Boolean 值，指定變形模式應該是整體 (true) 或各自 (false) 的型態。

傳回值
無。

說明
方法：將 3D 轉譯套用至選取範圍。這個方法只能用於影片片段。

範例
在下列範例中，選取範圍先相對於舞台 (整體) 轉譯，然後相對於它本身 (各自) 轉譯。

```
var myDocument = fl.getDocumentDOM();
myDocument.translate3DSelection({x:52.0, y:0, z:0}, true);
myDocument.translate3DSelection({x:52.0, y:0, z:-55.2}, false);
```

請參閱
document.translate3DCenter()

document.transformSelection()

適用版本
Flash MX 2004。

用法
```
document.transformSelection(a, b, c, d)
```

參數
a 浮點數：指定變形矩陣的 (0,0) 元素。
b 浮點數：指定變形矩陣的 (0,1) 元素。
c 浮點數：指定變形矩陣的 (1,0) 元素。
d 浮點數：指定變形矩陣的 (1,1) 元素。

傳回值
無。

說明
方法：套用引數中指定的矩陣，在目前的選取範圍中執行一般變形。如需詳細資訊，請參閱 element.matrix 屬性。

範例
下列範例將選取範圍的 x 方向延伸 2 倍：

```
fl.getDocumentDOM().transformSelection(2.0, 0.0, 0.0, 1.0);
```

document.unGroup()

適用版本
Flash MX 2004。

用法
```
document.unGroup()
```

參數
無。

傳回值
無。

說明
方法；解散目前選取範圍的群組。

範例
下列範例解散目前選取範圍元素的群組：

```
fl.getDocumentDOM().unGroup();
```

請參閱
document.group()

document.union()

適用版本
Flash 8。

用法
```
document.union()
```

參數
無。

傳回值
Boolean 值；如果成功的話，會傳回 true；否則便傳回 false。

說明
方法；結合所有的選取形狀到繪圖物件。

範例
下列範例結合所有的選取形狀到繪圖物件：

```
fl.getDocumentDOM().union();
```

請參閱
document.crop()、document.deleteEnvelope()、document.intersect()、document.punch()、shape.isDrawingObject

document.unlockAllElements()

適用版本
Flash MX 2004。

用法
```
document.unlockAllElements()
```

參數
無。

傳回值
無。

說明
方法：解除目前選取影格上所有鎖定元素的鎖定。

範例
下列範例會解除目前選取影格上所有鎖定元素的鎖定：

```
fl.getDocumentDOM().unlockAllElements();
```

請參閱
element.locked

document.viewMatrix

適用版本
Flash MX 2004。

用法
```
document.viewMatrix
```

說明
唯讀屬性：Matrix 物件。文件為編輯模式時，使用 viewMatrix 從物件空間變形為文件空間。工具接收滑鼠位置時，滑鼠位置是相對於目前編輯的物件。請參閱 Matrix 物件。

例如建立元件時，您按兩下元件進行編輯，並使用 PolyStar 工具繪製，點 (0,0) 為元件的註冊點。不過，drawingLayer 物件會預期接受文件空間中的值，因此，如果使用 drawingLayer 從 (0,0) 繪製線段，將會從「舞台」的左上角開始繪製。viewMatrix 屬性提供從編輯物件空間變形為文件空間的方式。

範例
下列範例取得 viewMatrix 屬性值：

```
var mat = fl.getDocumentDOM().viewMatrix;
```

document.width

適用版本
Flash MX 2004。

用法
```
document.width
```

說明
屬性；整數；指定文件 (「舞台」) 的寬度像素。

範例
下列範例設定 「舞台」寬度為 400 像素。

```
fl.getDocumentDOM().width= 400;
```

請參閱
document.height

document.xmlPanel()

適用版本
Flash MX 2004。

用法
```
document.xmlPanel(fileURI)
```

參數
fileURI 字串；指定定義面板控制項的 XML 檔路徑，表示為 file:/// URI。需要完整路徑。

傳回值
定義有 XML 檔所有定義控制項屬性的物件。所有的屬性都以字串傳回。傳回物件都有預先定義的屬性，名稱為 "dismiss"，字串值為 "accept" 或 "cancel"。

說明
方法；公佈 XMLUI 對話方塊。請參閱 fl.xmlui。

範例
下列範例載入 Test.xml 檔，並顯示檔案中的每個屬性：

```
var obj = fl.getDocumentDOM().xmlPanel(fl.configURI + "Commands/Test.xml");
for (var prop in obj) {
    fl.trace("property " + prop + " = " + obj[prop]);
}
```

document.zoomFactor

適用版本

Flash 8。

用法

```
document.zoomFactor
```

說明

屬性；指定編寫期間的 「舞台」縮放比例。值為 1 等於 100% 縮放；8 等於 800% 縮放；.5 等於 50% 縮放，以此類推。

範例

下列範例會將 「舞台」的縮放係數設定為 200%。

```
fl.getDocumentDOM().zoomFactor = 2;
```

第 12 章 drawingLayer 物件

適用版本
Flash MX 2004。

說明
drawingLayer 物件可當做 Flash 物件的子物件，從 JavaScript 存取。如果使用者想要在拖曳時 (例如，建立選取圈選範圍時) 暫時繪製，則 drawingLayer 物件可做為可擴充工具。您必須先呼叫 drawingLayer.beginFrame()，才能再呼叫任何其它 drawingLayer 方法。

方法摘要
drawingLayer 物件可使用的方法如下：

方法	說明
drawingLayer.beginDraw()	將 Flash 設為繪圖模式。
drawingLayer.beginFrame()	擦除先前使用 drawingLayer 的繪圖，準備執行其它繪圖命令。
drawingLayer.cubicCurveTo()	使用參數做為三次方線段的座標，從目前鋼筆位置開始繪製三次方曲線。
drawingLayer.curveTo()	從目前繪製位置開始到指定點結束，繪製二次方曲線線段。
drawingLayer.drawPath()	繪製指定路徑。
drawingLayer.endDraw()	結束繪圖模式。
drawingLayer.endFrame()	指出繪圖命令群組結束。
drawingLayer.lineTo()	從目前繪製位置至點 (x,y) 繪製一條線段。
drawingLayer.moveTo()	設定目前繪圖位置。
drawingLayer.newPath()	傳回新的 Path 物件。
drawingLayer.setColor()	設定後續繪圖資料的顏色。
drawingLayer.setFill()	這個方法無法使用。
drawingLayer.setStroke()	這個方法無法使用。

drawingLayer.beginDraw()

適用版本
Flash MX 2004。

用法
```
drawingLayer.beginDraw([persistentDraw])
```

參數
persistentDraw　Boolean 值 (選擇性)；如果設定為 true，表示最後影格中的繪圖保留在「舞台」上，直到呼叫新的 beginDraw() 或 beginFrame() (上述內容中，「影格」表示繪圖的開始和結束位置，不是時間軸影格)。例如，使用者繪製矩形時，可以在拖曳滑鼠時預覽形狀的外框。如果想在使用者放開滑鼠按鍵後保留預覽形狀，則將 persistentDraw 設定為 true。

傳回值
無。

說明
方法：將 Flash 切換到繪圖模式。繪圖模式用於按下滑鼠按鍵又想暫時繪圖時。一般僅在建立可擴充工具時才使用此方法。

範例
下列範例將 Flash 切換到繪圖模式：

```
fl.drawingLayer.beginDraw();
```

drawingLayer.beginFrame()

適用版本
Flash MX 2004。

用法
```
drawingLayer.beginFrame()
```

參數
無。

傳回值
無。

說明
方法：擦除先前使用 drawingLayer 的繪圖，準備執行其它繪圖命令。必須在 drawingLayer.beginDraw() 之後呼叫。將 drawingLayer.beginFrame() 和 drawingLayer.endFrame() 之間所繪製的一切保留在「舞台」上，直到呼叫下一個 beginFrame() 和 endFrame() (上述內容中，「影格」表示繪圖的開始和結束位置，不是時間軸影格)。一般僅在建立可擴充工具時才使用此方法。請參閱 drawingLayer.beginDraw()。

drawingLayer.cubicCurveTo()

適用版本
Flash MX 2004。

用法
```
drawingLayer.cubicCurveTo(x1Ctrl, y1Ctrl, x2Ctl, y2Ctl, xEnd, yEnd)
```

參數
x1Ctl 浮點數值；為第一個控制點的 x 位置。
y1Ctl 浮點數值；為第一個控制點的 y 位置。
x2Ctl 浮點數值；為中間控制點的 x 位置。
y2Ctl 浮點數值；為中間控制點的 y 位置。
xEnd 浮點數值；為結尾控制點的 x 位置。

yEnd 浮點數值；為結尾控制點的 y 位置。

傳回值
無。

說明
方法；使用參數做為三次方線段的座標，從目前鋼筆位置開始繪製三次方曲線。一般僅在建立可擴充工具時才使用此方法。

範例
下列範例使用指定控制點繪製三次方曲線：

```
fl.drawingLayer.cubicCurveTo(0, 0, 1, 1, 2, 0);
```

drawingLayer.curveTo()

適用版本
Flash MX 2004。

用法
```
drawingLayer.curveTo(xCtl, yCtl, xEnd, yEnd)
```

參數
xCtl 浮點數值；為控制點的 x 位置。

yCtl 浮點數值；為控制點的 y 位置。

xEnd 浮點數值；為結尾控制點的 x 位置。

yEnd 浮點數值；為結尾控制點的 y 位置。

傳回值
無。

說明
方法；從目前繪製位置開始到指定點結束，繪製二次方曲線線段。一般僅在建立可擴充工具時才使用此方法。

範例
下列範例使用指定控制點繪製二次方曲線：

```
fl.drawingLayer.curveTo(0, 0, 2, 0);
```

drawingLayer.drawPath()

適用版本
Flash MX 2004。

用法
```
drawingLayer.drawPath(path)
```

參數
path 要繪製的 Path 物件。

傳回值
無。

說明
方法；繪製 path 參數指定的路徑。一般僅在建立可擴充工具時才使用此方法。

範例
下列範例繪製 Path 物件 (名稱為 gamePath) 指定的路徑：

```
fl.drawingLayer.drawPath(gamePath);
```

drawingLayer.endDraw()

適用版本
Flash MX 2004。

用法
```
drawingLayer.endDraw()
```

參數
無。

傳回值
無。

說明
方法；結束繪圖模式。繪圖模式用於按下滑鼠按鍵又想暫時繪圖時。一般僅在建立可擴充工具時才使用此方法。

範例
下列範例結束繪圖模式：

```
fl.drawingLayer.endDraw();
```

drawingLayer.endFrame()

適用版本
Flash MX 2004。

用法
```
drawingLayer.endFrame()
```

參數
無。

傳回值
無。

說明
方法；指出繪圖命令群組結束。繪圖命令群組是指 drawingLayer.beginFrame() 和 drawingLayer.endFrame() 之間繪製的一切。下次呼叫 drawingLayer.beginFrame() 時，將擦除此繪圖命令群組中繪製的一切。一般僅在建立可擴充工具時才使用此方法。

drawingLayer.lineTo()

適用版本
Flash MX 2004。

用法
```
drawingLayer.lineTo(x, y)
```

參數
x 浮點數值；為繪製線段端點的 x 座標。

y 浮點數值；為繪製線段端點的 y 座標。

傳回值
無。

說明
方法；從目前繪製位置至點 (x,y) 繪製一條線段。一般僅在建立可擴充工具時才使用此方法。

範例
下列範例從目前繪製位置至點 (20,30) 繪製一條線段：

```
fl.drawingLayer.lineTo(20, 30);
```

drawingLayer.moveTo()

適用版本
Flash MX 2004。

用法
```
drawingLayer.moveTo(x, y)
```

參數
x 浮點數值；指定開始繪製的 x 座標位置。

y 浮點數值；指定開始繪製的 y 座標位置。

傳回值
無。

說明
方法；設定目前繪圖位置。一般僅在建立可擴充工具時才使用此方法。

範例
下列範例將目前繪圖位置設定於點 (10,15)：

```
fl.drawingLayer.moveTo(10, 15);
```

drawingLayer.newPath()

適用版本
Flash MX 2004。

用法

```
drawingLayer.newPath()
```

參數
無。

傳回值
Path 物件。

說明
方法；傳回新的 Path 物件。一般僅在建立可擴充工具時才使用此方法。請參閱 Path 物件。

範例
下列範例傳回新的 Path 物件：

```
fl.drawingLayer.newPath();
```

drawingLayer.setColor()

適用版本
Flash MX 2004。

用法

```
drawingLayer.setColor(color)
```

參數
color　後續繪圖資料的顏色，格式為下列其中一種：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

傳回值
無。

說明
方法；設定後續繪圖資料的顏色。僅適用於永續性資料。若要使用此方法，必須將傳遞到 drawingLayer.beginDraw() 的參數設定為 true。一般僅在建立可擴充工具時才使用此方法。請參閱 drawingLayer.beginDraw()。

範例
下列範例在「舞台」上繪製一條紅色線段：

```
fl.drawingLayer.beginDraw( true );
fl.drawingLayer.beginFrame();
fl.drawingLayer.setColor( "#ff0000" );
fl.drawingLayer.moveTo(0,0);
fl.drawingLayer.lineTo(100,100);
fl.drawingLayer.endFrame();
fl.drawingLayer.endDraw();
```

drawingLayer.setFill()

這個方法無法使用。

drawingLayer.setStroke()

這個方法無法使用。

第 13 章 Edge 物件

適用版本
Flash MX 2004。

說明
Edge 物件代表「舞台」上形狀的邊緣。

方法摘要
Edge 物件可以使用下列方法：

方法	說明
edge.getControl()	取得設定至邊緣指定控制點位置的點物件。
edge.getHalfEdge()	傳回 HalfEdge 物件。
edge.setControl()	設定邊緣控制點的位置。
edge.splitEdge()	將邊緣斷開成兩個部分。

屬性摘要
Edge 物件可以使用下列屬性：

屬性	說明
edge.cubicSegmentIndex	整數，指定邊緣三次方線段的索引值。
edge.id	唯讀；整數，代表邊緣的唯一識別名稱。
edge.isLine	唯讀；整數，含有值 0 或 1。
edge.stroke	Stroke 物件。

edge.cubicSegmentIndex

適用版本
Flash CS4 Professional。

用法
```
edge.cubicSegmentIndex
```

說明
唯讀屬性；整數，指定邊緣三次方線段的索引值 (請參閱 shape.getCubicSegmentPoints())。

範例
下列程式碼會顯示指定邊緣中所有三次方線段的索引值：

```
var theShape = fl.getDocumentDOM().selection[0];
var edgesArray = theShape.edges;
for(var i=0;i<edgesArray.length; i++) {
    fl.trace(edgesArray[i].cubicSegmentIndex);
}
```

edge.getControl()

適用版本
Flash MX 2004。

用法
```
edge.getControl(i)
```

參數
i 整數；指定傳回的邊緣控制點。指定 0 為第一個控制點、1 為中間控制點，或 2 為結尾控制點。如果 edge.isLine 屬性為 true，則會設定中間控制點為連接開始和結束控制點的線段中間點。

傳回值
指定的控制點。

說明
方法；取得設定至邊緣指定控制點位置的點物件。

範例
下列範例將指定形狀的第一個控制點儲存於 pt 變數：

```
var shape = fl.getDocumentDOM().selection[0];
var pt = shape.edges[0].getControl(0);
```

edge.getHalfEdge()

適用版本
Flash MX 2004。

用法
```
edge.getHalfEdge(index)
```

參數
index 整數；指定傳回的不完整邊緣。index 的值必須是 0 (代表第一個不完整邊緣)，或是 1 (代表第二個不完整邊緣)。

傳回值
HalfEdge 物件。

說明
方法；傳回 HalfEdge 物件。

範例
下列範例會將指定邊緣的不完整邊緣儲存於 hEdge0 和 hEdge1 變數：

```
var shape = fl.getDocumentDOM().selection[0];
var edge = shape.edges[0];
var hEdge0 = edge.getHalfEdge(0);
var hEdge1 = edge.getHalfEdge(1);
```

edge.id

適用版本
Flash MX 2004。

用法
```
edge.id
```

說明
唯讀屬性；整數；代表邊緣的唯一識別名稱。

範例
下列範例會將指定邊緣的唯一識別名稱儲存於 my_shape_id 變數：

```
var shape = fl.getDocumentDOM().selection[0];
var my_shape_id = shape.edges[0].id;
```

edge.isLine

適用版本
Flash MX 2004。

用法
```
edge.isLine
```

說明
唯讀屬性；整數；含有值 0 或 1。值 1 表示邊緣為直線。在此情況之下，中間控制點將連接兩個端點的線段一分為二。

範例
下列範例會判斷指定的邊緣是否為直線，並在「輸出」面板中顯示值 1 (為直線) 或 0 (不為直線)：

```
var shape = fl.getDocumentDOM().selection[0];
fl.trace(shape.edges[0].isLine);
```

edge.setControl()

適用版本
Flash MX 2004。

用法

```
edge.setControl(index, x, y)
```

參數

index 整數，指定要設定的控制點。使用值 0、1 或 2 來分別指定開始、中間和結尾控制點。

x 浮點數值；指定控制點的水平位置。若「舞台」為編輯中或使用原地編輯模式，點座標會與編輯物件相對應。否則，點座標會與「舞台」相對應。

y 浮點數值；指定控制點的垂直位置。若「舞台」為編輯中或使用原地編輯模式，點座標會與編輯物件相對應。否則，點座標會與「舞台」相對應。

傳回值

無。

說明

方法；設定邊緣控制點的位置。使用這個方法之前，您必須呼叫 shape.beginEdit()。請參閱 shape.beginEdit()。

範例

下列範例將指定邊緣的開始控制點設定為 (0, 1) 座標：

```
x = 0; y = 1;
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.edges[0].setControl(0, x, y);
shape.endEdit();
```

edge.splitEdge()

適用版本

Flash MX 2004。

用法

```
edge.splitEdge(t)
```

參數

t 浮點數值；介於 0 和 1 之間，用於指定斷開邊緣的位置。值 0 表示一個端點，而值 1 則表示另一個端點。例如，傳遞值為 0.5 時，會在邊緣中間斷開，就是線段的中間點。如果邊緣為曲線，則 0.5 代表曲線的參數中點。

傳回值

無。

說明

方法；將邊緣斷開成兩個部分。使用這個方法之前，您必須呼叫 shape.beginEdit()。

範例

下列範例將指定邊緣斷開為二：

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit()
shape.edges[0].splitEdge( 0.5 );
shape.endEdit()
```

edge.stroke

適用版本

Flash CS4 Professional。

用法

```
edge.stroke
```

說明

屬性；Stroke 物件。

範例

下列範例會顯示所選物件第一個邊緣的筆畫顏色：

```
var shape = fl.getDocumentDOM().selection[0];
fl.trace(shape.edges[0].stroke.color);
```

第 14 章 Element 物件

適用版本

Flash MX 2004。

說明

「舞台」上的所有物件類型皆為 Element。下列程式碼範例可讓您選取元素：

```
var el = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
```

方法摘要

Element 物件可使用的方法如下：

方法	說明
element.getPersistentData()	擷取 *name* 參數指定的資料值。
element.getTransformationPoint()	取得指定元素之變形點的值
element.hasPersistentData()	決定指定的資料是否已附加到指定的元素。
element.removePersistentData()	移除任何指定名稱附加到物件的永續性資料。
element.setPersistentData()	儲存含有元素的資料。
element.setTransformationPoint()	設定元素之變形點的位置

屬性摘要

Element 物件可使用的屬性如下：

屬性	說明
element.depth	唯讀；整數；檢視中的物件深度，值大於 0。
element.elementType	唯讀；字串；代表指定元素的類型。
element.height	浮點值；指定元素高度，以像素為單位。
element.layer	唯讀；代表元素所在位置的 Layer 物件。
element.left	唯讀；浮點值，代表元素的左側。
element.locked	Boolean 值；如果已鎖定元素，則為 true；否則為 false。
element.matrix	Matrix 物件。matrix 具有屬性 a、b、c、d、tx 和 ty。a、b、c 和 d 為浮點值，而 tx 和 ty 則為座標。
element.name	字串；指定元素名稱，通常是指實體名稱。
element.rotation	介於 -180 和 180 之間的整數或浮點值，指定物件的順時針旋轉，以度數為單位。
element.scaleX	浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 x 縮放值。
element.scaleY	浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 y 縮放值。
element.selected	Boolean 值，指定元素是否已被選取。
element.skewX	介於 -180 和 180 之間的浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 x 傾斜值。

屬性	說明
element.skewY	介於 -180 和 180 之間的浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 y 傾斜值。
element.top	唯讀；元素上方。
element.transformX	浮點數；指定所選元素變形點的 x 值，且其在元素父輩的座標系統內。
element.transformY	浮點數；指定所選元素變形點的 y 值，且其在元素父輩的座標系統內。
element.width	浮點值；指定元素寬度，以像素為單位。
element.x	浮點值，指定選取元素之註冊點的 x 值。
element.y	浮點值，指定選取元素之註冊點的 y 值。

element.depth

適用版本
Flash MX 2004。

用法
```
element.depth
```

說明
唯讀屬性；整數；檢視中的物件深度，值大於 0。「舞台」上的物件繪圖順序會指定物件的排列順序。您也可以使用 「修改 > 排列」選單項目管理物件順序。

範例
下列範例在 「輸出」面板中顯示指定元素的深度。

```
// Select an object and run this script.
fl.trace("Depth of selected object: " + fl.getDocumentDOM().selection[0].depth);
```

請參閱 element.elementType 的範例。

element.elementType

適用版本
Flash MX 2004。

用法
```
element.elementType
```

說明
唯讀屬性；字串；代表指定元素的類型。此值為下面其中之一："shape"、"text"、"instance" 或 "shapeObj"。"shapeObj" 是使用可擴充工具建立的。

範例
下列範例會將第一個元素的類型儲存於 eType 變數中：

```
// In a new file, place a movie clip on first frame top layer, and
// then run this line of script.
var eType = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].elementType; // eType =
instance
```

下列範例會顯示在目前圖層或影格中所有元素的幾項屬性：

```
var tl = fl.getDocumentDOM().getTimeline()
var elts = tl.layers[tl.currentLayer].frames[tl.currentFrame].elements;
for (var x = 0; x < elts.length; x++) {
    var elt = elts[x];
    fl.trace("Element "+ x +" Name = " + elt.name + " Type = " + elt.elementType + " location = " + elt.left
+ "," + elt.top + " Depth = " + elt.depth);
}
```

element.getPersistentData()

適用版本

Flash MX 2004。

用法

```
element.getPersistentData(name)
```

參數

name　字串，識別要傳回的資料。

傳回值

name 參數指定的資料；如果資料不存在，則為 0。

說明

方法；擷取 name 參數指定的資料值。資料的類型取決於儲存的資料類型 (請參閱 element.setPersistentData())。只有元件和點陣圖支援永續性資料。

範例

下列範例設定和取得指定元素的資料，在「輸出」面板上顯示值，然後移除資料：

```
// At least one symbol or bitmap is selected in the first layer, first frame.
var elt = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
elt.setPersistentData("myData","integer", 12);
if (elt.hasPersistentData("myData")){
fl.trace("myData = "+ elt.getPersistentData("myData"));
elt.removePersistentData( "myData" );
fl.trace("myData = "+ elt.getPersistentData("myData"));
}
```

element.getTransformationPoint()

適用版本

Flash CS3 Professional。

用法
```
element.getTransformationPoint()
```

參數
無。

傳回值
點，指定在元素座標系統中變形點 (以及 「原點」或 「零點」) 的位置 (例如 {x:10, y:20}，其中 x 和 y 是浮點數字)。

說明
方法：取得指定元素之變形點的值。

根據選取項目的類型而定，變形點會相對於不同的位置。如需詳細資訊，請參閱 element.setTransformationPoint()。

範例
下列範例會取得文件中第一個圖層上第九個影格中之第三個元素的變形點。transPoint.x 屬性會提供變形點的 x 座標。transPoint.y 屬性會提供變形點的 y 座標。

```
var transPoint =
fl.getDocumentDOM().getTimeline().layers[0].frames[8].elements[2].getTransformationPoint();
```

請參閱
document.getTransformationPoint()、element.setTransformationPoint()、element.transformX、element.transformY

element.hasPersistentData()

適用版本
Flash MX 2004。

用法
```
element.hasPersistentData(name)
```

參數
name 字串，指定要測試之資料項目的名稱。

傳回值
Boolean 值：如果指示資料已附加到物件，則為 true；否則為 false。

說明
方法：決定指定的資料是否已附加到指定的元素。只有元件和點陣圖支援永續性資料。

範例
請參閱 element.getPersistentData()。

element.height

適用版本
Flash MX 2004。

用法
```
element.height
```

說明
屬性；浮點值；指定元素高度，以像素為單位。

請不要使用此屬性來調整文字欄位大小。請選取文字欄位並使用 document.setTextRectangle()。使用這項屬性搭配文字欄位就會縮放文字的大小。

範例
下列範例將指定元素高度設定為 100：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].height = 100;
```

element.layer

適用版本
Flash 8。

用法
```
element.layer
```

說明
唯讀屬性；代表元素所在位置的 Layer 物件。

範例
下列範例會將包含元素的 Layer 物件儲存於 theLayer 變數中：

```
var theLayer = element.layer;
```

element.left

適用版本
Flash MX 2004。

用法
```
element.left
```

說明
唯讀屬性；浮點值，代表元素的左側。對場景中的元素而言，element.left 的值是相對於 「舞台」的左上方；如果元素儲存在元件內，則相對於元件的註冊點 (也稱為 「原點」或 「零點」)。請使用 document.setSelectionBounds() 或 document.moveSelectionBy() 來設定這個屬性。

範例

下列範例將說明元素移動時此屬性值如何變更：

```
// Select an element on the Stage and then run this script.
var sel = fl.getDocumentDOM().selection[0];
fl.trace("Left (before) = " + sel.left);
fl.getDocumentDOM().moveSelectionBy({x:100, y:0});
fl.trace("Left (after) = " + sel.left);
```

請參閱 element.elementType 的範例。

element.locked

適用版本

Flash MX 2004。

用法

```
element.locked
```

說明

屬性；Boolean 值：如果已鎖定元素，則為 true，否則為 false。如果 element.elementType 的值為 "shape"，則會忽略此屬性。

範例

下列範例將第一個元素鎖定在最上層圖層的第一個影格：

```
// Similar to Modify > Arrange > Lock:
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].locked = true;
```

element.matrix

適用版本

Flash MX 2004。

用法

```
element.matrix
```

說明

屬性；Matrix 物件。矩陣具備屬性 a、b、c、d、tx 和 ty。a、b、c 和 d 屬性為浮點值；tx 和 ty 屬性為座標。請參閱 Matrix 物件。

範例

下列範例會將指定元素在 x 移動 10 像素，在 y 移動 20 像素：

```
var mat = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].matrix;
mat.tx += 10;
mat.ty += 20;
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].matrix = mat;
```

element.name

適用版本
Flash MX 2004。

用法
```
element.name
```

說明
屬性；字串；指定元素名稱，通常是指實體名稱。如果 element.elementType 的值為 "shape"，則會忽略此屬性。請參閱 element.elementType。

範例
下列範例會將最上層圖層影格 1 中的第一個元素實體名稱設定為 "clip_mc"：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].name = "clip_mc";
```

請參閱 element.elementType 的範例。

element.removePersistentData()

適用版本
Flash MX 2004。

用法
```
element.removePersistentData(name)
```

參數
name 字串，指定要移除的資料名稱。

傳回值
無。

說明
方法；移除任何指定名稱附加到物件的永續性資料。只有元件和點陣圖支援永續性資料。

範例
請參閱 element.getPersistentData()。

element.rotation

適用版本
Flash CS3 Professional。

用法
```
element.rotation
```

說明
屬性：介於 -180 和 180 之間的整數或浮點值，指定物件的順時針旋轉，以度數為單位。

範例
下列範例會將目前選取元素的旋轉設定為 45 度：

```
var element = fl.getDocumentDOM().selection[0];
fl.trace("Element rotation = " + element.rotation);
element.rotation = 45;
fl.trace("After setting rotation to 45: rotation = " + element.rotation);
```

element.scaleX

適用版本
Flash CS3 Professional。

用法
```
element.scaleX
```

說明
屬性：浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 x 縮放值。值 1 是表示 100% 縮放。

範例
下列範例會將目前選取範圍的 x 縮放值設定為 2 (將其值加倍)：

```
var element = fl.getDocumentDOM().selection[0];
element.scaleX = 2;
```

請參閱
element.scaleY

element.scaleY

適用版本
Flash CS3 Professional。

用法
```
element.scaleY
```

說明
屬性：浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 y 縮放值。值 1 是表示 100% 縮放。

範例
下列範例會將目前選取範圍的 y 縮放值設定為 2 (將其值加倍)：

```
var element = fl.getDocumentDOM().selection[0];
element.scaleY = 2;
```

請參閱
element.scaleX

element.selected

適用版本
Flash 8。

用法
```
element.selected
```

說明
屬性；Boolean 值，指定已選取 (true) 或未選取 (false) 元素。

範例
下列範例會選取元素：

```
element.selected = true;
```

element.setPersistentData()

適用版本
Flash MX 2004。

用法
```
element.setPersistentData(name, type, value)
```

參數
name 字串，指定要與資料產生關聯的名稱。此名稱是用來擷取資料。

type 字串，定義資料的類型。允許的值包括 "integer"、"integerArray"、"double"、"doubleArray"、"string" 和 "byteArray"。

value 指定要與物件產生關聯的值。value 的資料類型取決於 type 參數的值。指定的值應適合 type 參數所指定的資料類型。

傳回值
無。

說明
方法；儲存含有元素的資料。在重新開啟包含元素的 FLA 檔時，即可取得資料。只有元件和點陣圖支援永續性資料。

範例
請參閱 element.getPersistentData()。

element.setTransformationPoint()

適用版本
Flash CS3 Professional。

用法
```
element.setTransformationPoint(transformationPoint)
```

參數
transformationPoint 點，指定元素或群組之變形點的值 (例如，{x:10, y:20}，其中 x 和 y 是浮點數)。

- 形狀：transformationPoint 的設定是相對於文件 (0,0 是在 「舞台」的左上角)。
- 元件：transformationPoint 的設定是相對於元件的註冊點 (0,0 位於註冊點)。
- 文字：transformationPoint 的設定是相對於文字欄位 (0,0 為文字欄位的左上角)。
- 點陣圖 / 視訊：transformationPoint 的設定是相對於點陣圖 / 視訊 (0,0 是在點陣圖或視訊的左上角)。
- 繪製物件、基本物件和群組：transformationPoint 的設定是相對於元素或群組的中心 (0,0 是在元素或群組的中心點)。

傳回值
無。

說明
方法：設定元素之變形點的位置。

這個方法幾乎與 document.setTransformationPoint() 完全相同。差異之處如下：

- 繪圖物件、基本物件和群組的變形點會設定為相對於元素或群組的中心，而非相對於 「舞台」。
- 您可以設定元素的變形點，而不用先選取這些元素。

  這個方法會移動變形點，但不會移動元素。反之，element.transformX 和 element.transformY 屬性則會移動元素。

範例
下列範例會設定 「舞台」上第三個元素的變形點為 100, 200：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[2].setTransformationPoint({x:100, y:200});
```

請參閱
document.setTransformationPoint(). element.getTransformationPoint()、element.transformX、element.transformY

element.skewX

適用版本
Flash CS3 Professional。

用法
```
element.skewX
```

說明
屬性：介於 -180 和 180 之間的浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 x 傾斜值。

範例
下列範例會將目前選取範圍的 x 傾斜值設定為 10：

```
var element = fl.getDocumentDOM().selection[0];
element.skewX = 10;
```

請參閱
document.setTransformationPoint()、element.skewY

element.skewY

適用版本
Flash CS3 Professional。

用法
```
element.skewY
```

說明
屬性；介於 -180 和 180 之間的浮點值，指定元件、繪圖物件和基本矩形與橢圓形的 y 傾斜值。

範例
下列範例會將目前選取範圍的 y 傾斜值設定為 10：

```
var element = fl.getDocumentDOM().selection[0];
element.skewY = 10;
```

請參閱
document.setTransformationPoint()、element.skewX

element.top

適用版本
Flash MX 2004。

用法
```
element.top
```

說明
唯讀屬性；元素上方。對場景中的元素而言，element.top 的值是相對於「舞台」的左上方；如果元素儲存在元件內，則相對於元件的註冊點。請使用 document.setSelectionBounds() 或 document.moveSelectionBy() 來設定這個屬性。

範例
下列範例會顯示元素移動時此屬性值如何變更：

```
// Select an element on the Stage and then run this script.
var sel = fl.getDocumentDOM().selection[0];
fl.trace("Top (before) = " + sel.top);
fl.getDocumentDOM().moveSelectionBy({x:0, y:100});
fl.trace("Top (after) = " + sel.top);
```

請參閱 element.elementType 的範例。

element.transformX

適用版本
Flash CS3 Professional。

用法
```
element.transformX
```

說明
屬性；浮點數；指定所選元素變形點的 x 值，且其在元素父輩的座標系統內。將這個屬性設定為新值會移動元素。反之，element.setTransformationPoint() 方法會移動變形點，但不會移動元素。

範例

請參閱
element.getTransformationPoint()、element.setTransformationPoint()、element.transformY

element.transformY

適用版本
Flash CS3 Professional。

用法
```
element.transformY
```

說明
屬性；浮點數；指定所選元素變形點的 y 值，且其在元素父輩的座標系統內。將這個屬性設定為新值會移動元素。反之，element.setTransformationPoint() 方法會移動變形點，但不會移動元素。

請參閱
element.getTransformationPoint()、element.setTransformationPoint()、element.transformX

element.width

適用版本
Flash MX 2004。

用法
```
element.width
```

說明
屬性；浮點值；指定元素寬度，以像素為單位。

請不要使用此屬性來調整文字欄位大小。請選取文字欄位並使用 document.setTextRectangle()。使用這項屬性搭配文字欄位就會縮放文字的大小。

範例
下列範例將指定元素寬度設定為 100：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].width= 100;
```

element.x

適用版本
Flash CS3 Professional。

用法
```
element.x
```

說明
屬性；浮點值，指定選取元素之註冊點的 x 值。

範例
下列範例會將指定元素之註冊點的值設定為 100, 200：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].x= 100;
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].y= 200;
```

請參閱
element.y

element.y

適用版本
Flash CS3 Professional。

用法
```
element.y
```

說明
屬性；浮點值，指定選取元素之註冊點的 y 值。

範例
請參閱 element.x。

第 15 章 Fill 物件

適用版本
Flash MX 2004。

說明
物件包含 「工具」面板或選取形狀 「填色」顏色設定的所有屬性。若要擷取 Fill 物件，請使用 document.getCustomFill()。

屬性摘要
Fill 物件可以使用下列屬性：

屬性	說明
fill.bitmapIsClipped	Boolean 值，針對大於點陣圖的形狀，指定要裁剪或重複其點陣圖填色。
fill.bitmapPath	字串，指定元件庫中點陣圖填色的路徑和名稱。
fill.color	字串，代表填色顏色的十六進位值或整數。
fill.colorArray	漸層中的顏色陣列。
fill.focalPoint	整數，指定距變形點的漸層焦點水平偏移值。
fill.linearRGB	Boolean 值，指定是否要將填色呈現為線性或放射狀 RGB 漸層。
fill.matrix	Matrix 物件，定義漸層填色的位置、方向和縮放。
fill.overflow	字串，指定漸層的溢出行為。
fill.posArray	整數陣列，每個整數都在 0 到 255 的範圍之內，表示相應顏色的位置。
fill.style	字串，指定填色樣式。

fill.bitmapIsClipped

適用版本
Flash CS4 Professional。

用法
```
fill.bitmapIsClipped
```

說明
屬性：Boolean 值，針對大於點陣圖的形狀，指定要裁剪 (true) 或重複 (false) 其點陣圖填色。只有當 fill.style 屬性值為 "bitmap" 時才能使用此屬性。如果該形狀小於點陣圖，則此值為 false。

範例
下列範例會適時將點陣圖填色是否已裁切的資訊顯示在 「輸出」面板中：

```
var fill = fl.getDocumentDOM().getCustomFill();
if (fill.style == "bitmap")
    fl.trace("Fill image is clipped: " + fill.bitmapIsClipped);
```

請參閱
fill.bitmapPath

fill.bitmapPath

適用版本
Flash CS4 Professional。

用法
```
fill.bitmapPath
```

說明
屬性；字串，指定元件庫中點陣圖填色的路徑和名稱。只有當 fill.style 屬性值為 "bitmap" 時才能使用此屬性。

範例
下列範例會將指定項目的填色樣式設定為元件庫的點陣圖影像：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "bitmap";
fill.bitmapPath = "myBitmap.jpg";
fl.getDocumentDOM().setCustomFill(fill);
```

請參閱
fill.bitmapIsClipped

fill.color

適用版本
Flash MX 2004。

用法
```
fill.color
```

說明
屬性；填色的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

範例
下列範例設定目前選取範圍的填色顏色：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.color = "#FFFFFF";
fl.getDocumentDOM().setCustomFill( fill );
```

fill.colorArray

適用版本
Flash MX 2004。

用法
```
fill.colorArray
```

說明
屬性；漸層中的顏色陣列，以整數表示。只有當 fill.style 屬性值為 radialGradient 或 "linearGradient" 時，才可以使用此屬性。
請參閱 fill.style

範例
下列範例會適時地在「輸出」面板中顯示目前選取範圍的顏色陣列。

```
var fill = fl.getDocumentDOM().getCustomFill();
if(fill.style == "linearGradient" || fill.style == "radialGradient")
    alert(fill.colorArray);
```

下列範例會將填色設定為指定的線性漸層：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "linearGradient";
fill.colorArray = ["#00ff00","#ff00ff"];
fill.posArray = [0, 255];
fl.getDocumentDOM().setCustomFill(fill);
```

fill.focalPoint

適用版本
Flash 8。

用法
```
fill.focalPoint
```

說明
屬性；整數，指定從變形點算起的漸層焦點水平偏移值。例如值為 10 時，焦點的位置應該是變形點至漸層邊緣距離的 10/255。值為 255 時，焦點的位置應該在漸層的左邊界。預設值為 0。

只有當 fill.style 屬性值為 "radialGradient" 時，才可以使用此屬性。

範例
下列範例會將目前選取範圍之放射性漸層的焦點設定為距離形狀中心右方 100 像素的位置：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "radialGradient";
fill.colorArray = ["#00ff00","#ff00ff"];
fill.posArray = [0, 255];
fill.focalPoint = 10100;
fl.getDocumentDOM().setCustomFill(fill);
```

fill.linearRGB

適用版本
Flash 8。

用法
```
fill.linearRGB
```

說明
屬性；Boolean 值，指定是否要將填色呈現為線性或放射狀 RGB 漸層。此屬性設定為 true 可指定線性插入漸層；設定為 false 則可指定放射狀插入漸層。預設值為 false。

範例
下列範例會指定目前選取範圍的漸層應該以線性 RGB 呈現：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.linearRGB style = true"radialGradient";
fill.colorArray = ["#00ff00","#ff00ff"];
fill.posArray = [0, 255];
fill.focalPoint = 100;
fill.linearRGB = true;
fl.getDocumentDOM().setCustomFill(fill);
```

fill.matrix

適用版本
Flash MX 2004。

用法
```
fill.matrix
```

說明
屬性；Matrix 物件，定義漸層填色的位置、方向和縮放。

範例
下列範例會使用 fill.matrix 屬性來指定目前選取範圍的漸層填色：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = 'radialGradient';
fill.colorArray = ['#00ff00','#ff00ff'];
fill.posArray = [0, 255];
fill.focalPoint = 100;
fill.linearRGB = false;
fill.overflow = 'repeat';
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 0.0167083740234375;
mat.b = -0.0096435546875;
mat.c = 0.0312957763671875;
mat.d = 0.05419921875;
mat.tx = 288.65;
mat.ty = 193.05;
for (i in mat) {
    fl.trace(i+' : '+mat[i]);
}
fl.getDocumentDOM().setCustomFill(fill);
```

fill.overflow

適用版本

Flash 8。

用法

```
fill.overflow
```

說明

屬性；字串，指定漸層溢位行為的字串。可接受的值為 "extend"、"repeat" 和 "reflect"；字串不需區分大小寫。預設值為 "extend"。

範例

下列範例會指定目前選取範圍的溢位行為應為 "extend"：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.overflow = "extend";
fl.getDocumentDOM().setCustomFill(fill);
```

fill.posArray

適用版本

Flash MX 2004。

用法

```
fill.posArray
```

說明

屬性；整數陣列，每個整數都在 0 到 255 的範圍之內，表示相應顏色的位置。只有當 fill.style 屬性值為 radialGradient 或 "linearGradient" 時，才可以使用此屬性。

範例

下列範例會指定要在目前選取範圍之線性漸層中使用的顏色：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style = "linearGradient";
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];
fill.posArray= [0,100, 200];
fl.getDocumentDOM().setCustomFill( fill );
```

fill.style

適用版本

Flash MX 2004。Flash CS4 Professional 新增了 "bitmap" 值。

用法

```
fill.style
```

說明

屬性；字串，指定填色樣式。可接受的值為 "bitmap"、"solid"、"linearGradient"、"radialGradient" 和 "noFill"。

如果此值為 "linearGradient" 或 "radialGradient"，也可以使用 fill.colorArray 與 fill.posArray 屬性。如果此值為 "bitmap"，也可以使用 fill.bitmapIsClipped 與 fill.bitmapPath 屬性。

範例

下列範例會指定要在目前選取範圍之線性漸層中使用的顏色：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.style= "linearGradient";
fill.colorArray = [ 0x00ff00, 0xff0000, 0x0000ff ];
fill.posArray= [0,100, 200];
fl.getDocumentDOM().setCustomFill( fill );
```

第 16 章 Filter 物件

適用版本
Flash 8。

說明
此物件包含所有濾鏡的所有屬性。filter.name 屬性會指定濾鏡的類型，並且判斷各個濾鏡可應用的屬性。請參閱 filter.name。

若要傳回一個以上物件的濾鏡清單，請使用 document.getFilters()。若要套用濾鏡到一個以上物件，請使用 document.setFilters()。請參閱 document.getFilters() 與 document.setFilters()。

屬性摘要
Filter 物件可搭配使用以下屬性：

屬性	說明
filter.angle	浮點值，指定陰影或反白標示顏色的角度，並以度數為單位。
filter.blurX	浮點值，指定 x 方向模糊化的量，並以像素為單位。
filter.blurY	浮點值，指定 y 方向模糊化的量。
filter.brightness	浮點值，指定濾鏡的亮度。
filter.color	字串，代表濾鏡顏色的十六進位值或整數。
filter.contrast	浮點值，指定濾鏡的對比值。
filter.distance	浮點值，指定濾鏡特效和物件之間的距離，以像素為單位。
filter.enabled	Boolean 值，指定是否已啟用指定濾鏡。
filter.hideObject	Boolean 值，指定是否隱藏來源影像。
filter.highlightColor	字串，代表反白標示顏色的十六進位值或整數。
filter.hue	浮點值，指定濾鏡的色相。
filter.inner	Boolean 值，指定陰影是否為內陰影。
filter.knockout	Boolean 值，指定濾鏡是否為去底色濾鏡。
filter.name	唯讀；字串，指定濾鏡類型。
filter.quality	字串，指定模糊化品質。
filter.saturation	浮點值，指定濾鏡的飽和度。
filter.shadowColor	字串，代表陰影顏色的十六進位值或整數。
filter.strength	整數，指定濾鏡的強度百分比。
filter.type	字串；指定斜角或光暈的類型。

filter.angle

適用版本
Flash 8。

用法
```
filter.angle
```

說明
屬性；浮點值，指定陰影或反白標示顏色的角度，並以度數為單位。有效值介於 0 和 360 之間。filter.name 屬性值為 "bevelFilter"、"dropShadowFilter"、"gradientBevelFilter" 或 "gradientGlowFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的「斜角」濾鏡角度設定成 120：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++) {
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].angle = 120;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty()

filter.blurX

適用版本
Flash 8。

用法
```
filter.blurX
```

說明
屬性；浮點值，指定 x 方向模糊化的量，以像素為單位。有效值介於 0 和 255 之間。此屬性是針對其 filter.name 屬性值為 "bevelFilter"、"blurFilter"、"dropShadowFilter"、"glowFilter"、"gradientBevelFilter" 或 "gradientGlowFilter" 的 Filter 物件所定義。

範例
下列範例會將選取物件上「模糊」濾鏡的 blurX 值設定成 30，並將 blurY 值設為 20：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'blurFilter'){
        myFilters[i].blurX = 30;
        myFilters[i].blurY = 20;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty(), filter.blurY

filter.blurY

適用版本
Flash 8。

用法
```
filter.blurY
```

說明
屬性；浮點值，指定 y 方向模糊化的量，以像素為單位。有效值介於 0 和 255 之間。此屬性是針對其 filter.name 屬性值為 "bevelFilter"、"blurFilter"、"dropShadowFilter"、"glowFilter"、"gradientBevelFilter" 或 "gradientGlowFilter" 的 Filter 物件所定義。

範例
請參閱 filter.blurX。

請參閱
document.setFilterProperty(), filter.blurX

filter.brightness

適用版本
Flash 8。

用法
```
filter.brightness
```

說明
屬性；浮點值，指定濾鏡的亮度。有效值介於 -100 到 100 之間。filter.name 屬性值為 "adjustColorFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的「調整顏色」濾鏡的亮度設定成 30.5：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].brightness = 30.5;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.color

適用版本
Flash 8。

用法
```
filter.color
```

說明
屬性；濾鏡的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

filter.name 屬性值為 "dropShadowFilter" 或 "glowFilter" 的 Filter 物件會定義此屬性。

範例
下列範例會將選取物件上「投影」濾鏡的顏色設定成 "#ff00003e"：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].color = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty()

filter.contrast

適用版本
Flash 8。

用法
```
filter.contrast
```

說明
屬性；浮點值，指定濾鏡的對比值。有效值介於 -100 到 100 之間。filter.name 屬性值為 "adjustColorFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的「調整顏色」濾鏡對比值設定成 -15.5：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].contrast = -15.5;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.distance

適用版本
Flash 8。

用法
```
filter.distance
```

說明
屬性；浮點值，指定濾鏡特效和物件之間的距離，以像素為單位。有效值介於 -255 到 255 之間。filter.name 屬性值為 "bevelFilter"、"dropShadowFilter"、"gradientBevelFilter" 或 "gradientGlowFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的「投影」濾鏡距離設定成 10 像素：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].distance = 10;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty()

filter.enabled

適用版本
Flash CS3 Professional。

用法
```
filter.enabled
```

說明
屬性；Boolean 值，指定要啟用 (true) 或停用 (false) 指定的濾鏡。

範例
下列範例會停用選取物件上的「顏色」濾鏡：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].enabled = false;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.hideObject

適用版本
Flash 8。

用法
```
filter.hideObject
```

說明
屬性；Boolean 值；指定要隱藏 (true) 或顯示 (false) 來源影像。filter.name 屬性值為 "dropShadowFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的「投影」濾鏡的 hideObject 值設定成 true：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'dropShadowFilter'){
        myFilters[i].hideObject = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.highlightColor

適用版本
Flash 8。

用法
```
filter.highlightColor
```

說明
屬性；反白標示的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

  filter.name 屬性值為 "bevelFilter" 的 Filter 物件會定義此屬性。

範例
下列範例會將選取物件上「斜角」濾鏡的反白標示顏色設定成 "#ff00003e"：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].highlightColor = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.hue

適用版本
Flash 8。

用法
```
filter.hue
```

說明
屬性；浮點值，指定濾鏡的色相。有效值介於 -180 到 180 之間。filter.name 屬性值為 "adjustColorFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的「調整顏色」濾鏡的色相設定成 120：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].hue = 120;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

filter.inner

適用版本
Flash 8。

用法
```
filter.inner
```

說明
屬性；Boolean 值；指定陰影是否為內部陰影，是為 true，不是則為 false。filter.name 屬性值為 "dropShadowFilter" 或 "glowFilter" 的 Filter 物件會定義此屬性。

範例
下列範例會將選取物件上「光暈」濾鏡的 inner 屬性值設定成 true：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].inner = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty()

filter.knockout

適用版本
Flash 8。

用法
```
filter.knockout
```

說明
屬性；Boolean 值；指定濾鏡是否為去底色濾鏡，是為 true，不是則為 false)。filter.name 屬性值為 "bevelFilter"、"dropShadowFilter"、"glowFilter"、"gradientBevelFilter" 或 "gradientGlowFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上「光暈」濾鏡的 knockout 屬性設定成 true：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].knockout = true;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty()

filter.name

適用版本
Flash 8。

用法
```
filter.name
```

說明
唯讀屬性；字串，指定濾鏡類型。這個屬性的值會決定 Filter 物件的哪些其它屬性是可用的。此值為下面其中之一："adjustColorFilter"、"bevelFilter"、"blurFilter"、"dropShadowFilter"、"glowFilter"、"gradientBevelFilter" 或 "gradientGlowFilter"。

範例
下列範例在「輸出」面板中顯示濾鏡名稱和索引位置：

```
var myFilters = fl.getDocumentDOM().getFilters();
var traceStr = "";
for(i=0; i < myFilters.length; i++){
    traceStr = traceStr + " At index " + i + ": " + myFilters[i].name;
}
fl.trace(traceStr);
```

請參閱
document.getFilters(), document.setFilterProperty()

filter.quality

適用版本
Flash 8。

用法
```
filter.quality
```

說明
屬性；字串，指定模糊化品質。可接受的值為 "low"、"medium" 和 "high" ("high" 類似高斯模糊)。filter.name 屬性值為 "bevelFilter"、"blurFilter"、"dropShadowFilter"、"glowFilter"、"gradientGlowFilter" 或 "gradientBevelFilter" 的 Filter 物件會定義此屬性。

範例
下列範例會將選取物件上「光暈」濾鏡的模糊化品質設定成 "medium"：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].quality = 'medium';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty()

filter.saturation

適用版本
Flash 8。

用法
```
filter.saturation
```

說明

屬性；浮點值，指定濾鏡的飽和度。有效值介於 -100 到 100 之間。filter.name 屬性值為 "adjustColorFilter" 的 Filter 物件會定義此屬性。

範例

下列範例會將選取物件上的 「調整顏色」濾鏡飽和度設定成 -100 (灰階)：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'adjustColorFilter'){
        myFilters[i].saturation = 0-100;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱

document.setFilterProperty()

filter.shadowColor

適用版本

Flash 8。

用法

```
filter.shadowColor
```

說明

屬性；陰影的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

filter.name 屬性值為 "bevelFilter" 的 Filter 物件會定義此屬性。

範例

下列範例會將選取物件上 「斜角」濾鏡的陰影顏色設定成 "#ff00003e"：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].shadowColor = '#ff00003e';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱

document.setFilterProperty()

filter.strength

適用版本
Flash 8。

用法
```
filter.strength
```

說明
屬性；整數，指定濾鏡的強度百分比。有效值介於 0 和 25,500 之間。filter.name 屬性值為 "bevelFilter"、"dropShadowFilter"、"glowFilter"、"gradientGlowFilter" 或 "gradientBevelFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的 「光暈」濾鏡的強度設定成 50：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'glowFilter'){
        myFilters[i].strength = 50;
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱
document.setFilterProperty()

filter.type

適用版本
Flash 8。

用法
```
filter.type
```

說明
屬性；字串，指定斜角或光暈濾鏡的類型。可接受的值為 "inner"、"outer" 和 "full"。filter.name 屬性值為 "bevelFilter"、"gradientGlowFilter" 或 "gradientBevelFilter" 的 Filter 物件會定義此屬性。

範例
下列範例將選取物件上的 「斜角」濾鏡類型設定成 "full"：

```
var myFilters = fl.getDocumentDOM().getFilters();
for(i=0; i < myFilters.length; i++){
    if(myFilters[i].name == 'bevelFilter'){
        myFilters[i].type = 'full';
    }
}
fl.getDocumentDOM().setFilters(myFilters);
```

請參閱

document.setFilterProperty()

第 17 章 flash 物件 (fl)

適用版本

Flash MX 2004。

說明

flash 物件代表 Flash 應用程式。您可以使用 flash 或 fl 來參照這個物件。本文件的所有程式碼樣本都使用 fl。

方法摘要

flash 物件可以搭配下列方法使用：

方法	說明
fl.addEventListener()	註冊要在收到特定事件時呼叫的函數。
fl.browseForFileURL()	開啟「開啟舊檔」或「儲存檔案」系統對話方塊，讓使用者指定要開啟或儲存的檔案。
fl.browseForFolderURL()	開啟「瀏覽檔案」對話方塊，讓使用者選取資料夾。
第 205 頁「fl.clearPublishCache()」	清除發佈快取。
fl.clipCopyString()	將指定的字串複製到「剪貼簿」。
fl.closeAll()	關閉所有開啟的文件，為先前未儲存的任何文件顯示「另存新檔」對話方塊。
fl.closeAllPlayerDocuments()	關閉所有使用「控制 > 測試影片」所開啟的 SWF 檔。
fl.closeDocument()	關閉指定的文件。
fl.createDocument()	開啟新文件並選取該文件。
第 212 頁「fl.exportPublishProfileString()」	匯出發佈設定的統一資源識別項 (URI)。
fl.fileExists()	檢查檔案是否已經存在於磁碟中。
fl.findDocumentDOM()	讓您使用唯一的識別名稱，將特定檔案設為目標。
fl.findDocumentIndex()	傳回整數陣列，代表文件在 fl.documents 陣列中的位置。
fl.findObjectInDocByName()	公開具有符合指定文字之實體名稱的元素。
fl.findObjectInDocByType()	公開文件中指定元素類型的元素。
fl.getAppMemoryInfo()	傳回整數，代表使用於 Flash.exe 記憶體指定區域中的位元組數目。
fl.getDocumentDOM()	擷取目前作用中文件的 DOM (Document 物件)。
fl.getSwfPanel()	依據面板的當地化名稱或其 SWF 檔案名稱傳回 SWFPanel 物件。
fl.isFontInstalled()	判斷是否已安裝指定的字體。
fl.mapPlayerURL()	將逸出的 Unicode URL 對應到 UTF-8 或 MBCS URL。
fl.openDocument()	開啟 Flash (FLA) 文件，以便在新的「Flash 文件」視窗中進行編輯，並且讓它成為焦點。
fl.openScript()	在 Flash 文字編輯器中開啟指令碼 (JSFL、AS、ASC) 或其它檔案 (XML、TXT)。

方法	說明
fl.quit()	結束 Flash，提示使用者儲存任何已變更的文件。
fl.reloadTools()	由 toolconfig.xml 檔重新建立「工具」面板。只能在建立可擴充工具時使用。
fl.removeEventListener()	取消註冊使用 fl.addEventListener() 所註冊的函數。
fl.resetAS3PackagePaths()	將 ActionScript 3.0 之「設定」對話方塊中的全域類別路徑設定重設為預設值。
fl.resetPackagePaths()	將 ActionScript 2.0 之「設定」對話方塊中的全域類別路徑設定重設為預設值。
fl.runScript()	執行 JavaScript 檔。
fl.saveAll()	儲存所有開啟的文件，為先前未儲存的任何文件顯示「另存新檔」對話方塊。
fl.saveDocument()	將指定文件儲存為 FLA 文件。
fl.saveDocumentAs()	為指定文件顯示「另存新檔」對話方塊。
fl.selectElement()	啟用元素的選取或編輯功能。
fl.selectTool()	在「工具」面板中選取指定的工具。
fl.setActiveWindow()	將作用中的視窗設定為指定文件。
fl.showIdleMessage()	讓您停用與指令碼執行太久有關的警告。
fl.toggleBreakpoint()	在指定的行切換指定 .as 檔案的中斷點。
fl.trace()	將文字字串傳送到「輸出」面板。

屬性摘要

flash 物件可以搭配下列屬性使用。

屬性	說明
fl.actionsPanel	唯讀；actionsPanel 物件。
fl.as3PackagePaths	字串，對應至 ActionScript 3.0 之「設定」對話方塊中的全域類別路徑設定。
fl.compilerErrors	唯讀；compilerErrors 物件。
fl.componentsPanel	唯讀；componentsPanel 物件，代表「組件」面板。
fl.configDirectory	唯讀；字串，將本機使用者的 Configuration 資料夾完整路徑指定為平台專用路徑。
fl.configURI	唯讀；字串，將本機使用者的 Configuration 目錄完整路徑指定為 file:/// URI。
fl.contactSensitiveSelection	Boolean 值，指定是否啟用接觸感應選取模式。
fl.createNewDocList	唯讀；字串陣列，代表能夠建立的各種文件類型。
fl.createNewDocListType	唯讀；字串陣列，代表能夠建立的文件類型副檔名。
fl.createNewTemplateList	唯讀；字串陣列，代表能夠建立的各種範本類型。
fl.documents	唯讀；Document 物件陣列 (請參閱 Document 物件)，代表目前開啟進行編輯的文件 (FLA 檔)。
fl.drawingLayer	唯讀；drawingLayer 物件，代表使用者在拖曳時若想暫時繪圖，可擴充工具所應使用的物件。

屬性	說明
fl.externalLibraryPath	字串，包含全域 ActionScript 3.0 外部元件庫路徑中的項目清單，而該路徑指定了做為執行階段共用元件庫使用的 SWC 檔案位置。
fl.flexSDKPath	字串，指定 Flex SDK 資料夾的路徑，該資料夾包含 bin、frameworks、lib 及其它資料夾。
fl.installedPlayers	傳回對應文件「屬性」檢測器中安裝之 Flash Player 的清單的一般物件陣列。
fl.languageCode	傳回識別應用程式使用者介面地區的五個字元碼。
fl.libraryPath	字串，包含全域 ActionScript 3.0 元件庫路徑中的項目清單，而該路徑指定了 SWC 檔案或包含 SWC 檔案之資料夾的位置。
fl.Math	唯讀；Math 物件，提供矩陣和點操作的方法。
fl.mruRecentFileList	唯讀；「最近使用」 (MRU) 清單的中完整檔案名稱陣列，由 Flash 編寫工具管理。
fl.mruRecentFileListType	唯讀；MRU 清單中的檔案類型陣列，由 Flash 編寫工具管理。
fl.packagePaths	字串，對應至 ActionScript 2.0 之「設定」對話方塊中的全域類別路徑設定。
fl.publishCacheDiskSizeMax	用來設定磁碟快取大小限制偏好設定的整數。
fl.publishCacheEnabled	用來設定是否啟用發佈快取的布林值。
fl.publishCacheMemoryEntrySizeLimit	用來設定記憶體快取項目大小上限偏好設定的整數屬性。
fl.publishCacheMemorySizeMax	用來設定記憶體快取大小限制偏好設定的整數。
fl.objectDrawingMode	整數，代表啟用的物件繪圖模式。
fl.outputPanel	唯讀；outputPanel 物件的參考。
fl.presetPanel	唯讀；presetPanel 物件。
fl.scriptURI	唯讀；字串，代表目前執行中 JSFL 指令碼的路徑，表示為 file:/// URI。
fl.sourcePath	字串，包含全域 ActionScript 3.0 來源路徑中的項目清單，而該路徑指定了 ActionScript 類別檔案的位置。
fl.swfPanels	已註冊的 swfPanel 物件陣列 (請參閱 swfPanel 物件)。
fl.tools	唯讀；Tools 物件的陣列。
fl.version	唯讀；Flash 編寫工具的長字串版本 (包括平台)。
fl.xmlui	唯讀；XMLUI 物件。

fl.actionsPanel

適用版本
Flash CS3 Professional。

用法
```
fl.actionsPanel
```

說明
唯讀屬性；actionsPanel 物件，代表目前顯示的「動作」面板。如需使用這項屬性的詳細資訊，請參閱 actionsPanel 物件。

fl.addEventListener()

適用版本
Flash CS3 Professional。

用法
```
fl.addEventListener(eventType, callbackFunction)
```

參數
eventType 字串，指定要傳遞給這個回呼函數的事件類型。可接受的值為 "documentNew"、"documentOpened"、"documentClosed"、"mouseMove"、"documentChanged"、"layerChanged" 和 "frameChanged"。

documentChanged 值不表示文件的內容已變更；其表示不同的文件現在位於前景中。也就是說，fl.getDocumentDOM() 將傳回不同的值，而不是在此事件發生前的值。

callbackFunction 每次發生事件時，您想要執行的函數名稱。

傳回值
無。

說明
方法；註冊要在特定事件發生時呼叫的函數。

使用這個方法時，請注意，如果事件經常發生 (如 mouseMove 的情況) 而且函數需要長時間才能執行，則應用程式可能會停止回應或進入錯誤狀態。

範例
下列範例會在關閉文件時，於 「輸出」面板中顯示一則訊息：

```
myFunction = function () {
    fl.trace('document was closed'); }
fl.addEventListener("documentClosed", myFunction);
```

請參閱
fl.removeEventListener()

fl.as3PackagePaths

適用版本
Flash CS3 Professional。

用法
```
fl.as3PackagePaths
```

說明
屬性；字串，對應至 ActionScript 3.0 之 「設定」對話方塊中的全域類別路徑設定。此字串中的項目會以分號隔開。若要檢視或變更 ActionScript 2.0 的類別路徑設定，請使用 fl.packagePaths。

範例
下列範例會說明如何變更 ActionScript 3.0 的類別路徑設定。

```
fl.trace(fl.as3PackagePaths);
// Output (assuming started with default value)
// .;$(AppConfig)/ActionScript 3.0/Classes
fl.as3PackagePaths="buying;selling";
fl.trace(fl.as3PackagePaths);
// Output
// buying; selling
```

請參閱
fl.resetAS3PackagePaths()

fl.browseForFileURL()

適用版本
Flash MX 2004。

用法

```
fl.browseForFileURL(browseType [, title [, previewArea]])
```

參數
browseType 字串，指定檔案瀏覽作業的類型。可接受的值為 "open"、"select" 或 "save"。"open" 和 "select" 值會開啟系統的 「開啟舊檔」對話方塊。每個提供的值皆相容於 Dreamweaver。"save" 值會開啟系統的 「儲存檔案」對話方塊。

title 字串，指定 「開啟舊檔」或 「儲存檔案」對話方塊的標題。如果省略這個參數，便會使用預設值。這個參數是選擇性參數。

previewArea 選擇性參數，遭 Flash 和 Fireworks 忽略，僅為與 Dreamweaver 相容而存在。

傳回值
檔案的 URL，以 file:///URI 形式來表示；若使用者取消對話方塊，則傳回 null。

說明
方法；開啟 「開啟舊檔」或 「儲存檔案」系統對話方塊，讓使用者指定要開啟或儲存的檔案。

範例
下列範例讓使用者選擇要開啟的 FLA 檔，並開啟檔案。(fl.browseForFileURL() 方法可以瀏覽任何檔案類型，但 fl.openDocument() 只能開啟 FLA 檔)。

```
var fileURL = fl.browseForFileURL("open", "Select file");
var doc = fl.openDocument(fileURL);
```

請參閱
fl.browseForFolderURL()

fl.browseForFolderURL()

適用版本
Flash 8。

用法
```
fl.browseForFolderURL([description])
```

參數
description 選擇性字串，指定 「瀏覽資料夾」對話方塊的說明。如果省略這個參數，則在說明區域無任何顯示。

傳回值
資料夾的 URL，以 file:///URI 形式來表示；若使用者取消對話方塊，則傳回 null。

說明
方法；顯示 「瀏覽資料夾」對話方塊，讓使用者選取資料夾。

備註：此對話方塊的標題一律為 「瀏覽資料夾」。請使用 description 參數在標題下方的說明區域中增加更多詳細資訊，例如「選取資料夾」或 「選取包含您想要匯入之描述檔的路徑」。

範例
下列範例會讓使用者選取資料夾，然後顯示該資料夾中的檔案清單：

```
var folderURI = fl.browseForFolderURL("Select a folder.");
var folderContents = FLfile.listFolder(folderURI);
```

請參閱
fl.browseForFileURL()、FLfile 物件

fl.clearPublishCache()

適用版本
Flash CS5.5 Professional。

用法
```
fl.clearPublishCache()
```

參數
無。

傳回值
無。

說明
方法；清空發佈快取。

範例
下列程式碼會清空發佈快取：

```
fl.clearPublishCache()
```

請參閱
fl.publishCacheDiskSizeMax、fl.publishCacheEnabled、第 227 頁「fl.publishCacheMemoryEntrySizeLimit」、第 228 頁「fl.publishCacheMemorySizeMax」

fl.clipCopyString()

適用版本
Flash CS3 Professional。

用法
```
fl.clipCopyString(string)
```

參數
string 要複製到「剪貼簿」的字串。

傳回值
無。

說明
方法；將指定的字串複製到「剪貼簿」。

若要將目前的選取範圍複製到「剪貼簿」，請使用 document.clipCopy()。

範例
下列範例會將目前文件的路徑複製到「剪貼簿」：

```
var documentPath = fl.getDocumentDOM().path;
fl.clipCopyString(documentPath);
```

fl.closeAll()

適用版本
Flash MX 2004。

用法
```
fl.closeAll([bPromptToSave])
```

參數
bPromptToSave 選擇性的 Boolean 值，指定是否要針對任何自先前儲存後已變更的檔案顯示「儲存」對話方塊，或是否要針對從未儲存檔案顯示「另存新檔」對話方塊。預設值為 true。

傳回值
無。

說明
方法：關閉所有開啟的檔案 (FLA 檔、SWF 檔和 JSFL 檔等)。如果您想要逕行關閉所有開啟的檔案，而不儲存任何變更，請將 false 傳遞給 bPromptToSave。這個方法無法終止應用程式。

範例
下列程式碼會關閉所有開啟的檔案，並提示使用者儲存任何全新或變更的檔案。

```
fl.closeAll();
```

請參閱
fl.closeAllPlayerDocuments()、fl.closeDocument()

fl.closeAllPlayerDocuments()

適用版本
Flash CS3 Professional。

用法
```
fl.closeAllPlayerDocuments()
```

參數
無。

傳回值
Boolean 值：如果有一個以上的影片視窗開啟，則為 true，否則為 false。

說明
方法：關閉所有使用 「控制 > 測試影片」所開啟的 SWF 檔。

範例
下列範例會關閉所有使用 「控制 > 測試影片」所開啟的 SWF 檔。

```
fl.closeAllPlayerDocuments();
```

請參閱
fl.closeAll()、fl.closeDocument()

fl.closeDocument()

適用版本
Flash MX 2004。

用法
```
fl.closeDocument(documentObject [, bPromptToSaveChanges])
```

參數
documentObject Document 物件。如果 documentObject 參考作用中的文件，則呼叫此方法的指令碼結束執行之前，「文件」視窗不會關閉。

bPromptToSaveChanges Boolean 值。當 bPromptToSaveChanges 為 false 時，如果文件包含未儲存的變更，則不會提示使用者；也就是說，會關閉檔案並捨棄變更。如果 bPromptToSaveChanges 為 true，而且文件包含未儲存的變更，則會以標準的「是」或「否」對話方塊提示使用者。預設值為 true。這個參數是選擇性參數。

傳回值
無。

說明
方法；關閉指定的文件。

範例
下列範例說明關閉文件的兩種方式。

```
// Closes the specified document and prompts to save changes.
fl.closeDocument(fl.documents[0]);
fl.closeDocument(fl.documents[0] , true); // Use of true is optional.
// Closes the specified document without prompting to save changes.
fl.closeDocument(fl.documents[0], false);
```

請參閱
fl.closeAll()

fl.compilerErrors

適用版本
Flash CS3 Professional。

用法
```
fl.compilerErrors
```

說明
唯讀屬性；compilerErrors 物件，代表「錯誤」面板。如需使用這項屬性的詳細資訊，請參閱 compilerErrors 物件。

fl.componentsPanel

適用版本
Flash MX 2004。

用法
```
fl.componentsPanel
```

說明
唯讀屬性；componentsPanel 物件，代表「組件」面板。

範例
下列範例會將 componentsPanel 物件儲存於 comPanel 變數中：

```
var comPanel = fl.componentsPanel;
```

fl.configDirectory

適用版本
Flash MX 2004。

用法
```
fl.configDirectory
```

說明
唯讀屬性；字串，使用平台專用格式指定本機使用者的 Configuration 目錄完整路徑。若要以 file:///URI 格式 (非平台專用格式) 指定此路徑，請使用 fl.configURI。

範例
下列範例會在「輸出」面板中顯示 Configuration 目錄：

```
fl.trace("My local configuration directory is " + fl.configDirectory);
```

fl.configURI

適用版本
Flash MX 2004。

用法
```
fl.configURI
```

說明
唯讀屬性；字串，將本機使用者的 Configuration 目錄完整路徑指定為 file:///URI。請參閱 fl.configDirectory。

範例
下列範例執行指定的指令碼。使用 fl.configURI 時，您無須瞭解執行指令碼的平台，就能指定指令碼的位置。

```
// To run a command in your commands menu, change "Test.Jsfl"
// to the command you want to run in the line below.
fl.runScript( fl.configURI + "Commands/Test.jsfl" );
```

fl.contactSensitiveSelection

適用版本
Flash 8。

用法
```
fl.contactSensitiveSelection
```

說明
Boolean 值，指定是 (true) 否 (false) 啟用接觸感應選取模式。

範例
下列範例說明如何在進行選擇前停用接觸感應選取模式，以及如何在進行選擇後，將其重設為原始值：

```
var contact = fl.contactSensitiveSelection;
fl.contactSensitiveSelection = false;
// Insert selection code here.
fl.contactSensitiveSelection = contact;
```

fl.createDocument()

適用版本
Flash MX 2004。

用法
```
fl.createDocument([docType])
```

參數
docType 字串，指定要建立的文件類型。可接受的值為 "timeline"、"presentation" 和 "application"。預設值為 "timeline"，這個值的作用與選擇 「檔案 > 新增 > Flash 檔案」 (ActionScript 3.0) 相同。這個參數是選擇性參數。

傳回值
如果方法成功，則為新建文件的 Document 物件。如果發生錯誤，則值為 undefined。

說明
方法：開啟新文件，並選取該文件。大小、解析度和顏色的值均與目前的預設值相同。

範例
下列範例會建立不同的文件類型：

```
// Create two Timeline-based Flash documents.
fl.createDocument();
fl.createDocument("timeline");
// Create a Slide Presentation document.
fl.createDocument("presentation");
// Create a Form Application document.
fl.createDocument("application");
```

fl.createNewDocList

適用版本
Flash MX 2004。

用法
```
fl.createNewDocList
```

說明
唯讀屬性：字串陣列，代表能夠建立的各種文件類型。

範例
下列範例會在「輸出」面板中顯示能夠建立的文件類型：

```
fl.trace("Number of choices " + fl.createNewDocList.length);
for (i = 0; i < fl.createNewDocList.length; i++)
    fl.trace("choice: " + fl.createNewDocList[i]);
```

fl.createNewDocListType

適用版本
Flash MX 2004。

用法
```
fl.createNewDocListType
```

說明
唯讀屬性：字串陣列，代表能夠建立的文件類型副檔名。在陣列中的項目會直接 (依索引) 對應至 fl.createNewDocList 陣列中的項目。

範例
下列範例會在「輸出」面板中顯示能夠建立的文件類型擴充功能：

```
fl.trace("Number of types " + fl.createNewDocListType.length);
for (i = 0; i < fl.createNewDocListType.length; i++) fl.trace("type: " + fl.createNewDocListType[i]);
```

fl.createNewTemplateList

適用版本
Flash MX 2004。

用法
```
fl.createNewTemplateList
```

說明
唯讀屬性：字串陣列，代表能夠建立的各種範本類型。

範例
下列範例會在「輸出」面板中顯示能夠建立的範本類型：

```
fl.trace("Number of template types: " + fl.createNewTemplateList.length); for (i = 0; i <
fl.createNewTemplateList.length; i++) fl.trace("type: " + fl.createNewTemplateList[i]);
```

fl.documents

適用版本
Flash MX 2004。

用法
```
fl.documents
```

說明
唯讀屬性：Document 物件陣列 (請參閱 Document 物件)，代表目前開啟進行編輯的文件 (FLA 檔)。

範例
下列範例會將已開啟文件的陣列儲存於 docs 變數中：

```
var docs = fl.documents;
```

下列範例會在 「輸出」面板中顯示目前開啟的文件名稱：

```
for (doc in fl.documents) {
    fl.trace(fl.documents[doc].name);
}
```

fl.drawingLayer

適用版本
Flash MX 2004。

用法
```
fl.drawingLayer
```

說明
唯讀屬性：drawingLayer 物件，代表使用者在拖曳時 (例如，建立選取圈選範圍) 若想暫時繪圖，所應使用的可擴充工具物件。

範例
請參閱 drawingLayer.setColor()。

fl.exportPublishProfileString()

適用版本
Flash Professional CS5。

用法
```
fl.exportPublishProfileString( ucfURI [, profileName] )
```

參數
ucfURI 此字串用以指定匯出發佈設定的檔案統一資源識別項 (URI)。

profileName 此字串指定要匯出的描述檔名稱。這個參數是選擇性參數。

傳回值
字串。

說明
不需開啟檔案即可傳回特定文件的發佈描述檔。您也可以指定發佈描述檔，但這是選擇性作業。

範例
下列範例會讀取發佈描述檔字串：

```
var ppXML = "";
var ucfURI = fl.browseForFileURL("open", "select a FLA");
if (ucfURI && ucfURI.length > 0)
ppXML = fl.exportPublishProfileString(ucfURI);
fl.trace(ppXML);
```

fl.externalLibraryPath

適用版本
Flash CS4 Professional。

用法
```
fl.externalLibraryPath
```

說明
屬性；字串，代表全域 ActionScript 3.0 外部元件庫路徑中的項目清單，而該路徑則指定了做為執行階段共用元件庫使用的 SWC 檔案位置。此字串中的項目會以分號隔開。在編寫工具中，您可以選擇「編輯 > 偏好設定 > ActionScript > ActionScript 3.0 設定」來指定這些項目。

範例
以下範例會將 /SWC_runtime 資料夾新增至全域 ActionScript 3.0 外部元件庫路徑：

```
fl.trace(fl.externalLibraryPath);
fl.externalLibraryPath = "/SWC_runtime;" + fl.externalLibraryPath;
fl.trace(fl.externalLibraryPath);
```

請參閱
fl.flexSDKPath、fl.libraryPath、fl.sourcePath、document.externalLibraryPath

fl.fileExists()

適用版本
Flash MX 2004。

用法
```
fl.fileExists(fileURI)
```

參數
fileURI 字串 (以 file:///URI 形式來表示)，其中包含檔案的路徑。

傳回值
Boolean 值：如果磁碟中存在檔案，則為 true，否則為 false。

說明
方法：檢查磁碟中是否已存在檔案。

範例
下列範例會依檔案是否存在，決定每項指定檔案應在 「輸出」面板中顯示 true 或 false。

```
alert(fl.fileExists("file:///C|/example.fla"));
alert(fl.fileExists("file:///C|/example.jsfl"));
alert(fl.fileExists(""));
```

fl.findDocumentDOM()

適用版本
Flash CS3 Professional。

用法
```
fl.findDocumentDOM(id)
```

參數
id 整數，代表文件的唯一識別名稱。

傳回值
Document 物件，但是如果不存在包含指定 id 的文件，則為 null。

說明
方法：讓您使用唯一的識別名稱 (而非索引值)，將特定檔案設為目標。與 document.id 搭配使用此方法。

範例
下列範例會說明如何讀取文件的 ID，然後使用此 ID，將該文件設為目標：

```
var originalDocID = fl.getDocumentDOM().id;
// other code here, maybe working in different files
var targetDoc = fl.findDocumentDOM(originalDocID);
// Set the height of the Stage in the original document to 400 pixels.
targetDoc.height = 400;
```

請參閱
fl.findDocumentIndex()

fl.findDocumentIndex()

適用版本
Flash MX 2004。

用法
```
fl.findDocumentIndex(name)
```

參數
name 要搜尋索引的文件名稱。此文件必須開啟。

傳回值
整數陣列，代表文件 name 在 fl.documents 陣列中的位置。

說明
方法；傳回整數陣列，代表文件 name 在 fl.documents 陣列中的位置。可以開啟一個以上的同名文件 (如果文件位於不同的資料夾中)。

範例
下列範例所顯示的資訊，與「輸出」面板中任何名為 test.fla 的開啟檔案的索引位置相關。

```
var filename = "test.fla"
var docIndex = fl.findDocumentIndex(filename);
for (var index in docIndex)
    fl.trace(filename + " is open at index " + docIndex[index]);
```

請參閱
fl.documents、fl.findDocumentDOM()

fl.findObjectInDocByName()

適用版本
Flash CS3 Professional。

用法
```
fl.findObjectInDocByName(instanceName, document)
```

參數
instanceName 字串，指定特定文件中某個項目的實體名稱。

document 要在其中搜尋指定項目的 Document 物件。

傳回值
一般物件的陣列。您可以使用陣列中每個項目的 .obj 屬性來取得物件。物件具有下列屬性：keyframe、layer、timeline 和 parent。您可以使用這些屬性來存取物件的階層架構。如需有關這些屬性以及如何加以存取的詳細資訊，請參閱 fl.findObjectInDocByType()。

您也可以存取 layer 和 timeline 值的方法和屬性；它們分別等同於 Layer 物件 和 Timeline 物件。

說明

方法：公開文件中具有符合指定文字之實體名稱的元素。

備註：在某些情況下，這個方法只有在 FLA 檔內以命令形式執行時才有用；如果目前正在檢查或編輯 JSFL 檔，則無法運作。

範例

下列範例會在目前文件中搜尋名為 "instance01" 的元素。

```
var nameToSearchFor = "instance01";
var doc = fl.getDocumentDOM();
var results = fl.findObjectInDocByName(nameToSearchFor, doc);
if (results.length > 0) {
    alert("success, found " + results.length + " objects");
}
else {
    alert("failed, no objects named " + nameToSearchFor + " found");
}
```

請參閱

fl.findObjectInDocByType()

fl.findObjectInDocByType()

適用版本

Flash CS3 Professional。

用法

```
fl.findObjectInDocByType(elementType, document)
```

參數

elementType 字串，代表要搜尋的元素類型。如需可接受的值，請參閱 element.elementType。

document 要在其中搜尋指定項目的 Document 物件。

傳回值

一般物件的陣列。您可以使用陣列中每個項目的 .obj 屬性來取得 Element 物件。每個物件都具有下列屬性：keyframe、layer、timeline 和 parent。您可以使用這些屬性來存取物件的階層架構。

您也可以存取 layer 和 timeline 值的方法和屬性；它們分別等同於 Layer 物件 和 Timeline 物件。

「範例」部分中的第二和第三個範例會說明如何存取這些屬性。

說明

方法：公開文件中指定元素類型的元素。

備註：在某些情況下，這個方法只有在 FLA 檔內以命令形式執行時才有用；如果目前正在檢查或編輯 JSFL 檔，則無法運作。

範例

下列範例會在目前文件中搜尋文字欄位，然後變更其內容：

```
var doc = fl.getDocumentDOM();
var typeToSearchFor = "text";
var results = fl.findObjectInDocByType(typeToSearchFor, doc);
if (results.length > 0) {
    for (var i = 0; i < results.length; i++) {
        results[i].obj.setTextString("new text");
    }
    alert("success, found " + results.length + " objects");
}
else {
    alert("failed, no objects of type " + typeToSearchFor + " found");
}
```

下列範例會說明如何存取由此方法傳回之物件的特殊屬性：

```
var doc = fl.getDocumentDOM();
var resultsArray = findObjectInDocByType("text", doc);
if (resultsArray.length > 0)
{
    var firstItem = resultsArray[0];

    // firstItem.obj- This is the element object that was found.

    // You can access the following properties of this object:
    // firstItem.keyframe- The keyframe that the element is on.
    // firstItem.layer- The layer that the keyframe is on.
    // firstItem.timeline- The timeline that the layer is on.
    // firstItem.parent- The parent of the timeline. For example,
    //      the timeline might be in a symbol instance.
}
```

下列範例說明如何使用 resultArray.obj 物件備份 DOM，以尋找文字欄位所在的圖層名稱：

```
var doc = fl.getDocumentDOM();
var typeToSearchFor = "text";
var resultsArray = fl.findObjectInDocByType(typeToSearchFor, doc);
if (resultsArray.length > 0) {
    for (var i = 0; i < resultsArray.length; i++) {
        resultsArray[i].obj.setTextString("new text");
        var firstItem = resultsArray[0];
        firstItemObj = firstItem.obj;
        fl.trace(firstItemObj.layer.name+"layerName");
    }
} else {
    alert("failed, no objects of type " + typeToSearchFor + " found");
}
```

請參閱
fl.findObjectInDocByName()

fl.flexSDKPath

適用版本
Flash CS4 Professional。

用法

```
fl.flexSDKPath
```

說明
屬性；字串，指定 Flex SDK 資料夾路徑，該資料夾包含 bin、frameworks、lib 及其他資料夾。在編寫工具中，您可以選擇「編輯 > 偏好設定 > ActionScript > ActionScript 3.0 設定」來指定這些項目。

範例
下列程式碼將在 「輸出」面板中顯示 Flex SDK 路徑：

```
fl.trace(fl.flexSDKPath);
```

請參閱
fl.externalLibraryPath、fl.libraryPath、fl.sourcePath

fl.getAppMemoryInfo()

適用版本
Flash 8 (僅適用於 Windows)。

用法
```
fl.getAppMemoryInfo(memType)
```

參數
memType 整數，指定要查詢的記憶體使用區域。如需可接受的值清單，請參閱以下說明。

傳回值
整數，代表使用於 Flash.exe 記憶體的指定區域中的位元組數目。

說明
方法 (僅適用於 Windows)：傳回整數，代表 Flash.exe 記憶體指定區域中使用的位元組數目。請使用下表決定要傳遞為 memType 的值：

memType	來源資料
0	PAGEFAULTCOUNT
1	PEAKWORKINGSETSIZE
2	WORKINGSETSIZE
3	QUOTAPEAKPAGEDPOOLUSAGE
4	QUOTAPAGEDPOOLUSAGE
5	QUOTAPEAKNONPAGEDPOOLUSAGE
6	QUOTANONPAGEDPOOLUSAGE
7	PAGEFILEUSAGE
8	PEAKPAGEFILEUSAGE

範例
下列範例會顯示目前運作的記憶體消耗量：

```
var memsize = fl.getAppMemoryInfo(2);
fl.trace("Flash current memory consumption is " + memsize + " bytes or " + memsize/1024 + " KB");
```

fl.getDocumentDOM()

適用版本
Flash MX 2004。

用法
```
fl.getDocumentDOM()
```

參數
無。

傳回值
Document 物件，如果沒有開啟任何的文件，則為 null。

說明
方法；擷取目前作用中文件 (FLA 檔) 的 DOM (Document 物件)。如果開啟一或多份文件，但文件不是目前的焦點 (例如，焦點為 JSFL 檔)，則擷取最新作用中文件的 DOM。

範例
下列範例在 「輸出」面板中顯示目前或最新作用中的文件名稱：

```
var currentDoc = fl.getDocumentDOM();
fl.trace(currentDoc.name);
```

fl.getSwfPanel()

適用版本
Flash CS5.5 Professional。

用法
```
fl.getSwfPanel(panelName, [useLocalizedPanelName])
```

參數
panelName 當地化的面板名稱或面板之 SWF 檔案的根檔案名稱。如果使用後者，會傳送 false 作為第二個參數。

useLocalizedPanelName 這是選擇性的。預設值為 true。如果是 false，panelName 參數會被假設為面板的英文名稱 (未當地化)，相對應於沒有副檔名的 SWF 檔案名稱。

傳回值
SWFPanel 物件。

說明
方法；依據面板的當地化名稱或其 SWF 檔案名稱 (沒有副檔名) 傳回 SWFPanel 物件。

範例
下列範例顯示「輸出」面板中稱為「專案」的面板名稱：

```
fl.trace('name of panel is: ' + fl.getSwfPanel('Project').name);
```

fl.installedPlayers

適用版本
Flash CS5.5 Professional。

用法
```
fl.installedPlayers()
```

參數
無。

傳回值
對應文件 PI 中安裝之 Flash Player 的清單的一般物件陣列。

說明
「唯讀」屬性；對應文件 PI 中所安裝之 Flash Player 的清單的一般物件陣列。

陣列中的每個物件都包含下列屬性：

name 文件的字串名稱。

version 可用來設定文件目前的播放器 (使用 Document.setPlayerVersion() 函數)。

minASVersion 文件所需的最低 ActionScript 版本。介於 minASVersion 和 maxASVersion (含) 之間的整數，可用來設定文件的 ActionScript 版本 (使用 Document.asVersion 屬性)。

maxASVersion 文件支援的最高 ActionScript 版本。

stageWidth 指定之目標的預設「舞台」寬度像素。例如，iPhone 的預設大小為 320 x 480 像素，而 Android 的預設大小為 480 x 800。

stageHeight 指定之目標的預設 「舞台」高度像素。例如，iPhone 的預設大小為 320 x 480 像素，而 Android 的預設大小為 480 x 800。

範例
下列範例會追蹤 installedPlayers 陣列中所有物件的屬性並輸出至輸出視窗。

```
var arr = fl.installedPlayers;
for (var i in arr) fl.trace("name: " + arr[i].name + " version: " + arr[i].version + " minASVersion: " +
arr[i].minASVersion + " maxASVersion: " + arr[i].maxASVersion + " stageWidth: " + arr[i].stageWidth + "
stageHeight: " + arr[i].stageHeight + " ");
```

fl.isFontInstalled()

適用版本
Flash CS4 Professional。

用法
```
fl.isFontInstalled(fontName)
```

參數
fontName 字串，指定裝置字體的名稱。

傳回值
如果指定的字體已安裝，會傳回 true 的 Boolean 值，否則便傳回 false。

說明
方法；判斷是否已安裝指定的字體。

範例
如果 Times 字體已安裝，則下列程式碼會在「輸出」面板中顯示 "true"。
```
fl.trace(fl.isFontInstalled("Times"));
```

fl.languageCode

適用版本
Flash CS5 Professional。

用法
```
fl.languageCode
```

說明
屬性；傳回識別應用程式使用者介面地區的五個字元碼字串。

範例
下列範例傳回 Flash 應用程式當地語系化使用者介面所指定的五個字元語言碼：
```
 locConfigURI = fl.applicationURI + fl.languageCode + "/Configuration";
```

fl.libraryPath

適用版本
Flash CS4 Professional。

用法
```
fl.libraryPath
```

說明
屬性；字串，包含全域 ActionScript 3.0 元件庫路徑中的項目清單，而該路徑則指定了 SWC 檔案或包含 SWC 檔案之資料夾位置。此字串中的項目會以分號隔開。在編寫工具中，您可以選擇「編輯 > 偏好設定 > ActionScript > ActionScript 3.0 設定」來指定這些項目。

範例
下列範例會將 /SWC 資料夾新增至全域 ActionScript 3.0 元件庫路徑：

```
fl.trace(fl.libraryPath);
fl.libraryPath = "/SWC;" + fl.libraryPath;
fl.trace(fl.libraryPath);
```

請參閱
fl.externalLibraryPath、fl.flexSDKPath、fl.sourcePath、document.libraryPath

fl.mapPlayerURL()

適用版本
Flash MX 2004。

用法
```
fl.mapPlayerURL(URI [, returnMBCS])
```

參數
URI 字串，包含要對應的逸出 Unicode URL。

returnMBCS Boolean 值，如果要傳回逸出 MBCS 路徑，則必須將其設定為 true。否則，方法會傳回 UTF-8。此預設值為 false。這個參數是選擇性參數。

傳回值
字串，為轉換的 URL。

說明
方法；將逸出的 Unicode URL 對應到 UTF-8 或 MBCS URL。在 ActionScript 中使用字串來存取外部資源時，請使用此方法。如果需要處理多位元組字元，則必須使用此方法。

範例
下列範例會將 URL 轉換為 UTF-8，讓播放程式能夠載入：

```
var url = MMExecute( "fl.mapPlayerURL(" + myURL + ", false);" );
mc.loadMovie( url);
```

fl.Math

適用版本
Flash MX 2004。

用法
```
fl.Math
```

說明
唯讀屬性；Math 物件會提供矩陣和點操作的方法。

範例

下列範例會顯示選取物件的變形矩陣及其逆矩陣：

```
// Select an element on the Stage and then run this script.
var mat =fl.getDocumentDOM().selection[0].matrix;
for(var prop in mat){
fl.trace("mat."+prop+" = " + mat[prop]);
}
var invMat = fl.Math.invertMatrix( mat );
for(var prop in invMat) {
fl.trace("invMat."+prop+" = " + invMat[prop]);
}
```

fl.mruRecentFileList

適用版本

Flash MX 2004。

用法

```
fl.mruRecentFileList
```

說明

唯讀屬性：「最近使用」 (MRU) 清單中的完整檔案名稱陣列，由 Flash 編寫工具管理。

範例

下列範例會在 「輸出」面板中顯示最近開啟的檔案數目，以及每個檔案的名稱：

```
fl.trace("Number of recently opened files: " + fl.mruRecentFileList.length);
for (i = 0; i < fl.mruRecentFileList.length; i++) fl.trace("file: " + fl.mruRecentFileList[i]);
```

fl.mruRecentFileListType

適用版本

Flash MX 2004。

用法

```
fl.mruRecentFileListType
```

說明

唯讀屬性：MRU 清單中的檔案類型陣列，由 Flash 編寫工具管理。此陣列對應於 fl.mruRecentFileList 屬性中的陣列。

範例

下列範例會在 「輸出」面板中顯示最近開啟的檔案數目，以及每個檔案的類型：

```
fl.trace("Number of recently opened files: " + fl.mruRecentFileListType.length);
for (i = 0; i < fl.mruRecentFileListType.length; i++) fl.trace("type: " + fl.mruRecentFileListType[i]);
```

fl.objectDrawingMode

適用版本
Flash 8。

用法
```
fl.objectDrawingMode
```

說明
屬性；Boolean 值，指定啟用物件繪圖模式 (true) 或啟用合併繪圖模式 (false)。

範例
下列範例會切換物件繪圖模式的狀態。

```
var toggleMode = fl.objectDrawingMode;
if (toggleMode) {
    fl.objectDrawingMode = false;
} else {
    fl.objectDrawingMode = true;
}
```

fl.openDocument()

適用版本
Flash MX 2004。

用法
```
fl.openDocument(fileURI)
```

參數
fileURI 字串 (以 file:///URI 形式來表示)，指定要開啟的檔案名稱。

傳回值
如果方法成功，則為新開啟文件的 Document 物件。如果找不到檔案，或無有效的 FLA 檔，則會報告錯誤，並且取消指令碼。

說明
方法；開啟 Flash 文件 (FLA 檔)，以便在新的 「Flash 文件」視窗中進行編輯，並且讓文件成為焦點。對使用者來說，特效等於選取 「檔案 > 開啟」，然後選取檔案。如果已開啟指定檔案，則包含文件的視窗會移到最上層。包含指定檔案的視窗會成為目前選取文件。

範例
下列範例會開啟儲存於 C 磁碟機根目錄中名為 Document.fla 的檔案。此程式碼會將代表該文件的 Document 物件儲存於 doc 變數中，然後將文件設定為目前選取的文件。也就是說，直到變更焦點之前，fl.getDocumentDOM() 都會參照此文件。

```
var doc = fl.openDocument("file:///c|/Document.fla");
```

fl.openScript()

適用版本
Flash MX 2004。在 Flash Professional CS5 中已加入選擇性參數。

用法
```
fl.openScript(fileURI [, createExtension, className])
```

參數
fileURI 字串 (表示為 file:/// URI)，指定應該載入 Flash 文字編輯器的 JSFL、AS、ASC、XML、TXT 或其它檔案的路徑。此參數可以是 Null。如果為 Null，則此方法會開啟 createExtension 參數所指定之類型的新指令碼。

createExtension 字串，決定 fileURI 為 Null 時所要建立的文件類型。預設值為 'AS'；允許的值包括 'JSFL'、'AS'、'ASC'、'XML'、'TXT'、'AS3_CLASS' 或 'AS3_INTERFACE'。在 Flash Professional CS5 中已加入此參數。

className 字串，指定已建立之類別或介面 (由 createExtension 參數決定) 的完整類別名稱。在 Flash Professional CS5 中已加入此參數。

傳回值
無。

說明
方法；會在 Flash 文字編輯器中開啟現有的檔案、建立新的指令碼 (JSFL、AS、ASC) 或其他檔案 (XML、TXT)。

範例
下列範例會開啟儲存於 C 磁碟機 /temp 目錄中名為 my_test.jsfl 的檔案：

```
fl.openScript("file:///c|/temp/my_test.jsfl");
```

範例
下列範例會建立具有空白 AS3 類別定義的新 .as 檔案：

```
fl.openScript(null, 'AS3_CLASS');
```

fl.outputPanel

適用版本
Flash MX 2004。

用法
```
fl.outputPanel
```

說明
唯讀屬性；參照到 outputPanel 物件。

範例
請參閱 outputPanel 物件。

fl.packagePaths

適用版本
Flash CS3 Professional。

用法
```
fl.packagePaths
```

說明
屬性；字串，對應至 ActionScript 2.0 之「設定」對話方塊中的全域類別路徑設定。此字串中的類別路徑會以分號 (;) 隔開。若要檢視或變更 ActionScript 3.0 的類別路徑設定，請使用 fl.as3PackagePaths。

範例
下列範例說明如何變更 ActionScript 2.0 的類別路徑設定：

```
fl.trace(fl.packagePaths);
// Output (assuming started with default value)
// .;$(LocalData)/Classes
fl.packagePaths="buying;selling";
fl.trace(fl.packagePaths);
// Output
// buying; selling
```

請參閱
fl.resetPackagePaths()

fl.presetPanel

適用版本
Flash CS4 Professional。

用法
```
fl.presetPanel
```

說明
唯讀屬性：presetPanel 物件。

fl.publishCacheDiskSizeMax

適用版本
Flash CS5.5 Professional。

用法
```
fl.publishCacheDiskSizeMax
```

說明
屬性：用來設定磁碟上的發佈快取之大小上限 (MB) 的整數。

範例
下列程式碼會將磁碟上的發佈快取大小上限設定為 1MB：

```
fl.publishCacheDiskSizeMax = 1
```

請參閱
fl.clearPublishCache()、fl.publishCacheEnabled、第 227 頁「fl.publishCacheMemoryEntrySizeLimit」、第 228 頁「fl.publishCacheMemorySizeMax」

fl.publishCacheEnabled

適用版本
Flash CS5.5 Professional。

用法
```
fl.publishCacheEnabled
```

說明
屬性：用來設定是否啟用發佈快取的布林值。

範例
下列程式碼會顯示是否有在「輸出」視窗中啟用發佈快取。

```
fl.trace(fl.publishCacheEnabled);
```

請參閱
fl.publishCacheDiskSizeMax、fl.clearPublishCache()、第 227 頁「fl.publishCacheMemoryEntrySizeLimit」、第 228 頁「fl.publishCacheMemorySizeMax」

fl.publishCacheMemoryEntrySizeLimit

適用版本
Flash CS5.5 Professional。

用法
```
fl.publishCacheMemoryEntrySizeLimit
```

說明
屬性：用來設定可新增至發佈快取之項目的大小上限 (KB) 的整數。等於或小於此大小的項目會保留在記憶體中，大於此大小的項目則會寫入磁碟。

若使用者具備充足記憶體，可以增加此值以增進效能，而記憶體較不足的使用者則可以降低此值，避免發佈快取消耗過多記憶體。

範例
下列程式碼會將可在記憶體中儲存的發佈快取項目大小上限設為 100KB：

```
fl.publishCacheMemoryEntrySizeLimit = 100
```

請參閱
fl.publishCacheDiskSizeMax、fl.publishCacheEnabled、第 205 頁「fl.clearPublishCache()」、第 228 頁「fl.publishCacheMemorySizeMax」

fl.publishCacheMemorySizeMax

適用版本
Flash CS5.5 Professional。

用法
```
fl.publishCacheMemorySizeMax
```

說明
屬性：用來設定記憶體中發佈快取之大小上限 (MB) 的整數。

範例
下列程式碼會將記憶體中發佈快取大小上限設定為 1MB：

```
fl.publishCacheMemorySizeMax = 1
```

請參閱
fl.publishCacheDiskSizeMax、fl.publishCacheEnabled、第 227 頁「fl.publishCacheMemoryEntrySizeLimit」、第 205 頁「fl.clearPublishCache()」

fl.publishDocument()

適用版本
Flash CS5 Professional。

用法
```
fl.publishDocument( flaURI [, publishProfile] )
```

參數
flaURI 字串 (以 file:///URI 形式表示)，指定應以無訊息方式發佈的 FLA 檔案路徑。

publishProfile 此字串用以指定發佈時要使用的發佈描述檔。如果省略此參數，便會使用預設的發佈描述檔。

傳回值
Boolean

說明
方法；不需開啟即可發佈 FLA 檔案。此 API 會在無標頭模式下開啟 FLA 並發佈 SWF (或描述檔所設定的任何項目)。第二個參數 (publishProfile) 是選擇性參數。傳回值是 Boolean 值，指出是否找到描述檔。如果沒有提供第二個參數，則傳回的值永遠為 true。

範例
下列範例會提示使用者選取 FLA 檔案，並使用 “Default” 發佈描述檔以無訊息方式發佈該 FLA 檔案：

```
var uri = fl.browseForFileURL("select", "select a FLA file to publish");
var publishProfileName = "Default";
fl.publishDocument(uri, publishProfileName);
```

fl.quit()

適用版本
Flash MX 2004。

用法
```
fl.quit([bPromptIfNeeded])
```

參數
bPromptIfNeeded Boolean 值，如果要提示使用者儲存任何修改的文件，則該值為 true (預設值)。如果不要提示使用者儲存修改的文件，則將此參數設為 false。前述第二個情況中，開啟文件中的任何修改都會遭到捨棄，而且應用程式會立即結束。此方法對於批次處理雖然十分有用，但請務必謹慎使用。這個參數是選擇性參數。

傳回值
無。

說明
方法：結束 Flash，提示使用者儲存任何變更的文件。

範例
下列範例會說明如何在結束時詢問及不詢問儲存修改的文件：

```
// Quit with prompt to save any modified documents.
fl.quit();
fl.quit(true); // True is optional.
// Quit without saving any files.
fl.quit(false);
```

fl.reloadEffects()

適用版本
Flash MX 2004。

用法
```
fl.reloadEffects()
```

參數
無。

傳回值
無。

說明
方法：重新載入使用者 Configuration Effects 資料夾中定義的所有特效描述器。此方法允許您在開發時快速變更指令碼，並提供不需重新啟動應用程式就能改善特效的機制。用於 Commands 資料夾中的命令時，此方法的效果最好。

範例
下列範例為可放置在 Commands 資料夾中的單行指令碼。在必須重新載入特效時，請到 「命令」選單並執行此指令碼。

```
fl.reloadEffects();
```

fl.reloadTools()

適用版本
Flash MX 2004。

用法
```
fl.reloadTools()
```

參數
無。

傳回值
無。

說明
方法：由 toolconfig.xml 檔案重新建立 「工具」面板。此方法僅使用在建立可擴充的工具時。在需要重新載入 「工具」面板時，請使用這個方法，例如，在修改 JSFL 檔之後 (此檔案定義的工具已位在面板中)。

範例
下列範例為可放置在 Commands 資料夾中的單行指令碼。當您需要重新載入 「工具」面板時，請從 「命令」選單執行指令碼。

```
fl.reloadTools();
```

fl.removeEventListener()

適用版本
Flash CS3 Professional。

用法
```
fl.removeEventListener(eventType)
```

參數
eventType 字串，指定要從這個回呼函數中移除的事件類型。可接受的值為 "documentNew"、"documentOpened"、"documentClosed"、"mouseMove"、"documentChanged"、"layerChanged" 和 "frameChanged"。

傳回值
如果成功移除事件偵聽程式，則傳回 Boolean 值 true；如果從未使用 fl.addEventListener() 方法將函數加入至清單，則為 false。

說明
取消註冊使用 fl.addEventListener() 所註冊的函數。

範例
下列範例會移除與 documentClosed 事件相關聯的事件偵聽程式：

```
fl.removeEventListener("documentClosed");
```

請參閱
fl.addEventListener()

fl.resetAS3PackagePaths()

適用版本
Flash CS3 Professional。

用法

```
fl.resetAS3PackagePaths()
```

參數
無。

說明
方法；將 ActionScript 3.0 之「設定」對話方塊中的全域類別路徑設定重設為預設值。若要重設 ActionScript 2.0 的全域類別路徑，請使用 fl.resetPackagePaths()。

範例
下列範例會將 ActionScript 3.0 的類別路徑設定重設為預設值。

```
fl.resetAS3PackagePaths();
```

請參閱
fl.as3PackagePaths

fl.resetPackagePaths()

適用版本
Flash CS3 Professional。

用法

```
fl.resetPackagePaths()
```

參數
無。

說明
方法：將 ActionScript 2.0 之「設定」對話方塊中的全域類別路徑設定重設為預設值。若要重設 ActionScript 3.0 的全域類別路徑，請使用 fl.resetAS3PackagePaths()。

範例
下列範例會將 ActionScript 2.0 的類別路徑設定重設為預設值。

```
fl.resetPackagePaths();
```

請參閱
fl.packagePaths

fl.revertDocument()

適用版本
Flash MX 2004。

用法
```
fl.revertDocument(documentObject)
```

參數
documentObject Document 物件。如果 documentObject 參照作用中的文件，則在呼叫此方法的指令碼結束執行之前，「文件」視窗不會回復。

傳回值
Boolean 值：如果成功完成回復作業，則為 true，否則為 false。

說明
方法：將指定的 FLA 文件回復成上一次儲存的版本。此方法不同於「檔案 > 回復」選單選項，並不會顯示警告視窗要求使用者確認此項作業。請參閱 document.revert() 與 document.canRevert()。

範例
下列範例將目前 FLA 檔回復成上一次儲存版本；上一次儲存後的任何變更都會遺失。

```
fl.revertDocument(fl.getDocumentDOM());
```

fl.runScript()

適用版本
Flash MX 2004。

用法
```
fl.runScript(fileURI [, funcName [, arg1, arg2, ...]])
```

參數
fileURI 字串 (以 file:///URI 形式來表示)，指定要執行的指令碼檔案名稱。

funcName 字串，識別要於 fileURI 中指定的 JSFL 檔中執行的函數。這個參數是選擇性參數。

arg 選擇性參數，指定一或多個要傳遞至 funcname 的引數。

傳回值

若指定了 funcName，就會將函數結果以字串形式傳回，否則不會傳回任何值。

說明

方法：執行 JavaScript 檔案。如果指定函數為其中的一個引數，會執行函數和不在函數內的指令碼中的任何程式碼。Script 中的其它程式碼會在執行函數之前執行。

範例

假設 C 磁碟根目錄中有一個名為 testScript.jsfl 的指令碼檔，且其內容如下：

```
function testFunct(num, minNum) {
    fl.trace("in testFunct: 1st arg: " + num + " 2nd arg: " + minNum);
}
for (i=0; i<2; i++) {
    fl.trace("in for loop i=" + i);
}
fl.trace("end of for loop");
// End of testScript.jsfl
```

如果您發出下列命令：

```
fl.runScript("file:///C|/testScript.jsfl", "testFunct", 10, 1);
```

「輸出」面板中會顯示下列資訊：

```
in for loop i=0
in for loop i=1
end of for loop
in testFunct: 1st arg: 10 2nd arg: 1
```

您不需執行函數也可以呼叫 testScript.jsfl，如下：

```
fl.runScript("file:///C|/testScript.jsfl");
```

這會在 「輸出」面板中產生下列資訊：

```
in for loop i=0
in for loop i=1
end of for loop
```

fl.saveAll()

適用版本

Flash MX 2004。

用法

```
fl.saveAll()
```

參數

無。

傳回值
無。

說明
方法：儲存所有開啟的文件。

如果檔案從未儲存，或者自從上次儲存後並未修改，則檔案不會儲存。若要允許未儲存或未修改的檔案進行儲存，請使用 fl.saveDocumentAs()。

範例
下列範例會儲存之前儲存過的所有已開啟文件，以及上次儲存以來已經變更的所有已開啟文件：

```
fl.saveAll();
```

請參閱
document.save()、document.saveAndCompact()、fl.saveDocument()、fl.saveDocumentAs()

fl.saveDocument()

適用版本
Flash MX 2004。

用法
```
fl.saveDocument(document [, fileURI])
```

參數
document Document 物件，指定要儲存的文件。如果 document 為 null，則會儲存作用中的文件。

fileURI 字串 (以 file:///URI 形式來表示)，指定儲存文件的名稱。如果 fileURI 參數為 null 或遭省略，則以目前名稱儲存文件。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功完成儲存作業，則為 true，否則為 false。

如果檔案從未儲存，或者自從上次儲存後並未修改，則檔案不會儲存，且會傳回 false。若要允許未儲存或未修改的檔案進行儲存，請使用 fl.saveDocumentAs()。

說明
方法：將指定文件儲存為 FLA 文件。

範例
下列範例會儲存目前文件和兩份指定的文件：

```
// Save the current document.
alert(fl.saveDocument(fl.getDocumentDOM()));
// Save the specified documents.
alert(fl.saveDocument(fl.documents[0], "file:///C|/example1.fla"));
alert(fl.saveDocument(fl.documents[1],"file:///C|/example2.fla"));
```

請參閱
document.save()、document.saveAndCompact()、fl.saveAll()、fl.saveDocumentAs()

fl.saveDocumentAs()

適用版本
Flash MX 2004。

用法
```
fl.saveDocumentAs(document)
```

參數
document Document 物件，指定要儲存的文件。如果 document 為 null，則會儲存作用中的文件。

傳回值
Boolean 值：如果成功完成 「另存新檔」作業，則為 true，否則為 false。

說明
方法；顯示指定文件的 「另存新檔」對話方塊。

範例
下列範例會提示使用者儲存指定的文件，然後顯示警告訊息，指出文件是否已儲存：

```
alert(fl.saveDocumentAs(fl.documents[1]));
```

請參閱
document.save()、document.saveAndCompact()、fl.saveAll()、fl.saveDocument()

fl.scriptURI

適用版本
Flash CS3 Professional。

用法
```
fl.scriptURI
```

說明
唯讀屬性；字串，以 file:///URI 來表示，代表目前執行中 JSFL 指令碼的路徑。如果指令碼是從 fl.runScript() 呼叫，這個屬性代表直屬父輩指令碼的路徑。也就是說，它不會詳細檢查多個 fl.runScript() 呼叫，以便尋找原始呼叫端指令碼的路徑。

範例
下列範例會在 「輸出」面板中顯示目前執行中 JSFL 指令碼的路徑：

```
fl.trace(fl.scriptURI);
```

請參閱
fl.runScript()

fl.selectElement()

適用版本
Flash CS3 Professional。

用法
```
fl.selectElement(elementObject, editMode)
```

參數
elementObject 您想要選取的 Element 物件。

editMode Boolean 值，指定您想要編輯元素 (true) 或只想要選取元素 (false)。

傳回值
如果成功選取元素，就會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；啟用元素的選取或編輯功能。一般來說，您會在由 fl.findObjectInDocByName() 或 fl.findObjectInDocByType() 傳回的物件上使用此方法。

範例
下列範例會選取名為 "second text field" 的元素 (如果在文件中找到元素的話)：

```
var nameToSearchFor = "second text field";
var doc = fl.getDocumentDOM();

// Start by viewing Scene 1 (index value of 0).
document.editScene(0);

// Search for element by name.
var results = fl.findObjectInDocByName(nameToSearchFor, doc);
if (results.length > 0) {
    // Select the first element found.
    // Pass false, so the symbolInstance you are searching for is selected.
    // If you pass true, the symbol instance will switch to edit mode.
    fl.selectElement(results[0], false);
    alert("success, found " + results.length + " objects")
    }
    else {
        alert("failed, no objects with name "" + nameToSearchFor + "" found");
    }
```

請參閱
fl.findObjectInDocByName()、fl.findObjectInDocByType()

fl.selectTool()

適用版本
Flash CS3 Professional。

用法

```
fl.selectTool(toolName)
```

參數

toolName 字串，指定要選取的工具名稱。如需這個參數可接受值的詳細資訊，請參閱下面的 「說明」。

說明

方法：在 「工具」面板中選取指定的工具。toolName 的可接受預設值包括 "arrow"、"bezierSelect"、"freeXform"、"fillXform"、"lasso"、"pen"、"penplus"、"penminus"、"penmodify"、"text"、"line"、"rect"、"oval"、"rectPrimitive"、"ovalPrimitive"、"polystar"、"pencil"、"brush"、"inkBottle"、"bucket"、"eyeDropper"、"eraser"、"hand" 和 "magnifier"。

如果您或使用者建立了自訂工具，這些工具的名稱也可以傳遞成 toolName 參數。工具名稱的清單位於下列檔案中：

- Windows Vista：

  < 開機磁碟 >\Users\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS3\< 語言 >\Configuration\Tools\toolConfig.xml

- Windows XP：

  < 開機磁碟 >\Documents and Settings\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS3\< 語言 >\Configuration\Tools\toolConfig.xml

- Mac OS X：

  Macintosh HD/Users/< 使用者名稱 >/Library/Application Support/Adobe/Flash CS3/< 語言 >/Configuration/Tools/toolConfig.xml

範例

下列範例會選取 「鋼筆」工具。

```
fl.selectTool("pen");
```

請參閱

Tools 物件、ToolObj 物件

fl.setActiveWindow()

適用版本

Flash MX 2004。

用法

```
fl.setActiveWindow(document [, bActivateFrame])
```

參數

document Document 物件，指定要選取為作用中視窗的文件。

bActivateFrame 選擇性參數，遭 Flash 和 Fireworks 忽略，僅為與 Dreamweaver 相容而存在。

傳回值

無。

說明
方法：將作用中的視窗設為指定文件。Dreamweaver 和 Fireworks 也支援此方法。如果文件有多個檢視 (由「視窗 > 多重視窗」建立)，則會選取最近一個作用中檢視。

範例
下列範例會顯示兩種啟用指定文件的方式：

```
fl.setActiveWindow(fl.documents[0]);

var theIndex = fl.findDocumentIndex("myFile.fla");
fl.setActiveWindow(fl.documents[theIndex]);
```

fl.showIdleMessage()

適用版本
Flash 8。

用法
```
fl.showIdleMessage(show)
```

參數
show Boolean 值，指定要啟用或停用指令碼執行過久的警告。

傳回值
無。

說明
方法：讓您停用指令碼執行過久的警告 (針對 show 傳遞 false)。需要長時間處理批次作業時，您可以執行此方法。若要重新啟用警告，請再次發出命令，而且這次針對 show 傳遞 true。

範例
下列範例會說明如何停用與重新啟用指令碼執行過久的警告：

```
fl.showIdleMessage(false);
var result = timeConsumingFunction();
fl.showIdleMessage(true); ;
var result = timeConsumingFunction();
```

fl.sourcePath

適用版本
Flash CS4 Professional。

用法
```
fl.sourcePath
```

說明

屬性；字串，包含全域 ActionScript 3.0 來源路徑中的項目清單，而該路徑則指定了 ActionScript 類別檔案的位置。此字串中的項目會以分號隔開。在編寫工具中，您可以選擇「編輯 > 偏好設定 > ActionScript > ActionScript 3.0 設定」來指定這些項目。

範例

下列範例會將 /Classes 資料夾新增至全域 ActionScript 3.0 來源路徑：

```
fl.trace(fl.sourcePath);
fl.sourcePath = "/Classes;" + fl.sourcePath;
fl.trace(fl.sourcePath);
```

請參閱

fl.flexSDKPath、fl.externalLibraryPath、fl.libraryPath、document.sourcePath

fl.swfPanels

適用版本

Flash CS4 Professional。

用法

```
fl.swfPanels
```

說明

唯讀屬性；已註冊的 swfPanel 物件陣列 (請參閱 swfPanel 物件)。如果 swfPanel 物件已經至少開啟一次，則會加以註冊。

陣列中的面板位置代表面板的開啟順序。如果第一個開啟的面板名為 TraceBitmap 而第二個開啟的面板名為 AnotherFunction，則 fl.swfPanels[0] 為 TraceBitmap swfPanel 物件，而 fl.swfPanels[1] 則為 AnotherFunction swfPanel 物件，以此類推。

範例

下列程式碼會在「輸出」面板中顯示任何已註冊 Window SWF 面板的名稱與路徑：

```
if(fl.swfPanels.length > 0){
    for(x = 0; x < fl.swfPanels.length; x++){
    fl.trace("Panel: " + fl.swfPanels[x].name + " -- Path: " + fl.swfPanels[x].path);
    }
}
```

fl.toggleBreakpoint()

適用版本

Flash Professional CS5。

用法

```
fl.toggleBreakPoint(String fileURI, int line, Boolean enable)
```

參數

fileURI 字串；切換中斷點的 AS 檔案 URI。

line 整數；切換中斷點的行號。

enable Boolean 值；如果設定為 true，則會啟用中斷點。如果設定為 false，則會停用中斷點。

說明
在指定的行切換指定 .as 檔案的中斷點。如果 enable 為 false，將會清除目前儲存在該行的中斷點。

範例
下列範例會在 AS 檔案 (位在 C:\AS\breakpointTest.as) 的第 10 行啟用中斷點：

```
fl.toggleBreakPoint("file:///C|/AS/breakpointTest.as", 10, 1);
```

fl.tools

適用版本
Flash MX 2004。

用法
```
fl.tools
```

說明
唯讀屬性：Tools 物件的陣列 (請參閱 Tools 物件)。此屬性僅在建立可擴充工具時使用。

fl.trace()

適用版本
Flash MX 2004。

用法
```
fl.trace(message)
```

參數
message 字串，顯示於 「輸出」面板中。

傳回值
無。

說明
方法：傳送以新的一行結束的文字字串到 「輸出」面板，如果仍不可見，便會顯示 「輸出」面板。此方法等同於 outputPanel.trace()，且運作方式與 ActionScript 中的 trace() 陳述式相同。

若要傳送空白行，請使用 fl.trace("") 或 fl.trace("\n")。您可以使用後行內命令，讓 \n 成為 message 字串的一部分。

範例
下列範例會在 「輸出」面板中顯示數行文字：

```
fl.outputPanel.clear();
fl.trace("Hello World!!!");
var myPet = "cat";
fl.trace("\nI have a " + myPet);
fl.trace("");
fl.trace("I love my " + myPet);
fl.trace("Do you have a " + myPet +"?");
```

fl.version

適用版本
Flash MX 2004。

用法
```
fl.version
```

說明
唯讀屬性：傳回 Flash 編寫工具的長字串版本 (包括平台)。

範例
下列範例會在「輸出」面板中顯示 Flash 編寫工具的版本：

```
alert(fl.version); // For example, WIN 10,0,0,540
```

fl.xmlui

適用版本
Flash MX 2004。

用法
```
fl.xmlui
```

說明
唯讀屬性：XMLUI 物件。此屬性讓您取得和設定 XMLUI 對話方塊中的 XMLUI 屬性，並讓您用程式設計的方式接受或取消對話方塊。

範例
請參閱 XMLUI 物件。

第 18 章 FLfile 物件

適用版本

Flash MX 2004 7.2。

說明

FLfile 物件可以讓您編寫 Flash 擴充功能，以存取、修改和移除本機檔案系統中的檔案和資料夾。FLfile API 會以 JavaScript API 擴充功能的形式提供。這個擴充功能稱為「共享元件庫」，位於下列資料夾中：

- Windows Vista：

  < 開機磁碟 >\Users\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS3\< 語言 >\Configuration\External Libraries\FLfile.dll

- Windows XP：

  < 開機磁碟 >\Documents and Settings\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS3\< 語言 >\Configuration\External Libraries\FLfile.dll

- Mac OS X：

  Macintosh HD/Users/< 使用者名稱 >/Library/Application Support/Adobe/Flash CS3/< 語言 >/Configuration/External Libraries/FLfile.dll

備註：請不要將 Flash 文件中包含元件的共享元件庫，與 JavaScript API 共享元件庫混淆。這兩者完全不同。

FLfile 方法會與磁碟上的檔案或資料夾 (目錄) 搭配使用。因此，每一個方法都會使用一個或多個參數來指定檔案或資料夾的位置。檔案或資料夾的位置是以字串表示，字串的形式與網站的 URL 非常類似，稱為「檔案 URI」(統一資源識別項)，格式如下 (包含引號)：

```
"file:///drive|/folder 1/folder 2/.../filename"
```

舉例來說，假如您要在 C 磁碟機中，建立一個名為 config 的資料夾，並將這個資料夾放置在 Program Files/MyApp 資料夾中，請使用下列命令：

```
FLfile.createFolder("file:///C|/Program Files/MyApp/config");
```

如果接下來要將名為 config.ini 的檔案放置在這個資料夾中，請使用下列命令：

```
FLfile.write("file:///C|/Program Files/MyApp/config/config.ini", "");
```

若要在 Macintosh 上建立資料夾，請使用下列命令：

```
FLfile.createFolder("file:///Macintosh/MyApp/config");
```

方法摘要

FLfile 物件可以搭配下列方法使用：

方法	說明
FLfile.copy()	複製檔案。
FLfile.createFolder()	建立一個或多個資料夾。
FLfile.exists()	判斷檔案或資料夾的存在。
FLfile.getAttributes()	得知檔案是否為可寫入、唯讀、隱藏、可見的，或是系統資料夾。
FLfile.getCreationDate()	指定在 1970 年 1 月 1 日和建立檔案或資料夾的時間之間所經過的秒數。

方法	說明
FLfile.getCreationDateObj()	取得檔案或資料夾的建立日期。
FLfile.getModificationDate()	指定在 1970 年 1 月 1 日和最後修改檔案或資料夾的時間之間所經過的秒數。
FLfile.getModificationDateObj()	取得檔案或資料夾的最後修改日期。
FLfile.getSize()	取得檔案的大小。
FLfile.listFolder()	列出資料夾的內容。
FLfile.platformPathToURI()	使用平台專用格式將檔案名稱轉換為 file:/// URI。
FLfile.read()	讀取檔案的內容。
FLfile.remove()	刪除檔案或資料夾。
FLfile.setAttributes()	將檔案或資料夾設定成唯讀、可寫入、隱藏或可見的。
FLfile.uriToPlatformPath()	將表示為 file:/// URI 形式的檔案名稱轉換為平台專用格式。
FLfile.write()	建立、寫入或附加至檔案。

FLfile.copy()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.copy(fileURI, copyURI)
```

參數
fileURI 字串；以 file:///URI 形式來表示，指定要複製的檔案。

copyURI 字串；以 file:///URI 形式來表示，指定被複製檔案的位置和名稱。

傳回值
如果成功的話，會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；將檔案從一個位置複製到另一個位置。如果 copyURI 已經存在，則這個方法會傳回 false。

範例
以下範例會建立一個組態檔 (名為 config.ini) 的備份，並且用新的名稱，將這個備份放置在原來的資料夾中：

```
var originalFileURI="file:///C|/Program Files/MyApp/config.ini";
var newFileURI="file:///C|/Program Files/MyApp/config_backup.ini";
FLfile.copy(originalFileURI, newFileURI);
```

您可以使用單一命令來執行相同的工作：

```
FLfile.copy("file:///C|:/Program Files/MyApp/config.ini", file:///C|/Program
Files/MyApp/config_backup.ini");
```

FLfile.createFolder()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.createFolder(folderURI)
```

參數
folderURI 資料夾的 URI，指定您要建立的資料夾結構。

傳回值
如果成功的話，會傳回 Boolean 值 true；如果 folderURI 已存在，則傳回 false。

說明
方法；在指定的位置建立一或多個資料夾。

您可以一次建立多個資料夾。例如，下列命令會同時建立 MyData 和 TempData 資料夾 (如果尚未存在)：

```
FLfile.createFolder("file:///c|/MyData/TempData")
```

範例
下列範例會在 Configuration 資料夾 (fl.configURI) 下建立一個資料夾和一個子資料夾：

```
fl.trace(FLfile.createFolder(fl.configURI+"folder01/subfolder01"));
```

下列範例會嘗試在 C 磁碟機根階層，建立名為 tempFolder 的資料夾，並顯示警告方塊，指出這個作業是否成功：

```
var folderURI = "file:///c|/tempFolder";
if (FLfile.createFolder(folderURI)) {
    alert("Created " + folderURI);
}
else {
    alert(folderURI + " already exists");
}
```

請參閱
FLfile.remove(), FLfile.write()

FLfile.exists()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.exists(fileURI)
```

參數
fileURI 字串；以 file:///URI 形式來表示，指定要驗證的檔案。

傳回值
如果成功的話，會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；用來判斷指定檔案是否存在。如果您指定資料夾和檔案名稱，此資料夾必須已經存在。若要建立資料夾，請參閱 FLfile.createFolder()。

範例
以下範例會檢查在暫存資料夾中名為 mydata.txt 的檔案，並顯示警告方塊，指出這個檔案是否存在：

```
var fileURI = "file:///c|/temp/mydata.txt";
if (FLfile.exists(fileURI)) {
    alert( fileURI + " exists.");
}
else {
    alert( fileURI + " does not exist.");
}
```

以下範例會檢查必要的組態檔是否存在於 MyApplication 資料夾。如果這個檔案不存在，便建立此檔案。

```
var configFile = "file:///C|/MyApplication/config.ini";
if (!FLfile.exists(configFile)) {
    FLfile.write(configFile,"");
}
```

請參閱
FLfile.write()

FLfile.getAttributes()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.getAttributes(fileOrFolderURI)
```

參數
fileOrFolderURI 字串；以 file:///URI 形式來表示，指定要擷取其特質的檔案或資料夾。

傳回值
字串，代表指定檔案或資料夾的特質。

如果這個檔案或資料夾不存在，結果便無法預料。使用這個方法之前，您必須使用 FLfile.exists()。

說明
方法；傳回代表指定檔案或資料夾特質的字串，如果檔案沒有特定的特質 (即並非唯讀、隱藏等等)，則傳回空字串。在使用這個方法前，請一律使用 FLfile.exists() 測試檔案或資料夾是否存在。

字串中的字元代表的特質如下：

- R — fileOrFolderURI 是唯讀的
- D — fileOrFolderURI 是資料夾 (目錄)

- H — fileOrFolderURI 是隱藏的 (僅適用於 Windows)
- S — fileOrFolderURI 是系統檔或資料夾 (僅適用於 Windows)
- A — fileOrFolderURI 已經就緒，可以封存 (僅適用於 Windows)

例如，如果 fileOrFolderURI 是隱藏的資料夾，則傳回的字串會是 "DH"。

範例
以下範例會取得 mydata.txt 檔的特質，而且如果這個檔案是唯讀的，則顯示警告方塊。

```
var URI = "file:///c|/temp/mydata.txt";
if (FLfile.exists(URI)){
    var attr = FLfile.getAttributes(URI);
        if (attr && (attr.indexOf("R") != -1)) { // Returned string contains R.
            alert(URI + " is read only!");
    }
}
```

請參閱
FLfile.setAttributes()

FLfile.getCreationDate()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.getCreationDate(fileOrFolderURI)
```

參數
fileOrFolderURI 字串，以 file:///URI 形式來表示，指定要以十六進位字串為形式擷取其建立日期和時間的檔案或資料夾。

傳回值
包含十六進位數字的字串，代表在 1970 年 1 月 1 日和建立檔案或資料夾的日期之間經過的秒數；如果這個檔案或資料夾不存在，則為 "00000000"。

說明
方法；指定在 1970 年 1 月 1 日和建立檔案或資料夾的時間之間所經過的秒數。這個方法主要是用來比較檔案或資料夾的建立或修改日期。

範例
以下範例會判斷檔案建立之後是否曾修改：

```
// Make sure the specified file exists
var fileURI = "file:///C|/MyApplication/MyApp.fla";
var creationTime = FLfile.getCreationDate(fileURI);
var modificationTime = FLfile.getModificationDate(fileURI);
if ( modificationTime > creationTime ) {
    alert("The file has been modified since it was created.");
}
else {
    alert("The file has not been modified since it was created.");
}
```

請參閱

FLfile.getCreationDateObj(), FLfile.getModificationDate()

FLfile.getCreationDateObj()

適用版本

Flash MX 2004 7.2。

用法

```
FLfile.getCreationDateObj(fileOrFolderURI)
```

參數

fileOrFolderURI 字串，以 file:///URI 形式來表示，指定要以 JavaScript Date 為形式擷取其建立日期和時間的檔案或資料夾。

傳回值

JavaScript Date 物件，代表建立指定檔案或資料夾的日期和時間。如果這個檔案不存在，則物件所包含的資訊會指出檔案或資料夾是在 1969 年 12 月 31 日 00:00 格林威治標準時間建立。

說明

方法；傳回 JavaScript Date 物件，代表建立指定檔案或資料夾的日期和時間。

範例

下列範例會在「輸出」面板中顯示建立檔案的日期 (以人們可讀取的形式)。

```
// Make sure the specified file exists.
var file1Date = FLfile.getCreationDateObj("file:///c|/temp/file1.txt");
fl.trace(file1Date);
```

請參閱

FLfile.getCreationDate(), FLfile.getModificationDateObj()

FLfile.getModificationDate()

適用版本

Flash MX 2004 7.2。

用法

```
FLfile.getModificationDate(fileOrFolderURI)
```

參數

fileOrFolderURI 字串，以 file:///URI 形式來表示，指定要以十六進位字串為形式擷取其修改日期和時間的檔案。

傳回值

包含十六進位數字的字串，代表在 1970 年 1 月 1 日和最後修改檔案或資料夾的時間之間經過的秒數；如果這個檔案不存在，則為 "00000000"。

說明

方法；指定在 1970 年 1 月 1 日和最後修改檔案或資料夾的時間之間經過的秒數。這個方法主要是用來比較檔案或資料夾的建立或修改日期。

範例

下列範例會比較兩個檔案的修改日期，並判斷這兩個檔案之中，何者是最後修改的檔案：

```
// Make sure the specified files exist.
file1 = "file:///C|/MyApplication/MyApp.fla";
file2 = "file:///C|/MyApplication/MyApp.as";
modificationTime1 = FLfile.getModificationDate(file1);
modificationTime2 = FLfile.getModificationDate(file2) ;
if(modificationTime1 > modificationTime2) {
    alert("File 2 is older than File 1") ;
}
else if(modificationTime1 < modificationTime2) {
    alert("File 1 is older than File 2") ;
}
else {
    alert("File 1 and File 2 were saved at the same time") ;
}
```

請參閱

FLfile.getCreationDate(), FLfile.getModificationDateObj()

FLfile.getModificationDateObj()

適用版本

Flash MX 2004 7.2。

用法

```
FLfile.getModificationDateObj(fileOrFolderURI)
```

參數

fileOrFolderURI 字串，以 file:///URI 形式來表示，指定要以 JavaScript Date 物件為形式擷取其修改日期和時間的檔案或資料夾。

傳回值

JavaScript Date 物件，代表最後修改指定檔案或資料夾的日期和時間。如果這個檔案或資料夾不存在，則物件所包含的資訊會指出檔案或資料夾是在 1969 年 12 月 31 日 00:00 格林威治標準時間建立。

說明
方法：傳回 JavaScript Date 物件，代表最後修改指定檔案或資料夾的日期和時間。

範例
下列範例會在「輸出」面板中顯示最後修改檔案的日期 (以人們可理解的形式)。

```
// Make sure the specified file exists.
var file1Date = FLfile.getModificationDateObj("file:///c|/temp/file1.txt");
trace(file1Date);
```

請參閱
FLfile.getCreationDateObj(), FLfile.getModificationDate()

FLfile.getSize()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.getSize(fileURI)
```

參數
fileURI 字串：以 file:///URI 形式來表示，指定要擷取其大小的檔案。

傳回值
整數，代表指定檔案大小的整數，並以位元組為單位。如果這個檔案不存在的話，會傳回 0。

說明
方法：傳回代表指定檔案大小的整數，並以位元組為單位。如果這個檔案不存在，會傳回 0。如果傳回值是 0，則可使用 FLfile.exists() 判斷檔案是 0 位元組的檔案，還是檔案並不存在。

只有當檔案的大小小於或等於 2GB 時，這個方法才會傳回正確的檔案大小值。

範例
下列範例會將 mydata.txt 檔案的大小儲存在 fileSize 變數中：

```
var URL = "file:///c|/temp/mydata.txt";
var fileSize = FLfile.getSize(URL);
```

FLfile.listFolder()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.listFolder(folderURI [, filesOrDirectories])
```

參數
folderURI 字串，以 file:///URI 形式來表示，指定要擷取其內容的資料夾。您可以在 folderURI 中使用萬用字元遮色片。有效的萬用字元為 * (代表一個或多個字元) 和 ? (代表單一字元)。

filesOrDirectories 選擇性字串，指定只傳回檔名，或只傳回資料夾 (目錄) 名稱。如果省略它，會同時傳回檔名和資料夾名稱。可接受的值為 "files" 和 "directories"。

傳回值
字串陣列，代表資料夾的內容。如果資料夾不存在，或是沒有檔案或資料夾符合指定的準則，則傳回空陣列。

說明
方法；傳回代表資料夾內容的字串陣列。

範例
下列範例會傳回 3 個陣列。第一個陣列代表 C:\temp 資料夾中的所有檔案；第二個代表 C:\temp 資料夾中的所有資料夾；第三個則代表 C:\temp 資料夾中的檔案和資料夾：

```
var fileURI = "file:///C|/temp/" ;
var folderURI = "file:///C|/temp" ;
var fileList1 = FLfile.listFolder(fileURI, "files"); // files
var fileList2 = FLfile.listFolder(folderURI, "directories"); //folders
var fileList3 = FLfile.listFolder(folderURI); //files and folders
fl.trace("Files: " + fileList1);
fl.trace("");
fl.trace("Folders:  " + fileList2);
fl.trace("");
fl.trace("Files and folders: " + fileList3);
```

以下範例會傳回暫存資料夾中所有文字檔 (.txt) 的陣列，並在警告方塊中顯示清單：

```
var folderURI = "file:///c|/temp";
var fileMask = "*.txt";
var list = FLfile.listFolder(folderURI + "/" + fileMask, "files");
if (list) {
    alert(folderURI + " contains: " + list.join(" "));
}
```

以下範例會在指定的 folderURI 中使用檔案遮色片，以傳回 Windows 應用程式資料夾中的所有可執行檔名稱：

```
var executables = FLfile.listFolder("file:///C|/WINDOWS/*.exe","files");
alert(executables.join("\n"));
```

FLfile.platformPathToURI()

適用版本
Flash CS4 Professional。

用法
```
FLfile.platformPathToURI(fileName)
```

參數
fileName 字串，使用平台專用格式指定您要轉換的檔案名稱。

傳回值
字串 (以 file:///URI 形式來表示)。

說明
方法；將平台專用格式的檔案名稱轉換為 file:/// URI。

範例
下列範例會將檔案名稱由平台專用格式轉換為 file:/// URI，並將後者傳遞給 outputPanel.save()：

```
var myFilename = "C:\\outputPanel.txt";
var myURI=FLfile.platformPathToURI(myFilename);
fl.outputPanel.save(myURI);
```

請參閱
FLfile.uriToPlatformPath()

FLfile.read()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.read()
```

參數
fileOrFolderURI 字串；以 file:///URI 形式來表示，指定要擷取其特質的檔案或資料夾。

傳回值
以字串形式指定的內容。如果讀取失敗，會傳回 null。

說明
方法；傳回字串形式的指定檔案內容。如果讀取失敗，會傳回 null。

範例
以下範例會讀取 mydata.txt 檔；如果讀取成功，會顯示警告方塊，其中包含這個檔案的內容。

```
var fileURI = "file:///c|/temp/mydata.txt";
var str = FLfile.read( fileURI);
if (str) {
    alert( fileURL + " contains: " + str);
}
```

以下範例會從類別檔案中讀取 ActionScript 程式碼，並將其儲存在 code 變數中：

```
var classFileURI = "file:///C|/MyApplication/TextCarousel.as";
var code = FLfile.read(classFileURI);
```

FLfile.remove()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.remove(fileOrFolderURI)
```

參數
fileOrFolderURI 字串；以 file:///URI 形式來表示，指定要移除 (刪除) 的檔案或資料夾。

傳回值
如果成功的話，會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；刪除指定的檔案或資料夾。如果資料夾中包含檔案，這些檔案也會一併刪除。帶有 R (唯讀) 特質的檔案無法移除。

範例
如果檔案已存在，以下範例就會警告使用者；如果使用者選擇刪除檔案，便刪除檔案：

```
var fileURI = prompt ("Enter file/folder to be deleted: ", "file:///c|/temp/delete.txt");
if (FLfile.exists(fileURI)) {
    var confirm = prompt("File exists. Delete it? (y/n)", "y");
    if (confirm == "y" || confirm == "Y") {
        if(FLfile.remove(fileURI)) {
            alert(fileURI + " is deleted.");
        }
        else {
            alert("fail to delete " + fileURI);
        }
    }
}
else {
    alert(fileURI + " does not exist");
}
```

以下範例會刪除應用程式所建立的組態檔：

```
if(FLfile.remove("file:///C|/MyApplication/config.ini")) {
    alert("Configuration file deleted");
}
```

以下範例會刪除 Configuration 資料夾及其內容：

```
FLfile.remove("file:///C|/MyApplication/Configuration/");
```

請參閱
FLfile.createFolder(), FLfile.getAttributes()

FLfile.setAttributes()

適用版本
Flash MX 2004 7.2。

用法

```
FLfile.setAttributes(fileURI, strAttrs)
```

參數

fileURI 字串，以 file:///URI 形式來表示，指定您要設定其特質的檔案。

strAttrs 字串，指定要設定特質的值。如需 strAttrs 的可接受值，請參閱下面的 「說明」一節。

傳回值

如果成功的話，會傳回 Boolean 值 true。

備註：如果這個檔案或資料夾不存在，結果便無法預料。使用這個方法之前，您必須使用 FLfile.exists()。

說明

方法：為指定的檔案指定系統階層特質。

下列為 strAttrs 的有效值：

- N - 無特定特質 (並非唯讀、隱藏等等)
- A - 已經就緒，可以封存 (僅適用於 Windows)
- R - 唯讀 (在 Macintosh 中，唯讀代表 「鎖定」)
- W - 可寫入 (會覆蓋 R)
- H - 隱藏 (僅適用於 Windows)
- V - 可見的 (會覆蓋 H，僅適用於 Windows)

如果您在 strAttrs 中同時包含 R 和 W，則會忽略 R，且將檔案設定為可寫入。同樣地，如果您傳送了 H 和 V，則會忽略 H，並將檔案設定為可見。

如果想要確定並未設定封存特質，請在設定特質之前，搭配 N 參數使用這個命令。也就是說，A 沒有直接相對應的值可以關閉封存特質。

範例

以下範例會將 mydata.txt 檔設定為唯讀的隱藏檔。這對封存特質沒有影響。

```
var URI = "file:///c|/temp/mydata.txt";
if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "RH");
}
```

以下範例會將 mydata.txt 檔設定為唯讀的隱藏檔。並確保不會設定封存特質。

```
var URI = "file:///c|/temp/mydata.txt";

if (FLfile.exists(URI)) {
    FLfile.setAttributes(URI, "N");
    FLfile.setAttributes(URI, "RH");
}
```

請參閱

FLfile.getAttributes()

FLfile.uriToPlatformPath()

適用版本
Flash CS4 Professional。

用法
```
FLfile.uriToPlatformPath(fileURI)
```

參數
fileURI 字串；以 file:///URI 形式來表示，指定要轉換的檔案名稱。

傳回值
字串，代表平台專用路徑。

說明
方法：將以 file:///URI 形式表示的檔案名稱轉換為平台專用格式。

範例
下列範例會將 file:/// URI 轉換為平台專用格式：

```
var dir =(fl.configDirectory);
var URI = FLfile.platformPathToURI(dir);
fl.trace(URI == fl.configURI); // displays "true"
```

請參閱
FLfile.platformPathToURI()

FLfile.write()

適用版本
Flash MX 2004 7.2。

用法
```
FLfile.write(fileURI, textToWrite, [ , strAppendMode])
```

參數
fileURI 字串；以 file:///URI 形式來表示，指定要寫入的檔案。

textToWrite 字串，代表要放置在檔案中的文字。

strAppendMode 選擇性字串，其中帶有 "append" 值，用來指定將 textToWrite 附加到現有的檔案中。如果忽略的話，會使用 textToWrite 來覆寫 fileURI。

傳回值
如果成功的話，會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；將指定的字串寫入指定的檔案 (以 UTF-8 編碼的方式)。如果指定的檔案不存在，便建立檔案。不過，要放置檔案的資料夾必須存在，才能使用這個方法。若要建立資料夾，請使用 FLfile.createFolder()。

範例

以下範例會嘗試將字串 "xxx" 寫入 mydata.txt 檔，並於寫入成功時顯示警告訊息。然後嘗試將字串 "aaa" 附加到檔案中，並於寫入成功時顯示第二則警告訊息。在執行這個指令碼之後，mydata.txt 檔就只包含文字 "xxxaaa"。

```
var URI = "file:///c|/temp/mydata.txt";
if (FLfile.write(URI, "xxx")) {
    alert("Wrote xxx to " + URI);
}
if (FLfile.write(URI, "aaa", "append")) {
    alert("Appended aaa to " + fileURI);
}
```

請參閱

FLfile.createFolder(), FLfile.exists()

第 19 章 folderItem 物件

繼承 Item 物件 > folderItem 物件

適用版本
Flash MX 2004。

說明
folderItem 物件為 Item 物件的子類別。folderItem 沒有唯一的方法或屬性。請參閱 Item 物件。

第 20 章 fontItem 物件

繼承　Item 物件 > fontItem 物件

適用版本
Flash MX 2004。

說明
fontItem 物件為 Item 物件的子類別 (請參閱 Item 物件)。

屬性摘要
除了 Item 物件屬性，fontItem 物件還可以使用下列屬性：

屬性	說明
fontItem.bitmap	指定是否將字體項目設定為點陣圖。
fontItem.bold	指定字體項目是否為粗體。
fontItem.embeddedCharacters	指定要內嵌的字元。
fontItem.embedRanges	指定可在 「字體內嵌」對話方塊中選取的項目。
fontItem.embedVariantGlyphs	指定是否使用發佈 SWF 檔時的字體輸出變化字形。
fontItem.font	與字體項目關聯的裝置字體名稱。
fontItem.isDefineFont4Symbol	指定發佈 SWF 檔時輸出的字體格式。
fontItem.italic	指定字體項目是否為斜體。
fontItem.size	字體項目的大小，以點為單位。

fontItem.bitmap

適用版本
Flash CS4 Professional。

用法
```
fontItem.bitmap
```

說明
屬性：Boolean 值，指定是否將字體項目變成點陣圖，是為 true，否則為 false。

範例
假定元件庫中的第一個項目是字體項目，則下列程式碼會在字體變成點陣圖時於 「輸出」面板顯示 true，否則為 false：

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.trace("bitmap: "+ theItem.bitmap);
```

fontItem.bold

適用版本
Flash CS4 Professional。

用法
```
fontItem.bold
```

說明
屬性；Boolean 值，指定字體項目是否為粗體，是為 true，否則為 false。

範例
假定元件庫中的第一個項目是字體項目，則下列程式碼會在字體為粗體時於 「輸出」面板顯示 true，否則為 false，並接著將其設為粗體。

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("bold: "+ theItem.bold);
theItem.bold=true;
fl.trace("bold: "+ theItem.bold);
```

fontItem.embeddedCharacters

適用版本
Flash CS5 Professional。

用法
```
fontItem.embeddedCharacters
```

說明
屬性；可讓您指定要在 SWF 檔案中內嵌字元的字串值，這樣字元便不需要存在於最後播放 SWF 檔案的裝置上。這個屬性提供與 「字體內嵌」對話方塊相同的功能。

您也可以讀取這個屬性，以利為指定的 「字體」項目，找出透過 「字體內嵌」對話方塊所指定的字元。

範例
假設 「元件庫」中的第一個項目是 「字體」項目，則下列程式碼會內嵌字元 a、b 和 c。

```
fl.getDocumentDOM().library.items[0].embeddedCharacters = "abc";
```

fontItem.embedRanges

適用版本
Flash CS5 Professional。

用法
```
fontItem.embedRanges
```

說明
屬性：指定一系列分隔的整數字串值，這些整數與 「字體內嵌」對話方塊中可選取的項目相對應。

您也可以讀取這個屬性，以利為指定的 「字體」項目，找出透過 「字體內嵌」對話方塊所指定的字元。

備註：範圍數字與在組態資料夾中找到的 FontEmbedding/UnicodeTables.xml 檔案相對應。

範例
假設 「元件庫」中的第一個項目是 「字體」項目，則下列程式碼會內嵌由整數 1、3 和 7 所識別的範圍。

```
fl.getDocumentDOM().library.items[0].embedRanges = "1|3|7";
```

假設 「元件庫」中的第一個項目是 「字體」項目，則下列程式碼會重設要內嵌的範圍。

```
fl.getDocumentDOM().library.items[0].embedRanges = "";
```

fontItem.embedVariantGlyphs

適用版本
Flash CS4 Professional。

用法
```
fontItem.embedVariantGlyphs
```

說明
備註：雖然在 Flash CS5 Professional 有提供此屬性，但是將它套用至 Text Layout Framework (TLF) 文字時不會有任何作用。從 Flash Professional CS5 開始，變化字形總是會內嵌在 TLF 文字所使用的字體中。以下所參考的 flash.text.engine (FTE) 只有在 Flash Professional CS4 中才有提供。

屬性：Boolean 值，指定是 (true) 否 (false) 使用發佈 SWF 檔時的字體輸出變化字形。將此值設定為 true 會增加 SWF 檔的大小。預設值為 false。

有些語言會將字元文字動態替代為輸入的文字 (例如，泰文、阿拉伯文、希伯來文和希臘文)。如果您要使用這些語言類型來配置或輸入文字，請將此屬性設定為 true。

範例
與 flash.text API 相容的字體元件會在元件庫中出現，使用者可以直接管理這些元件。然而，與 flash.text.engine (FTE) API 相容的字體元件不會在元件庫中出現，因此必須手動加以管理。下列函數會將新字體加入至可與 FTE API 搭配使用的元件庫。

```
function embedFontSymbol(symbolName, fontName, includeVariants) {
    var doc = fl.getDocumentDOM();
    if (doc) {
        // look up the item. if it exists, delete it.
        var index = doc.library.findItemIndex(symbolName);
        if (index > -1)
            doc.library.deleteItem(symbolName);

        // make a new font symbol in the library
        doc.library.addNewItem('font', symbolName);

        // look up the symbol by its name
        var index = doc.library.findItemIndex(symbolName);
        if (index > -1) {
            // get the item from the library and set the attributes of interest
            var fontObj = doc.library.items[index];
            fontObj.isDefineFont4Symbol = true;
            fontObj.font = fontName;
            fontObj.bold = false;
            fontObj.italic = false;
            fontObj.embedVariantGlyphs = includeVariants;
            // this is what forces the font into the SWF stream
            fontObj.linkageExportForAS = true;
            fontObj.linkageExportInFirstFrame = true;
        }
    }
}
```

下列函數會在「輸出」面板中顯示所有字體元件。

```
function dumpFontSymbols()
{
    var doc = fl.getDocumentDOM();
    if (doc) {
        var items = doc.library.items;
        fl.trace("items length = " + items.length);
        var i;
        for(i=0; i<items.length; i++) {
            var item = items[i];
            fl.trace("itemType = " + item.itemType);
            if (item.itemType == 'font') {
                fl.trace("name = " + item.name);
                fl.trace("DF4 symbol = " + item.isDefineFont4Symbol);
                fl.trace("font = " + item.font);
            }
        }
    }
}
```

請參閱

fontItem.isDefineFont4Symbol、text.embedVariantGlyphs

fontItem.font

適用版本

Flash CS4 Professional。

用法

```
fontItem.font
```

說明

屬性：字串，指定與字體項目相關的裝置字體名稱。如果您輸入的字體並未對應至已安裝的裝置字體，就會顯示錯誤訊息。若要判斷字體是否存在於系統中，請使用 fl.isFontInstalled()。

備註：當您設定此值時，最後產生的屬性值可能會與您輸入的字串不同。請參閱下列範例。

範例

假定元件庫中的第一個項目是字體項目，則下列程式碼會顯示目前與字體項目相關的裝置字體名稱，然後將其變更為 Times 字體：

```
fl.outputPanel.clear();
var theItem = fl.getDocumentDOM().library.items[0];
fl.trace(theItem.font);
theItem.font = "Times";
// depending on your system, the following may display something like "Times-Roman"
fl.trace(theItem.font);
```

fontItem.isDefineFont4Symbol

適用版本

Flash CS4 Professional。

用法

```
fontItem.isDefineFont4Symbol
```

說明

屬性：Boolean 值，指定發佈 SWF 檔時輸出的字體格式。如果此值為 true，Flash 會輸出可與 flash.text.engine (FTE) API 搭配使用的字體。如果此值為 false，則此字體可與 flash.text API (包括文字欄位) 搭配使用。預設值為 false。

範例

請參閱 fontItem.embedVariantGlyphs。

fontItem.italic

適用版本

Flash CS4 Professional。

用法

```
fontItem.italic
```

說明

屬性：Boolean 值，指定字體項目是否為斜體，是為 true，否則為 false。

範例

假定元件庫中的第一個項目是字體項目，則下列程式碼會在字體為斜體時於「輸出」面板顯示 true，否則為 false，並接著將其設為斜體。

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("italic: "+ theItem.italic);
theItem.italic=true;
fl.trace("italic: "+ theItem.italic);
```

fontItem.size

適用版本

Flash CS4 Professional。

用法

```
fontItem.size
```

說明

屬性：整數，代表字體項目大小，並以點為單位。

範例

假設元件庫中的第一個項目是字體項目，則下列程式碼會在「輸出」面板中顯示項目的點大小，接著將其設為 24 點。

```
var theItem = fl.getDocumentDOM().library.items[0];
fl.outputPanel.clear();
fl.trace("font size: "+ theItem.size);
theItem.size=24;
fl.trace("font size: "+ theItem.size);
```

第 21 章 Frame 物件

適用版本

Flash MX 2004。

說明

Frame 物件代表圖層中的影格。

方法摘要

Frame 物件可搭配使用以下方法：

方法	說明
frame.convertMotionObjectTo2D()	將選取的移動物件轉換為 2D 移動物件。
frame.convertMotionObjectTo3D()	將選取的移動物件轉換為 3D 移動物件。
frame.getCustomEase()	傳回每個具有 x 和 y 屬性的 JavaScript 物件陣列。
frame.getMotionObjectXML()	從選取的移動物件傳回 Motion XML。
frame.hasMotionPath()	通知您目前的選取範圍是否有移動補間動畫。
frame.is3DMotionObject()	通知您目前的選取範圍是否為 3D 移動物件。
frame.isMotionObject()	通知您目前的選取範圍是否為移動物件。
frame.selectMotionPath()	選取或取消選取目前移動物件的移動路徑。
frame.setCustomEase()	指定當做自訂加 / 減速曲線使用的三次方貝茲曲線。
frame.setMotionObjectDuration()	指定目前所選移動物件的持續時間 (補間動畫範圍長度)。
frame.setMotionObjectXML()	將指定的 Motion XML 套用至選取的移動物件。

屬性摘要

Frame 物件可搭配使用以下屬性：

屬性	說明
frame.actionScript	字串，代表 ActionScript 程式碼。
frame.duration	唯讀：整數，代表影格序列中的影格數目。
frame.elements	唯讀：Element 物件的陣列 (請參閱 Element 物件)。
frame.hasCustomEase	Boolean 值，指定影格是否從自訂加 / 減速曲線取得其加 / 減速資訊。
frame.labelType	字串，指定影格名稱類型。
frame.motionTweenOrientToPath	Boolean 值，指定補間動畫元素是否要在沿路徑移動時旋轉元素，以維持與路徑上每個點相對的角度。
frame.motionTweenRotate	字串，指定補間動畫元素的旋轉方式。
frame.motionTweenRotateTimes	整數，指定起始關鍵影格和下一個關鍵影格之間，補間動畫元素的旋轉次數。
frame.motionTweenScale	Boolean 值，指定補間動畫元素是 (true) 否 (false) 要在之後的關鍵影格中縮放至物件的大小，並隨著補間動畫中的每一個影格增加大小。

屬性	說明
frame.motionTweenSnap	Boolean 值，指定補間動畫元素是 (true) 否 (false) 要自動貼齊與此影格圖層相關的移動導引圖層中的最近點。
frame.motionTweenSync	Boolean 值，如果設定為 true，會使補間動畫物件的動畫與主要時間軸同步。
frame.name	字串，指定影格的名稱。
frame.shapeTweenBlend	字串，指定在補間動畫開始時的關鍵影格形狀與之後的關鍵影格形狀的轉換效果。
frame.soundEffect	字串，對直接附加至影格的聲音 (frame.soundLibraryItem) 指定其特效。
frame.soundLibraryItem	用來建立聲音的元件庫項目 (請參閱 SoundItem 物件)。
frame.soundLoop	整數值，指定直接附加至影格 (frame.soundLibraryItem) 的聲音播放次數。
frame.soundLoopMode	字串，指定直接附加至影格的聲音 (frame.soundLibraryItem) 應該播放特定的次數，或無限重複。
frame.soundName	字串，指定直接附加至影格的聲音 (frame.soundLibraryItem) 名稱，這個名稱會與儲存在元件庫的名稱相同。
frame.soundSync	字串，對直接附加至影格的聲音 (frame.soundLibraryItem) 指定其同步化行為。
frame.startFrame	唯讀：序列中第一個影格的索引。
frame.tweenEasing	整數，指定要套用至補間動畫物件的加 / 減速量。
frame.tweenInstanceName	將實體名稱指定給指定的移動物件。
frame.tweenType	字串，指定補間動畫的類別。
frame.useSingleEaseCurve	Boolean 值，指定所有屬性的加 / 減速資訊是否都使用單一的自訂加 / 減速曲線。

frame.convertMotionObjectTo2D()

適用版本
Flash Professional CS5。

用法
```
frame.convertMotionObjectTo2D()
```

說明
方法；將選取的移動物件轉換為 2D 移動物件。

範例
下列範例會將選取的移動物件轉換為 2D 移動物件：

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && the()){
theFrame.convertMotionObjectTo2D();
}else{
fl.trace("It isn't motion or it's already a 2D motion");
}
```

frame.convertMotionObjectTo3D()

適用版本
Flash Professional CS5。

用法
```
frame.convertMotionObjectTo3D()
```

說明
方法：將選取的移動物件轉換為 3D 移動物件。

範例
下列範例會將選取的移動物件轉換為 3D 移動物件：

```
var doc = fl.getDocumentDOM();
v   a r my_tl = doc.getTimeline();
this.getCurrentF r ame = functi on() {
var layer = my_tl.layers[my   _tl.cu  rrentLa yer];
var frame = layer.frames[my_tl.currentFrame ];
retur n frame;}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && !theFrame.is3DMotionObject()){
theFrame.convertMotionObjectTo3D();
}else{
fl.trace("It isn't motion or it's already a 3D motion");
}
```

frame.actionScript

適用版本
Flash MX 2004。

用法
```
frame.actionScript
```

說明
屬性：字串，代表 ActionScript 程式碼。若要插入換行字元，請使用 "\n"。

範例
以下範例會將 stop() 指定給第一個影格的最上層圖層動作：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].actionScript = 'stop();';
```

frame.duration

適用版本
Flash MX 2004。

用法
```
frame.duration
```

說明
唯讀屬性：整數，代表影格序列中的影格數目。

範例
以下範例會將影格序列 (從最上層圖層的第一個影格開始) 中的影格數目儲存在 frameSpan 變數中：

```
var frameSpan = fl.getDocumentDOM().getTimeline().layers[0].frames[0].duration;
```

frame.elements

適用版本
Flash MX 2004。

用法
```
frame.elements
```

說明
唯讀屬性：Element 物件的陣列 (請參閱 Element 物件)。元素的順序就是元素儲存在 FLA 檔中的順序。如果 「舞台」上有多個形狀且都未分組，則 Flash 會將它們視為一個元素。如果每個形狀都已分組，則 「舞台」上會有多個群組，Flash 會將它們視為不同的元素。換句話說，Flash 會將新的且未分組的形狀視為單一元素，無論 「舞台」上有多少不同的形狀都是一樣。舉例來說，如果影格包含 3 個新的且未分組的形狀，則該影格中的 elements.length 會傳回值 1。若要解決這個問題，請個別選取每個形狀並加以分組。

範例
下列範例會將最上層圖層第一個影格的目前元素陣列儲存在 myElements 變數中：

```
var myElements = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements;
```

frame.getCustomEase()

適用版本
Flash 8。

用法
```
Frame.getCustomEase([property])
```

參數
property 選擇性字串，指定您要傳回自訂加 / 減速值的屬性。可接受的值為 "all"、"position"、"rotation"、"scale"、"color" 和 "filters"。預設值為 "all"。

傳回值
傳回每個具有 x 和 y 屬性的 JavaScript 物件陣列。

說明
方法：傳回物件的陣列，代表定義加 / 減速曲線的三次方貝茲曲線的控制點。

範例
以下範例會傳回最上層圖層中，第一個影格 position 屬性的自訂加 / 減速值：

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
var easeArray = theFrame.getCustomEase("position");
```

請參閱
frame.hasCustomEase, frame.setCustomEase(), frame.useSingleEaseCurve

frame.getMotionObjectXML()

適用版本
Flash Professional CS5。

用法
```
Frame.getMotionObjectXML()
```

說明
從選取的移動物件傳回 Motion XML 的字串。

範例
下列範例會從選取的移動物件傳回 Motion XML。

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
var layer = my_tl.layers[my _tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()) {
//fl.trace(theFrame.getMotionObjectXML());
}else{
fl.trace("It is not motion.");
}
```

frame.hasCustomEase

適用版本
Flash 8。

用法
```
frame.hasCustomEase
```

說明
屬性；Boolean 值。如果是 true，影格會從自訂加 / 減速曲線中，取得其加 / 減速資訊。如果是 false，影格會從加 / 減速值中，取得其加 / 減速資訊。

範例
以下範例會指定最上層圖層的第一個影格，一定要從加 / 減速值中取得影格的加 / 減速資訊，而不要從自訂的加 / 減速曲線中取得：

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
theFrame.hasCustomEase = false;
```

請參閱
frame.getCustomEase(), frame.setCustomEase(), frame.useSingleEaseCurve

frame.hasMotionPath()

適用版本
Flash Professional CS5。

用法
```
Frame.hasMotionPath()
```

說明
方法；Boolean 值。讓您知道目前的選取範圍是否包括移動路徑。

範例
下列範例會傳回 trace 陳述式，以通知您目前的選取範圍是否具有移動路徑。

```
var doc    = fl.getDocumentDOM();
var  my_tl = doc.getTimeline() ;
t his .getCurrentFrame = function(){
var layer =  my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()){
if (theFrame.hasMotionPath()){
fl.trace("There is a motion path");
}else{
fl.trace("There is no motion path");
}
```

frame.is3DMotionObject()

適用版本
Flash Professional CS5。

用法
```
Frame.is3DMotionObject()
```

說明
方法：Boolean 值。讓您知道目前的選取範圍是否為 3D 移動物件。

範例
下列範例會傳回 trace 陳述式，以通知您目前的選取範圍是否為 3D 移動物件。

```
var doc = fl.getDocumentDOM();
va     r  my_tl = doc.getTimeline();
this.getCurrentFr ame = func t i o n(){
var layer = my_tl.layers[my_  t  l. c  u   r re ntL  aye     r];
var frame = layer .frame s[my_t l.curr entFrame] ;
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject() && theFrame.is3DMotionObject()){
fl.trace("This selection is 3D Motion");
}else{
fl.trace("This selection is not 3D motion");
}
```

frame.isMotionObject()

適用版本
Flash Professional CS5。

用法
```
Frame.isMotionObject()
```

說明
方法：Boolean 值。讓您知道目前的選取範圍是否為移動物件。

範例
下列範例會傳回 trace 陳述式，以通知您目前的選取範圍是否為移動物件。

```
var my_tl = doc.getTimeline()      ;
this.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl.currentLayer];<
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()) {
fl.trace("This selection is motion.");
}else{
fl.trace("This selection is not motion.");
}
```

frame.labelType

適用版本
Flash MX 2004。

用法
`frame.labelType`

說明
屬性；字串，指定影格的名稱類型。可接受的值為 "none"、"name"、"comment"，和 "anchor"。將標籤設定為 "none" 會清除 frame.name 屬性。

範例
以下範例會將最上層圖層的第一個影格名稱，設定為 "First Frame"，並且將其標籤設定為 "comment"：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';
fl.getDocumentDOM().getTimeline().layers[0].frames[0].labelType = 'comment';
```

frame.motionTweenOrientToPath

適用版本
Flash MX 2004。

用法
`frame.motionTweenOrientToPath`

說明
屬性；Boolean 值，指定補間動畫元素是 (true) 否 (false) 在沿著路徑移動時旋轉元素，與路徑上的每一個點都維持一定的角度。

如果您要指定這個屬性的值，您必須將 frame.motionTweenRotate 設定為 "none"。

frame.motionTweenRotate

適用版本
Flash MX 2004。

用法
```
frame.motionTweenRotate
```

說明
屬性；字串，指定補間動畫元素的旋轉方式。可接受的值為 "none"、"auto"、"clockwise" 和 "counter-clockwise"。值 "auto" 表示物件會朝著移動需求最小的方向旋轉，以符合之後關鍵影格中的物件旋轉。

如果您要指定 frame.motionTweenOrientToPath 的值，請將這個屬性設定為 "none"。

範例
請參閱 frame.motionTweenRotateTimes。

frame.motionTweenRotateTimes

適用版本
Flash MX 2004。

用法
```
frame.motionTweenRotateTimes
```

說明
屬性；整數，指定起始關鍵影格和下一個關鍵影格之間，補間動畫元素的旋轉次數。

範例
以下範例會將這個影格中的元素，在到達下一個關鍵影格之前，依反時針方向旋轉三次：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotate = "counter-clockwise";
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenRotateTimes = 3;
```

frame.motionTweenScale

適用版本
Flash MX 2004。

用法
```
frame.motionTweenScale
```

說明
屬性；Boolean 值，指定補間動畫元素是 (true) 否 (false) 要在之後的關鍵影格中縮放至物件的大小，並隨著補間動畫中的每一個影格增加大小。

範例
下列範例指定補間動畫應縮放成下列關鍵影格中的物件大小，並且隨著補間動畫中的每個影格增加大小。

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenScale = true;
```

frame.motionTweenSnap

適用版本
Flash MX 2004。

用法
```
frame.motionTweenSnap
```

說明
屬性；Boolean 值，指定補間動畫元素是 (true) 否 (false) 要自動貼齊與此影格圖層相關的移動導引圖層中的最近點。

frame.motionTweenSync

適用版本
Flash MX 2004。

用法
```
frame.motionTweenSync
```

說明
屬性；Boolean 值，如果設定為 true，會使補間動畫物件的動畫與主要時間軸同步。

範例
以下範例會指定補間動畫物件須與時間軸同步化：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].motionTweenSync = true;
```

frame.name

適用版本
Flash MX 2004。

用法
```
frame.name
```

說明
屬性；字串，指定影格的名稱。

範例
以下範例會將最上層圖層第一個影格的名稱，設定為 "First Frame"，並將 name 值儲存在 frameLabel 變數中：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].name = 'First Frame';
var frameLabel = fl.getDocumentDOM().getTimeline().layers[0].frames[0].name;
```

frame.selectMotionPath()

適用版本
Flash Professional CS5。

用法
```
Frame.selectMotionPath()
```

說明
方法：Boolean 值。選取 (true) 或取消選取 (false) 目前移動物件的移動路徑。

範例
此範例會選取或取消選取目前移動物件的移動路徑。

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
t   his.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl. c u rrentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()){
if (theFrame.hasMotionPath()){
theFrame.selectMotionPath(true);
}
else{
fl.trace("There is no motion path");
}
}else{
fl.trace("It is no motion");
}
```

frame.setCustomEase()

適用版本
Flash 8。

用法
```
frame.setCustomEase(property, easeCurve)
```

參數
property 字串，指定應該使用加 / 減速曲線的屬性。可接受的值為 "all"、"position"、"rotation"、"scale"、"color" 和 "filters"。

easeCurve 物件陣列，定義加 / 減速曲線。每個陣列元素都必須是帶有 x 和 y 屬性的 JavaScript 物件。

傳回值
無。

說明

方法：指定控制點和正切端點座標的陣列，說明要當做自訂加 / 減速曲線使用的三次方貝茲曲線。這個陣列由控制點和正切端點的水平 (順序：由左到右) 位置來建構。

範例

下列範例會針對第一個圖層上第一個影格的所有屬性，將加 / 減速曲線設定為 easeCurve 陣列所指定的貝茲曲線：

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0];
var easeCurve = [ {x:0,y:0}, {x:.3,y:.3}, {x:.7,y:.7}, {x:1,y:1} ];
theFrame.setCustomEase( "all", easeCurve );
```

請參閱

frame.getCustomEase(), frame.hasCustomEase, frame.useSingleEaseCurve

frame.setMotionObjectDuration()

適用版本

Flash Professional CS5。

用法

```
Frame.setMotionObjectDuration( duration [, stretchExistingKeyframes] )
```

參數

duration 指定所選移動物件之補間動畫範圍的影格數目。

stretchExistingKeyframes Boolean 值，判斷補間動畫範圍是否延伸或影格是否新增至最後一個影格的結尾。

說明

方法：設定目前所選移動物件的持續時間 (補間動畫範圍長度)。

範例

下列範例為所選移動物件指定 11 個影格的持續時間。

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
if(theFrame.isMotionObject()){
theFrame.setMotionObjectDuration(11);
}else{
fl.trace("It isn't motion");
}
```

frame.setMotionObjectXML()

適用版本
Flash Professional CS5。

用法
```
Frame.setMotionObjectXML( xmlstr [, endAtCurrentLocation] )
```

參數
xmlstr 指定 XML 字串的字串值。

endAtCurrentLocation 判斷補間動畫是否開始或結束於目前位置的 Boolean 值。

說明
方法：將指定的 Motion XML 套用至選取的移動物件。

範例
此範例指定將識別為 myMotionXML 的 Motion XML 套用至選取的移動物件。

```
var doc = fl.getDocumentDOM();
var my_tl = doc.getTimeline();
this.getCurrentFrame = function(){
var layer = my_tl.layers[my_tl.currentLayer];
var frame = layer.frames[my_tl.currentFrame];
return frame;
}
var theFrame = getCurrentFrame();
theFrame.setMotionObjectXML(myMotionXML.toString(), false);
```

frame.shapeTweenBlend

適用版本
Flash MX 2004。

用法
```
frame.shapeTweenBlend
```

說明
屬性；字串，指定在補間動畫開始時的關鍵影格形狀與之後的關鍵影格形狀的轉換效果。可接受的值為 "distributive" 和 "angular"。

frame.soundEffect

適用版本
Flash MX 2004。

用法
```
frame.soundEffect
```

說明
屬性；字串，對直接附加至影格的聲音 (frame.soundLibraryItem) 指定其特效。可接受的值為 "none"、"left channel"、"right channel"、"fade left to right"、"fade right to left"、"fade in"、"fade out" 和 "custom"。

範例
以下範例會指定附加至第一個影格的聲音應該要淡入：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundEffect = "fade in";
```

frame.soundLibraryItem

適用版本
Flash MX 2004。

用法
```
frame.soundLibraryItem
```

說明
屬性；用來建立聲音的元件庫項目 (請參閱 SoundItem 物件)。聲音會直接附加至影格。

範例
下列範例將元件庫中的第一個項目指定給第一個影格的 soundLibraryItem 屬性：

```
// The first item in the library must be a sound object.
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLibraryItem
=fl.getDocumentDOM().library.items[0];
```

frame.soundLoop

適用版本
Flash MX 2004。

用法
```
frame.soundLoop
```

說明
屬性；整數值，用來指定直接附加至影格的聲音 (frame.soundLibraryItem) 播放次數。如果您要指定這個屬性的值，請將 frame.soundLoopMode 設定為 "repeat"。

範例
請參閱 frame.soundLoopMode。

frame.soundLoopMode

適用版本
Flash MX 2004。

用法
```
frame.soundLoopMode
```

說明
屬性；字串，用來指定直接附加至影格的聲音 (frame.soundLibraryItem) 是否應該播放特定的次數或無限重複。可接受的值為 "repeat" 和 "loop"。如果要指定聲音應該播放的次數，請設定 frame.soundLoop 的值。

範例
以下範例會指定聲音應該播放兩次：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoopMode = "repeat";
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundLoop = 2;
```

frame.soundName

適用版本
Flash MX 2004。

用法
```
frame.soundName
```

說明
屬性；字串，指定直接附加至影格的聲音 (frame.soundLibraryItem) 名稱，這個名稱會與儲存在元件庫的名稱相同。

範例
以下範例會將第一個影格的 soundName 屬性變更為 "song1.mp3"；song1.mp3 必須存在元件庫中：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundName = "song1.mp3";
```

frame.soundSync

適用版本
Flash MX 2004。

用法
```
frame.soundSync
```

說明
屬性；字串，對直接附加至影格的聲音 (frame.soundLibraryItem) 指定其同步化行為。可接受的值為 "event"、"stop"、"start" 和 "stream"。

範例
以下範例會指定聲音要串流：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].soundSync = 'stream';
```

frame.startFrame

適用版本
Flash MX 2004。

用法
```
frame.startFrame
```

說明
唯讀屬性；序列中第一個影格的索引。

範例
在以下範例中，stFrame 會是影格序列中第一個影格的索引。在這個例子中，影格序列會橫跨六個影格，從影格 5 到影格 10。因此，影格 5 到影格 10 之間，任何影格的 stFrame 值都會是 4 (請記住，索引值與影格編號值是不同的)。

```
var stFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[4].startFrame;
fl.trace(stFrame); // 4
var stFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[9].startFrame;
fl.trace(stFrame); // 4
```

frame.tweenEasing

適用版本
Flash MX 2004。

用法
```
frame.tweenEasing
```

說明
屬性；整數，指定要套用至補間動畫物件的加 / 減速量。有效值介於 -100 到 100 之間。如果要移動補間動畫慢慢開始，並且朝動畫結尾的方向逐漸加速補間動畫，請使用 -1 到 –100 之間的值。如果要移動補間動畫快速開始，並且朝動畫結尾的方向逐漸減速，請使用 1 到 100 之間的正值。

範例
以下範例會指定補間動畫物件的移動，應該以相當快的速度開始，並且朝動畫結尾的方向逐漸減速：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenEasing = 50;
```

frame.tweenInstanceName

適用版本
Flash Professional CS5。

用法
```
Frame.tweenInstanceName()
```

說明
屬性；將實體名稱指定到所選移動物件的字串。

範例
下列範例將實體名稱 MyMotionTween 指定到指定的移動物件。

```
theFrame.tweenInstanceName = "MyMotionTween";
```

frame.tweenType

適用版本
Flash MX 2004。

用法
```
frame.tweenType
```

說明
屬性；指定補間動畫類型的字串；可接受的值為 "motion"、"shape" 或 "none"。值 "none" 會移除移動補間動畫。請使用 timeline.createMotionTween() 方法來建立移動補間動畫。

如果您指定 "motion"，則影格中的物件會是元件、文字欄位或群組物件。物件會從目前關鍵影格的位置中開始進行補間動畫，一直進行到下一個關鍵影格中的位置。

如果您指定 "shape"，影格中的物件必須是形狀。物件會從目前關鍵影格中的形狀，轉換為下一個關鍵影格中的形狀。

範例
以下範例會指出物件是移動補間動畫，因此必須從目前關鍵影格中的位置進行補間動畫，一直進行到下一個關鍵影格中的位置：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].tweenType = "motion";
```

frame.useSingleEaseCurve

適用版本
Flash 8。

用法
```
frame.useSingleEaseCurve
```

說明

屬性；Boolean 值。如果是 true，會使用單一自訂加 / 減速曲線，做為所有屬性的加 / 減速資訊。如果是 false，每一個屬性都會擁有自己的加 / 減速曲線。

如果影格沒有套用自訂加 / 減速，會忽略這個屬性。

範例

以下範例會指定在第一個圖層上，第一個影格的所有屬性，應該使用單一自訂的加 / 減速曲線：

```
var theFrame = fl.getDocumentDOM().getTimeline().layers[0].frames[0]
theFrame.useSingleEaseCurve = true;
```

請參閱

frame.getCustomEase(), frame.hasCustomEase, frame.setCustomEase()

第 22 章 HalfEdge 物件

適用版本
Flash MX 2004。

說明
HalfEdge 物件是 Shape 物件邊緣的有向面 (directed side)。每個邊緣有兩個不完整邊緣。您可以在這些不完整邊緣上「移動」，橫向截斷形狀的輪廓。例如，您可以從形狀輪廓的某個不完整邊緣開始，在所有不完整邊緣上移動，然後再返回原來的不完整邊緣。

不完整邊緣是有順序的。一個不完整邊緣代表邊緣的某一邊，而另一個不完整邊緣代表另一邊。

方法摘要
HalfEdge 物件可使用的方法如下：

方法	說明
halfEdge.getEdge()	取得 HalfEdge 物件的 Edge 物件。
halfEdge.getNext()	取得目前輪廓的下一個不完整邊緣。
halfEdge.getOppositeHalfEdge()	取得邊緣另一邊的 HalfEdge 物件。
halfEdge.getPrev()	取得目前輪廓的前一個 HalfEdge 物件。
halfEdge.getVertex()	在 HalfEdge 物件的開頭取得 Vertex 物件。

屬性摘要
HalfEdge 物件可使用的屬性如下：

屬性	說明
halfEdge.id	唯讀；HalfEdge 物件的唯一整數識別名稱。
halfEdge.index	整數，具有 0 或 1 的值，指定這個 HalfEdge 物件在父邊緣中的索引。

halfEdge.getEdge()

適用版本
Flash MX 2004。

用法
```
halfEdge.getEdge()
```

參數
無。

傳回值
Edge 物件。

說明
方法：取得 HalfEdge 物件的 Edge 物件。請參閱 Edge 物件。

範例
以下範例會說明取得指定形狀的邊緣和不完整邊緣：

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var edge = hEdge.getEdge();
```

halfEdge.getNext()

適用版本
Flash MX 2004。

用法
```
halfEdge.getNext()
```

參數
無。

傳回值
HalfEdge 物件。

說明
方法：取得目前輪廓的下一個不完整邊緣。

備註：不完整邊緣有方向和順序，邊緣則沒有這些特性。

範例
以下範例將指定輪廓的下一個不完整邊緣，儲存在 nextHalfEdge 變數中：

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge( 0 );
var nextHalfEdge = hEdge.getNext();
```

halfEdge.getOppositeHalfEdge()

適用版本
Flash MX 2004。

用法
```
halfEdge.getOppositeHalfEdge()
```

參數
無。

傳回值
HalfEdge 物件。

說明
方法；取得邊緣另一邊的 HalfEdge 物件。

範例
下列範例會將 hEdge 對面的不完整邊緣，儲存在 otherHalfEdge 變數中：

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var otherHalfEdge = hEdge.getOppositeHalfEdge();
```

halfEdge.getPrev()

適用版本
Flash MX 2004。

用法
```
halfEdge.getPrev()
```

參數
無。

傳回值
HalfEdge 物件。

說明
方法；取得目前輪廓的前一個 HalfEdge 物件。

備註：不完整邊緣有方向和順序，邊緣則沒有這些特性。

範例
以下範例將指定輪廓的前一個不完整邊緣，儲存在 prevHalfEdge 變數中：

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge( 0 );
var prevHalfEdge = hEdge.getPrev();
```

halfEdge.getVertex()

適用版本
Flash MX 2004。

用法
```
halfEdge.getVertex()
```

參數
無。

傳回值
Vertex 物件

說明
方法：在 HalfEdge 物件開頭的地方取得 Vertex 物件。請參閱 Vertex 物件

範例
以下範例將 hEdge 開頭的 Vertex 物件，儲存在 vertex 變數中：

```
var shape = fl.getDocumentDOM().selection[0];
var edge = shape.edges[0];
var hEdge = edge.getHalfEdge(0);
var vertex = hEdge.getVertex();
```

halfEdge.id

適用版本
Flash MX 2004。

用法
```
halfEdge.id
```

說明
唯讀屬性：HalfEdge 物件的唯一整數識別名稱。

範例
以下範例會在「輸出」面板中，顯示指定不完整邊緣的唯一識別名稱：

```
var shape = fl.getDocumentDOM().selection[0];
alert(shape.contours[0].getHalfEdge().id);
```

halfEdge.index

適用版本
Flash MX 2004。

用法
```
halfEdge.index
```

說明
唯讀屬性：具有 0 或 1 值的整數，指定這個 HalfEdge 物件在父邊緣中的索引。

範例

以下範例會在「輸出」面板中，顯示指定不完整邊緣的索引值：

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var heIndex = hEdge.index;
```

第 23 章 Instance 物件

繼承　Element 物件 > Instance 物件

適用版本
Flash MX 2004。

說明
Instance 是 Element 物件的子類別。

屬性摘要
除了所有的 Element 物件屬性之外，Instance 還具有下列屬性：

屬性	說明
instance.instanceType	唯讀；字串，代表 Instance 類型。
instance.libraryItem	用來初始化這個實體的元件庫項目。

instance.instanceType

適用版本
Flash MX 2004；已經加入 Flash 8 中的 "video" 可能值。

用法
```
instance.instanceType
```

說明
唯讀屬性；字串，代表 Instance 類型。可能的值為 "symbol"、"bitmap"、"embedded video"、"linked video"、"video" 和 "compiled clip"。

在 Flash MX 2004 中，利用 library.addNewItem("video") 加入至元件庫的項目，其 instance.instanceType 的值為 "embedded_video"。在 Flash 8 和更新版本中，該值為 "video"。請參閱 library.addNewItem()。

範例
下列範例顯示影片片段的實體類型為 symbol：

```
// Select a movie clip and then run this script.
var type = fl.getDocumentDOM().selection[0].instanceType;
fl.trace("This instance type is " + type);
```

instance.libraryItem

適用版本
Flash MX 2004。

用法

```
instance.libraryItem
```

說明

屬性：用來初始化這個實體的元件庫項目。您只能將這個屬性變更為相同類型的另一個元件庫項目 (也就是說，您無法設定 symbol 實體參照點陣圖)。請參閱 library 物件。

範例

以下範例將選取的元件變更為參照元件庫中的第一個項目：

```
fl.getDocumentDOM().selection[0].libraryItem = fl.getDocumentDOM().library.items[0];
```

第 24 章 Item 物件

適用版本
Flash MX 2004。

說明
Item 物件為抽象的基底類別。元件庫中的所有內容，都是衍生自 Item。請參閱 library 物件。

方法摘要
Item 物件可使用的方法如下：

方法	說明
item.addData()	將指定的資料新增至元件庫項目。
item.getData()	擷取指定資料的值。
item.hasData()	判斷元件庫項目是否有命名資料。
item.removeData()	從元件庫項目中移除永續性資料。

屬性摘要
Item 物件可使用的屬性如下：

屬性	說明
item.itemType	唯讀；指定元素類型的字串。
item.linkageBaseClass	字串，指定要與元件產生關聯的 ActionScript 3.0 類別。
item.linkageClassName	字串，指定要與元件產生關聯的 ActionScript 2.0 類別。
item.linkageExportForAS	Boolean 值。如果是 true，就會匯出項目以供 ActionScript 使用。
item.linkageExportForRS	Boolean 值。如果是 true，會匯出項目以供執行階段共享使用。
item.linkageExportInFirstFrame	Boolean 值。如果是 true，就會匯出項目至第一個影格。
item.linkageIdentifier	字串，指定連結至目標 SWF 檔時，Flash 要用來識別資源的名稱。
item.linkageImportForRS	Boolean 值。如果是 true，會匯入項目以供執行階段共享使用。
item.linkageURL	字串，指定包含共享資源的 SWF 檔所在 URL。
item.name	字串，指定包含資料夾結構的元件庫項目名稱。

item.addData()

適用版本
Flash MX 2004。

用法

```
item.addData(name, type, data)
```

參數
name 指定資料名稱的字串。

type 指定資料類型的字串。有效的類型為 "integer"、"integerArray"、"double"、"doubleArray"、"string" 和 "byteArray"。

data 要新增至指定元件庫項目的資料。料的類型，依照類型參數的值而有不同。例如，如果類型是 "integer"，資料的值必須是整數，依此類推。

傳回值
無。

說明
方法；將指定的資料新增至元件庫項目。

範例
下列範例會將含整數值 12，名為 myData 的資料，加入至元件庫中的第一個項目：

```
fl.getDocumentDOM().library.items[0].addData("myData", "integer", 12);
```

item.getData()

適用版本
Flash MX 2004。

用法
```
item.getData(name)
```

參數
name 字串，指定要擷取的資料名稱。

傳回值
由 name 參數所指定的資料。傳回的資料類型，依照已儲存資料的類型而有不同。

說明
方法；擷取指定資料的值。

範例
下列範例會從元件庫中的第一個項目取得資料 (名稱為 myData) 的值，並將其儲存在變數 libData 中：

```
var libData = fl.getDocumentDOM().library.items[0].getData("myData");
```

item.hasData()

適用版本
Flash MX 2004。

用法
```
item.hasData(name)
```

參數
name 字串，指定要在元件庫項目中檢查的資料名稱。

傳回值
Boolean 值：如果指定的資料已存在，會傳回 true，否則便傳回 false。

說明
方法：判斷元件庫項目是否有命名資料。

範例
下列範例會在元件庫中的第一個項目包含名為 myData 的資料時，於 「輸出」面板中顯示訊息：

```
if (fl.getDocumentDOM().library.items[0].hasData("myData")){
    fl.trace("Yep, it's there!");
}
```

item.itemType

適用版本
Flash MX 2004。

用法
```
item.itemType
```

說明
唯讀屬性：指定元素類型的字串。此值為下面其中之一："undefined"、"component"、"movie clip"、"graphic"、"button"、"folder"、"font"、"sound"、"bitmap"、"compiled clip"、"screen" 或 "video"。如果這個屬性是 "video"，您可以判斷視訊類型；請參閱 videoItem.videoType。

範例
以下範例會顯示 「輸出」面板中，指定元件庫項目的類型：

```
fl.trace(fl.getDocumentDOM().library.items[0].itemType);
```

item.linkageBaseClass

適用版本
Flash CS3 Professional。

用法
```
item.linkageBaseClass
```

說明
屬性：字串，指定要與元件產生關聯的 ActionScript 3.0 類別。在此指定的值會出現在編寫環境的 「連結」對話方塊中，以及其它包括 「連結」對話方塊控制項的對話方塊，如 「元件屬性」對話方塊 (若要為 ActionScript 2.0 類別指定此值，請使用 item.linkageClassName)。

如果基底類別是元件類型的預設值 (例如，「flash.display.MovieClip」代表影片片段、「flash.display.SimpleButton」代表按鈕，依此類推)，這個屬性會是空字串 ("")。同樣的，若要使項目成為預設的基底類別，請設定此值為空白字串。

當您設定這個值時，不會執行 「連結」對話方塊執行的任何檢查，而且如果 Flash 無法將基底類別設定為指定的值，也不會擲回任何錯誤。例如，在 「連結」對話方塊中設定這個值會強制檢查，以便確保可以在 FLA 檔的類別路徑中找到基底類別。它會確保已在 「發佈設定」對話方塊的 「Flash」索引標籤中選擇了 ActionScript 3.0，依此類推。當您在指令碼中設定這項屬性時，並不會執行這些檢查。

範例
下列程式碼行將會說明一些使用這項屬性的方式：

```
// sets the library item base class to "Sprite"
fl.getDocumentDOM().library.items[0].linkageBaseClass = "flash.display.Sprite";
// sets the library item base class to the default for that item type
fl.getDocumentDOM().library.items[0].linkageBaseClass = "";
// finds and displays the library item's base class
fl.trace(fl.getDocumentDOM().library.items[0].linkageBaseClass);
```

請參閱
document.docClass

item.linkageClassName

適用版本
Flash MX 2004。

用法
```
item.linkageClassName
```

說明
屬性；字串，指定要與元件產生關聯的 ActionScript 2.0 類別 (若要為 ActionScript 3.0 類別指定此值，請使用 item.linkageBaseClass)。

若要定義此屬性，則必須將 item.linkageExportForAS 與 / 或 item.linkageExportForRS 屬性設為 true，並將 item.linkageImportForRS 屬性設為 false。

範例
以下範例會指定與元件庫第一個項目產生關聯的 ActionScript 2.0 類別名稱為 myClass：

```
fl.getDocumentDOM().library.items[0].linkageClassName = "myClass";
```

item.linkageExportForAS

適用版本
Flash MX 2004。

用法
```
item.linkageExportForAS
```

說明
屬性；Boolean 值。如果這個屬性是 true，就會匯出項目以供 ActionScript 使用。您也可以將 item.linkageExportForRS 與 item.linkageExportInFirstFrame 屬性設為 true。

如果您將這個屬性設定為 true，則 item.linkageImportForRS 屬性必須設定為 false。此外，您也必須指定識別名稱 (item.linkageIdentifier) 和 URL (item.linkageURL)。

範例
以下範例會為指定的元件庫項目設定這個屬性：

```
fl.getDocumentDOM().library.items[0].linkageExportForAS = true;
```

item.linkageExportForRS

適用版本
Flash MX 2004。

用法
```
item.linkageExportForRS
```

說明
屬性；Boolean 值。如果這個屬性是 true，會匯出項目以供執行階段共享使用。您也可以將 item.linkageExportForAS 與 item.linkageExportInFirstFrame 屬性設為 true。

如果您將這個屬性設定為 true，則 item.linkageImportForRS 屬性必須設定為 false。此外，您也必須指定識別名稱 (item.linkageIdentifier) 和 URL (item.linkageURL)。

範例
以下範例會為指定的元件庫項目設定這個屬性：

```
fl.getDocumentDOM().library.items[0].linkageExportForRS = true;
```

item.linkageExportInFirstFrame

適用版本
Flash MX 2004。

用法
```
item.linkageExportInFirstFrame
```

說明
屬性；Boolean 值。如果是 true，則在第一個影格中匯出項目；如果是 false，則在第一個實體的影格中匯出項目。如果項目未出現在「舞台」上，表示未匯出。

這個屬性只有當 item.linkageExportForAS 與 / 或 item.linkageExportForRS 設為 true 時，才可以設為 true。

範例
以下範例指定，指定的元件庫項目已匯出至第一個影格：

```
fl.getDocumentDOM().library.items[0].linkageExportInFirstFrame = true;
```

item.linkageIdentifier

適用版本
Flash MX 2004。

用法
```
item.linkageIdentifier
```

說明
屬性；字串，指定連結至目的 SWF 檔時，Flash 要用來識別資源的名稱。當 item.linkageImportForRS、item.linkageExportForAS 和 item.linkageExportForRS 皆設為 false 時，Flash 會忽略這個屬性。相反地，如果這些屬性中有任何是設定為 true，就必須設定這個屬性。

範例
以下範例會指定當字串 my_mc 連結至匯出的目標 SWF 檔時，即使用該字串識別元件庫項目：

```
fl.getDocumentDOM().library.items[0].linkageIdentifier = "my_mc";
```

請參閱
item.linkageURL

item.linkageImportForRS

適用版本
Flash MX 2004。

用法
```
item.linkageImportForRS
```

說明
屬性；Boolean 值；如果是 true，會匯入項目以供執行階段共享使用。如果此屬性設為 true，則 item.linkageExportForAS 和 item.linkageExportForRS 必須同時設為 false。此外，您也必須指定識別名稱 (item.linkageIdentifier) 和 URL (item.linkageURL)。

範例
以下範例會針對指定的元件庫項目，將這個屬性設定為 true：

```
fl.getDocumentDOM().library.items[0].linkageImportForRS = true;
```

item.linkageURL

適用版本
Flash MX 2004。

用法
```
item.linkageURL
```

說明
屬性；字串，指定包含共享資源之 SWF 檔所在位置的 URL。當 item.linkageImportForRS、item.linkageExportForAS 和 item.linkageExportForRS 皆設為 false 時，Flash 會忽略這個屬性。相反地，如果這些屬性中有任何是設定為 true，就必須設定這個屬性。您也可以使用平台專用的格式 (亦即，正斜線 [/] 或反斜線 [\]，依照平台而定)，來指定網站 URL 或檔案名稱。

範例
以下範例會為指定的元件庫項目，指定連結 URL：
```
fl.getDocumentDOM().library.items[0].linkageURL = "theShareSWF.swf";
```

請參閱
item.linkageIdentifier

item.name

適用版本
Flash MX 2004。

用法
```
item.name
```

說明
方法；字串，指定包含資料夾結構的元件庫項目名稱。例如，如果 Symbol_1 位於名稱為 Folder_1 的資料夾之內，則 Symbol_1 的 name 屬性就是 "Folder_1/Symbol_1"。

範例
以下範例會顯示 「輸出」面板中，指定元件庫項目的名稱：
```
fl.trace(fl.getDocumentDOM().library.items[0].name);
```

item.removeData()

適用版本
Flash MX 2004。

用法
```
item.removeData(name)
```

參數
name 指定要從元件庫項目中移除的資料名稱。

傳回值
無。

說明

屬性：從元件庫項目中移除永續性資料。

範例

下列範例會從元件庫的第一個項目中，移除名為 myData 的資料：

```
fl.getDocumentDOM().library.items[0].removeData("myData");
```

第 25 章 Layer 物件

適用版本
Flash MX 2004。

說明
Layer 物件代表時間軸內的圖層。timeline.layers 屬性包含 Layer 物件的陣列，可以透過 fl.getDocumentDOM().getTimeline().layers 進行存取。

屬性摘要
Layer 物件可使用的屬性如下：

屬性	說明
layer.color	字串；用來指定圖層外框的顏色的十六進位值或整數。
layer.frameCount	唯讀；指定圖層中之影格數目的整數。
layer.frames	唯讀；Frame 物件的陣列。
layer.height	指定圖層高度百分比的整數；與 「圖層屬性」對話方塊中的 「圖層」高度值相同。
layer.layerType	指定目前使用圖層的字串；與 「圖層屬性」對話方塊中的 「類型」設定相同。
layer.locked	指定圖層鎖定狀態的 Boolean 值。
layer.name	指定圖層名稱的字串。
layer.outline	針對圖層上所有物件，指定其外框狀態的 Boolean 值。
layer.parentLayer	Layer 物件，代表圖層的包含資料夾、導引或遮色片圖層。
layer.visible	Boolean 值，指定是否顯示或隱藏 「舞台」上圖層的物件。

layer.color

適用版本
Flash MX 2004。

用法
```
layer.color
```

說明
屬性；用來指定圖層外框的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

  這個屬性與 「圖層屬性」對話方塊中的 「外框」顏色設定相同。

範例
下列範例會將第一個圖層的值儲存在 colorValue 變數：

```
var colorValue = fl.getDocumentDOM().getTimeline().layers[0].color;
```

下列範例示範將第一個圖層的顏色設定為紅色的三種方法：

```
fl.getDocumentDOM().getTimeline().layers[0].color=16711680;
fl.getDocumentDOM().getTimeline().layers[0].color="#ff0000";
fl.getDocumentDOM().getTimeline().layers[0].color=0xFF0000;
```

layer.frameCount

適用版本
Flash MX 2004。

用法

```
layer.frameCount
```

說明
唯讀屬性：指定圖層中影格數目的整數。

範例
下列範例會將第一個圖層的影格數目儲存在 fcNum 變數：

```
var fcNum = fl.getDocumentDOM().getTimeline().layers[0].frameCount;
```

layer.frames

適用版本
Flash MX 2004。

用法

```
layer.frames
```

說明
唯讀屬性：Frame 物件的陣列 (請參閱 Frame 物件)。

範例
下列範例會為目前文件中的影格，將變數 frameArray 設定至 Frame 物件的陣列：

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
```

若要判斷影格是否為關鍵影格，請檢查 frame.startFrame 屬性是否符合陣列索引，如下列範例所示：

```
var frameArray = fl.getDocumentDOM().getTimeline().layers[0].frames;
var n = frameArray.length;
for (i=0; i<n; i++) {
    if (i==frameArray[i].startFrame) {
        alert("Keyframe at: " + i);
    }
}
```

layer.height

適用版本
Flash MX 2004。

用法
```
layer.height
```

說明
屬性；指定圖層高度百分比的整數；與 「圖層屬性」對話方塊中的 「圖層」高度值相同。可接受的值代表預設高度的百分比：100、200 或 300。

範例
下列範例會儲存第一個圖層高度設定的百分比值：

```
var layerHeight = fl.getDocumentDOM().getTimeline().layers[0].height;
```

下列範例會將第一個圖層的高度設定為 300%：

```
fl.getDocumentDOM().getTimeline().layers[0].height = 300;
```

layer.layerType

適用版本
Flash MX 2004。

用法
```
layer.layerType
```

說明
屬性；指定目前使用圖層的字串；與 「圖層屬性」對話方塊中的 「類型」設定相同。可接受的值包括 "normal"、"guide"、"guided"、"mask"、"masked" 和 "folder"。

範例
下列範例會將時間軸中的第一個圖層設定為 folder 類型：

```
fl.getDocumentDOM().getTimeline().layers[0].layerType = "folder";
```

layer.locked

適用版本
Flash MX 2004。

用法
```
layer.locked
```

說明
屬性；指定圖層鎖定狀態的 Boolean 值。如果設定為 true，則會鎖定圖層。預設值為 false。

範例
下列範例會將第一個圖層狀態的 Boolean 值儲存在 lockStatus 變數：

```
var lockStatus = fl.getDocumentDOM().getTimeline().layers[0].locked;
```

下列範例會將第一個圖層的狀態設定為未鎖定：

```
fl.getDocumentDOM().getTimeline().layers[0].locked = false;
```

layer.name

適用版本
Flash MX 2004。

用法

```
layer.name
```

說明
屬性；指定圖層名稱的字串。

範例
下列範例會將目前文件中的第一個圖層名稱設定為 foreground：

```
fl.getDocumentDOM().getTimeline().layers[0].name = "foreground";
```

layer.outline

適用版本
Flash MX 2004。

用法

```
layer.outline
```

說明
屬性；針對圖層上所有物件，指定其外框狀態的 Boolean 值。如果設定為 true，圖層上的所有物件都只會出現外框。如果設定為 false，物件會以建立時的外觀出現。

範例
下列範例會讓第一個圖層上的所有物件都只出現外框：

```
fl.getDocumentDOM().getTimeline().layers[0].outline = true;
```

layer.parentLayer

適用版本
Flash MX 2004。

用法

```
layer.parentLayer
```

說明

屬性；Layer 物件，代表圖層的包含資料夾、導引或遮色片圖層。父圖層必須為圖層前面的資料夾、導引或遮色片圖層，或前後圖層的 parentLayer。設定圖層的 parentLayer 不會移動圖層在清單中的位置；嘗試將圖層的 parentLayer 設定至需要移動的圖層，不會有效果。最上層圖層請使用 null。

範例

以下範例會使用相同時間軸的相同層級中的兩個圖層。第一個圖層 (layers[0]) 會轉換成資料夾，然後設定為第二個圖層 (layers[1]) 的父資料夾。這個動作會將第二個圖層移入第一個圖層中。

```
var parLayer = fl.getDocumentDOM().getTimeline().layers[0];
parLayer.layerType = "folder";
fl.getDocumentDOM().getTimeline().layers[1].parentLayer = parLayer;
```

layer.visible

適用版本

Flash MX 2004。

用法

```
layer.visible
```

說明

屬性；Boolean 值，指定是否顯示或隱藏 「舞台」上圖層的物件。如果設定為 true，圖層中的所有物件都是可見的；如果是 false，則是隱藏的。預設值為 true。

範例

下列範例會讓第一個圖層中的所有物件都不可見：

```
fl.getDocumentDOM().getTimeline().layers[0].visible = false;
```

第 26 章 library 物件

適用版本
Flash MX 2004。

說明
Library 物件代表 「元件庫」面板。它是 Document 物件的屬性 (請參閱 document.library)，可以使用 fl.getDocumentDOM().library 存取。

library 物件包括不同類型項目的陣列，包括元件、點陣圖、聲音和視訊。

方法摘要
library 物件可使用的方法如下：

方法	說明
library.addItemToDocument()	將目前或指定的項目，新增至 「舞台」的指定位置。
library.addNewItem()	在 「元件庫」面板中，建立一個指定類型的新項目，並將這個新項目設定為目前選取的項目。
library.deleteItem()	從 「元件庫」面板中，刪除目前的項目或指定的項目。
library.duplicateItem()	製作目前選取或指定項目的副本。
library.editItem()	以 「編輯」模式開啟目前選取或指定的項目。
library.expandFolder()	在元件庫中，展開或收合目前選取或指定的資料夾。
library.findItemIndex()	傳回元件庫項目的索引值 (從零開始)。
library.getItemProperty()	取得選取項目的屬性。
library.getItemType()	取得目前由元件庫路徑選取或指定的物件類型。
library.getSelectedItems()	取得元件庫中目前選取的所有項目的陣列。
library.importEmbeddedSWF()	將 SWF 檔以編譯後的影片片段匯入元件庫中。
library.itemExists()	檢查指定的項目是否存在元件庫中。
library.moveToFolder()	將目前選取或指定的元件庫項目移至指定的資料夾中。
library.newFolder()	在目前選取的資料夾中，使用指定的名稱建立一個新的資料夾，如果未提供 folderName 參數的話，則使用預設的名稱 ("untitled folder #") 建立資料夾。
library.renameItem()	在 「元件庫」面板中，重新命名目前選取的元件庫項目。
library.selectAll()	選取或取消選取所有元件庫中的項目。
library.selectItem()	選取指定的元件庫項目。
library.selectNone()	取消選取所有元件庫項目。
library.setItemProperty()	設定所有選取元件庫項目的屬性 (忽略資料夾)。
library.updateItem()	更新指定項目。

library 物件的屬性摘要
library 物件可使用的屬性如下：

屬性	說明
library.items	元件庫中 Item 物件的陣列

library.addItemToDocument()

適用版本
Flash MX 2004。

用法
```
library.addItemToDocument(position [, namePath])
```

參數
position 點，指定 「舞台」上項目中心所在的 x,y 位置。

namePath 指定項目名稱的字串。如果項目位於資料夾中，您可以使用斜線標記語法來指定項目的名稱和路徑。如果未指定 namePath，則會使用目前選取的元件庫。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功將項目新增至文件中，便傳回 true；否則便傳回 false。

說明
方法；將目前或指定的項目，新增至 「舞台」的指定位置。

範例
下列範例會將目前選取的項目，新增至 「舞台」位置 (3, 60)：

```
fl.getDocumentDOM().library.addItemToDocument({x:3, y:60});
```

下列範例會將位於元件庫資料夾 folder1 中的 Symbol1 項目，加入至 「舞台」的 (550, 485) 位置：

```
fl.getDocumentDOM().library.addItemToDocument({x:550.0, y:485.0}, "folder1/Symbol1");
```

library.addNewItem()

適用版本
Flash MX 2004。

用法
```
library.addNewItem(type [, namePath])
```

參數
type 字串，指定建立的項目類型。type 的可接受值只有 "video"、"movie clip"、"button"、"graphic"、"bitmap"、"screen" 和 "folder" (因此，舉例來說，您無法使用這個方法將聲音新增至元件庫)。指定資料夾路徑，等同於呼叫這個方法之前使用 library.newFolder()。

namePath 字串，指定要新增的項目名稱。如果項目位於資料夾中，請使用斜線標記語法來指定項目的名稱和路徑。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功建立項目，會傳回 true；否則便傳回 false。

說明
方法：在 「元件庫」面板中，建立一個指定類型的新項目，並將這個新項目，設定至目前選取的項目中。如需有關將項目 (包括聲音等項目) 匯入至元件庫的詳細資訊，請參閱 document.importFile()。

範例
下列範例會在名為 folderTwo 的新資料夾中，建立名為 start 的新按鈕項目：

```
fl.getDocumentDOM().library.addNewItem("button", "folderTwo/start");
```

library.deleteItem()

適用版本
Flash MX 2004。

用法
```
library.deleteItem([namePath])
```

參數
namePath 字串，指定要刪除的項目名稱。如果項目位於資料夾中，您可以使用斜線標記語法來指定項目的名稱和路徑。如果您傳送了資料夾名稱，則資料夾和其中所有的項目都會被刪除。如果未指定名稱，Flash 會刪除目前選取的項目。若要刪除「元件庫」面板中所有的項目，請在使用這個方法前，先選取所有項目。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功刪除項目，會傳回 true；否則便傳回 false。

說明
方法：從 「元件庫」面板中，刪除目前的項目或指定的項目。如果同時選取多個項目，這個方法會同時影響這些項目。

範例
以下範例會刪除目前選取的項目：

```
fl.getDocumentDOM().library.deleteItem();
```

下列範例會從元件庫資料夾 Folder_1 中，刪除項目 Symbol_1：

```
fl.getDocumentDOM().library.deleteItem("Folder_1/Symbol_1");
```

library.duplicateItem()

適用版本
Flash MX 2004。

用法
```
library.duplicateItem( [ namePath ] )
```

參數
namePath 字串，指定要重製的項目名稱。如果項目位於資料夾中，您可以使用斜線標記語法來指定項目的名稱和路徑。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功重製項目，會傳回 true；否則便傳回 false。如果選取一個以上的項目，Flash 會傳回 false。

說明
方法：製作目前選取或指定項目的副本。新的項目擁有預設的名稱 (如 item copy)，而且會設定為目前選取的項目。如果選取一個以上的項目，這個命令會失敗。

範例
下列範例會在元件庫資料夾 test 中，建立項目 square 的副本：

```
fl.getDocumentDOM().library.duplicateItem("test/square");
```

library.editItem()

適用版本
Flash MX 2004。

用法
```
library.editItem([namePath])
```

參數
namePath 指定項目名稱的字串。如果項目位於資料夾中，您可以使用斜線標記語法來指定項目的名稱和路徑。如果未指定 namePath，會以「編輯」模式開啟單一選取元件庫項目。如果目前未選取或選取一個以上的元件庫項目，會出現主要時間軸的第一個場景，以供編輯。這個參數是選擇性參數。

傳回值
Boolean 值：如果指定的項目已存在而且可以編輯，會傳回 true；否則便傳回 false。

說明
方法：以「編輯」模式開啟目前選取或指定的項目。

範例
下列範例會在元件庫的 test 資料夾中，開啟項目 circle 以供編輯：

```
fl.getDocumentDOM().library.editItem("test/circle");
```

library.expandFolder()

適用版本
Flash MX 2004。

用法
```
library.expandFolder(bExpand [, bRecurseNestedParents [, namePath]])
```

參數
bExpand Boolean 值：如果是 true，會展開資料夾；如果是 false (預設值)，會收合資料夾。

bRecurseNestedParents Boolean 值：如果是 true，指定的資料夾內的所有資料夾會根據 bExpand 的值，展開或收合。預設值為 false。這個參數是選擇性參數。

namePath 字串：用以指定要展開或收合的資料夾名稱，也可以選擇性地指定資料夾路徑。如果未指定這個參數，會將這個方法套用至目前選擇的資料夾。這個參數是選擇性參數。

傳回值
Boolean 值：如果項目已成功地展開或收合，會傳回 true；如果不成功，或是指定的項目不是資料夾，會傳回 false。

說明
方法；在元件庫中，展開或收合目前選取或指定的資料夾。

範例
下列範例會收合元件庫中的 test 資料夾，也會收合 test 資料夾中的所有資料夾 (如果有的話)：

```
fl.getDocumentDOM().library.expandFolder(false, true, "test");
```

library.findItemIndex()

適用版本
Flash MX 2004。

用法
```
library.findItemIndex(namePath)
```

參數
namePath 指定項目名稱的字串。如果項目位於資料夾中，您可以使用斜線標記語法來指定項目的名稱和路徑。

傳回值
整數值；代表項目的索引值 (從零開始)。

說明
方法；傳回元件庫項目的索引值 (從零開始)。元件庫索引是平面的，所以資料夾會被視為主索引的一部分。可以使用資料夾路徑來指定巢狀項目。

範例
下列範例會將 test 資料夾中，元件庫項目 square 的索引值 (從零開始)，儲存在變數 sqIndex 中，然後在對話方塊中顯示索引值：

```
var sqIndex = fl.getDocumentDOM().library.findItemIndex("test/square");
alert(sqIndex);
```

library.getItemProperty()

適用版本
Flash MX 2004。

用法
```
library.getItemProperty(property)
```

參數
property 字串。如果您要取得可以做為 property 參數值的清單，請參閱 Item 物件以及其子類別的屬性摘要。

傳回值
屬性的字串值。

說明
方法；取得選取項目的屬性。

範例
下列範例會顯示一個對話方塊，其中包含了使用 ActionScript 或執行階段共享時，參考元件的「連結識別名稱」值：

```
alert(fl.getDocumentDOM().library.getItemProperty("linkageIdentifier"));
```

library.getItemType()

適用版本
Flash MX 2004。

用法
```
library.getItemType([namePath])
```

參數
namePath 指定項目名稱的字串。如果項目位於資料夾中，請使用斜線標記語法來指定項目的名稱和路徑。如果未指定 namePath，Flash 會提供目前選取的類型。如果目前選取了一個以上的項目，而且未提供 namePath，Flash 會忽略這個命令。這個參數是選擇性參數。

傳回值
指定物件類型的字串值。如需可能的傳回值，請參閱 item.itemType。

說明
方法；取得目前由元件庫路徑選取或指定的物件類型。

範例
下列範例會顯示一個對話方塊，其中包含 Folder_1/Folder_2 資料夾中的 Symbol_1 項目類型：

```
alert(fl.getDocumentDOM().library.getItemType("Folder_1/Folder_2/Symbol_1"));
```

library.getSelectedItems()

適用版本
Flash MX 2004。

參數
無。

傳回值
元件庫中目前所有選取項目值的陣列。

說明
方法；取得元件庫中目前選取的所有項目的陣列。

範例
下列範例會將目前選取元件庫項目的陣列 (以本例來說，是數個音效檔)，儲存在 selItems 變數中，然後將陣列中第一個音效檔的 sampleRate 屬性變更為 11 kHz：

```
var selItems = fl.getDocumentDOM().library.getSelectedItems();
selItems[0].sampleRate = "11 kHz";
```

library.importEmbeddedSWF()

適用版本
Flash MX 2004。

用法
```
library.importEmbeddedSWF(linkageName, swfData [, libName])
```

參數
linkageName 字串；提供根節點影片片段的 SWF 連結名稱。

swfData 二進位 SWF 資料陣列；來自外部元件庫或 DLL。

libName 字串，指定建立項目的元件庫名稱。如果這個名稱已被使用，這個方法會建立一個替代的名稱。這個參數是選擇性參數。

傳回值
無。

說明
方法；將 SWF 檔以編譯後的影片片段匯入元件庫中。這個方法可讓您在元件庫裡內嵌編譯後的 SWF 檔，與使用 「檔案 > 匯入 > SWF」的方式不同。但沒有相對應的使用者介面功能，而且這個方法必須和外部元件庫或 DLL 搭配使用 (請參閱：第 489 頁 「C 語言層次擴充」)。

您所匯入的 SWF 檔必須具有一個包含所有內容的最上層影片片段。該影片片段的連結識別名稱應該設為與傳送至此方法的 linkageName 參數相同的值。

範例
下列範例會將 SWF 檔 (帶有 MyMovie 的 linkageName 值) 當做已編譯的影片加入至元件庫中，影片名稱為 Intro：

```
fl.getDocumentDOM().library.importEmbeddedSWF("MyMovie", swfData, "Intro");
```

library.itemExists()

適用版本
Flash MX 2004。

用法

```
library.itemExists(namePath)
```

參數
namePath 指定項目名稱的字串。如果項目位於資料夾中，請使用斜線標記語法來指定項目的名稱和路徑。

傳回值
Boolean 值：如果指定的項目已存在元件庫中，會傳回 true；否則便傳回 false。

說明
方法：檢查指定的項目是否存在元件庫中。

範例
下列範例會根據項目 Symbol_1 是否存在於元件庫資料夾 Folder_1 中，在對話方塊中顯示 true 或 false：

```
alert(fl.getDocumentDOM().library.itemExists('Folder_1/Symbol_1'));
```

library.items

適用版本
Flash MX 2004。

用法

```
library.items
```

說明
屬性：元件庫中項目物件的陣列。

範例
下列範例會將所有元件庫項目的陣列儲存在 itemArray 變數中：

```
var itemArray = fl.getDocumentDOM().library.items;
```

library.moveToFolder()

適用版本
Flash MX 2004。

用法
```
library.moveToFolder(folderPath [, itemToMove [, bReplace]])
```

參數
folderPath 字串；以 "FolderName" 或 "FolderName/FolderName" 格式指定資料夾路徑。若要將項目移至最上層，請為 folderPath 指定空字串 ("")。

itemToMove 字串，指定要移動的項目名稱。如果未指定 itemToMove，會移動目前選取的項目。這個參數是選擇性參數。

bReplace Boolean 值。如果具有相同名稱的項目已經存在，將 bReplace 參數指定為 true，會以要移動的項目來取代現存的項目。如果是 false，要丟棄的項目名稱會變更為唯一名稱。預設值為 false。這個參數是選擇性參數。

傳回值
Boolean 值：如果項目已經移動，會傳回 true；否則便傳回 false。

說明
方法；將目前選取或指定的元件庫項目移至指定的資料夾中。如果 folderPath 參數是空的，項目會移動至最上層。

範例
下列範例會將項目 Symbol_1 移至元件庫資料夾 new，並取代資料夾中相同名稱的項目：

```
fl.getDocumentDOM().library.moveToFolder("new", "Symbol_1", true);
```

library.newFolder()

適用版本
Flash MX 2004。

用法
```
library.newFolder([folderPath])
```

參數
folderPath 字串，指定要建立的資料夾名稱。如果指定為路徑，而且這個路徑不存在，則會建立路徑。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功建立資料夾，會傳回 true；否則便傳回 false。

說明
方法；在目前選取的資料夾中，使用指定的名稱建立一個新的資料夾，如果未提供 folderName 參數的話，則使用預設的名稱 ("untitled folder #") 建立資料夾。

範例
下列範例會建立兩個新的元件庫資料夾。第二個資料夾是第一個資料夾的子資料夾：

```
fl.getDocumentDOM().library.newFolder("first/second");
```

library.renameItem()

適用版本
Flash MX 2004。

用法
```
library.renameItem(name)
```

參數
name　指定元件庫項目新名稱的字串。

傳回值
如果項目名稱變更成功，Boolean 值為 true，否則為 false。如果選取了多個項目，則不會變更名稱，而且傳回值會是 false (以符合介面行為)。

說明
方法；在「元件庫」面板中，重新命名目前選取的元件庫項目。

範例
下列範例會將目前選取的元件庫項目重新命名為 new name：

```
fl.getDocumentDOM().library.renameItem("new name");
```

library.selectAll()

適用版本
Flash MX 2004。

用法
```
library.selectAll([bSelectAll])
```

參數
bSelectAll　Boolean 值；指定是否選取或取消選取元件庫中的所有項目。忽略這個參數或使用預設值 true，會選取元件庫中的所有項目；使用 false，則會取消選取所有元件庫項目。這個參數是選擇性參數。

傳回值
無。

說明
方法；選取或取消選取所有元件庫中的項目。

範例

下列範例會選取元件庫中的所有項目：

```
fl.getDocumentDOM().library.selectAll();
fl.getDocumentDOM().library.selectAll(true);
```

下列範例會取消選取元件庫中的所有項目：

```
fl.getDocumentDOM().library.selectAll(false);
fl.getDocumentDOM().library.selectNone();
```

library.selectItem()

適用版本

Flash MX 2004。

用法

```
library.selectItem(namePath [, bReplaceCurrentSelection [, bSelect]])
```

參數

namePath 指定項目名稱的字串。如果項目位於資料夾中，您可以使用斜線標記語法來指定項目的名稱和路徑。

bReplaceCurrentSelection Boolean 值；指定是取代目前的選擇，或是將項目新增至目前的選擇中。預設值為 true (取代目前的選取範圍)。這個參數是選擇性參數。

bSelect Boolean 值；指定是否選取或取消選取項目。預設值為 true (選取)。這個參數是選擇性參數。

傳回值

Boolean 值；如果指定的項目已經存在，會傳回 true；否則便傳回 false。

說明

方法；選取指定的元件庫項目。

範例

下列範例會將元件庫中的目前選取範圍變更為 untitled Folder_1 中的 Symbol_1：

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1");
```

下列範例會延伸元件庫中目前選取項目的範圍，以便將 untitled Folder_1 中的 Symbol_1 包含在內：

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", false);
```

下列範例會取消選取 untitled Folder_1 中的 Symbol_1，但不變更其它選取的項目：

```
fl.getDocumentDOM().library.selectItem("untitled Folder_1/Symbol_1", true, false);
```

library.selectNone()

適用版本

Flash MX 2004。

用法

```
library.selectNone()
```

參數
無。

傳回值
無。

說明
方法：取消選取所有元件庫項目。

範例
下列範例會取消選取元件庫中的所有項目：

```
fl.getDocumentDOM().library.selectNone();
fl.getDocumentDOM().library.selectAll(false);
```

library.setItemProperty()

適用版本
Flash MX 2004。

用法
```
library.setItemProperty(property, value)
```

參數
property 字串，代表要設定的屬性名稱。如需屬性的清單，請參閱 Item 物件的屬性摘要表以及其子類別的屬性摘要。若要瞭解哪些物件是 Item 物件的子類別，請參閱 第 10 頁 「DOM 結構摘要」。

value 指定給指定屬性的值。

傳回值
無。

說明
方法；設定所有選取元件庫項目的屬性 (忽略資料夾)。

範例
下列範例會將值按鈕指定給一或多個選取元件庫項目的 symbolType 屬性。在這個例子中，項目必須是 SymbolItem 物件；symbolType 是 SymbolItem 物件的有效屬性。

```
fl.getDocumentDOM().library.setItemProperty("symbolType", "button");
```

library.updateItem()

適用版本
Flash MX 2004。

用法

```
library.updateItem([namePath])
```

參數

namePath 指定項目名稱的字串。如果項目位於資料夾中，請使用斜線標記語法來指定項目的名稱和路徑。這個方式等於在項目上按一下滑鼠右鍵，然後從使用者介面選單中選取 「更新」。如果未提供名稱，則會更新目前的選取範圍。這個參數是選擇性參數。

傳回值

Boolean 值：如果 Flash 成功更新了這個項目，會傳回 true；否則便傳回 false。

說明

方法；更新指定的項目。

範例

下列範例會顯示一個對話方塊，顯示更新目前選取的項目 (true)，或是不更新 (false)：

```
alert(fl.getDocumentDOM().library.updateItem());
```

第 27 章 Math 物件

適用版本
Flash MX 2004。

說明
Math 物件是做為 flash 物件的唯讀屬性使用；請參閱 fl.Math。這個物件提供執行一般數學運算的方法。

方法摘要
Math 物件可使用的方法如下：

方法	說明
Math.concatMatrix()	執行矩陣連接並傳回結果。
Math.invertMatrix()	傳回指定矩陣的反轉。
Math.pointDistance()	計算兩點之間的距離。

Math.concatMatrix()

適用版本
Flash MX 2004。

用法
```
Math.concatMatrix(mat1, mat2)
```

參數
mat1, mat2 指定要連接的 Matrix 物件 (請參閱 Matrix 物件)。每一個參數都必須是具備 a、b、c、d、tx 和 ty 欄位的物件。

傳回值
已連接物件的矩陣。

說明
方法；執行矩陣連接並傳回結果。

範例
下列範例會將目前選取的物件儲存在 elt 變數中，將物件矩陣和視圖矩陣相乘，並將計算值儲存在 mat 變數中：

```
var elt = fl.getDocumentDOM().selection[0];
var mat = fl.Math.concatMatrix( elt.matrix , fl.getDocumentDOM().viewMatrix );
```

Math.invertMatrix()

適用版本
Flash MX 2004。

用法

```
Math.invertMatrix(mat)
```

參數

mat 指出要反轉的 Matrix 物件 (請參閱 Matrix 物件)。它必須具備下列欄位：a、b、c、d、tx 和 ty。

傳回值

Matrix 物件是原始矩陣的反轉。

說明

方法；傳回指定矩陣的反轉。

範例

下列範例會將目前選取的物件儲存在 elt 變數中，將這個矩陣指定給 mat 變數，並將矩陣的反轉儲存在 inv 變數中：

```
var elt = fl.getDocumentDOM().selection[0];
var mat = elt.matrix;
var inv = fl.Math.invertMatrix( mat );
```

Math.pointDistance()

適用版本

Flash MX 2004。

用法

```
Math.pointDistance(pt1, pt2)
```

參數

pt1, pt2 指定要測量距離的兩個點。

傳回值

浮點數值，代表點與點之間的距離。

說明

方法；計算兩點之間的距離。

範例

下列範例會將 pt1 和 pt2 之間的距離值儲存在 dist 變數中：

```
var pt1 = {x:10, y:20}
var pt2 = {x:100, y:200}
var dist = fl.Math.pointDistance(pt1, pt2);
```

第 28 章 Matrix 物件

適用版本
Flash MX 2004。

說明
Matrix 物件代表變形矩陣。

屬性摘要
Matrix 物件可使用的屬性如下：

屬性	說明
matrix.a	在變形矩陣中，指定 (0,0) 元素的浮點數值。
matrix.b	在矩陣中，指定 (0,1) 元素的浮點數值。
matrix.c	在矩陣中，指定 (1,0) 元素的浮點數值。
matrix.d	在矩陣中，指定 (1,1) 元素的浮點數值。
matrix.tx	浮點值，用以指定元件註冊點或形狀中心的 x 軸位置。
matrix.ty	浮點值，用以指定元件註冊點或形狀中心的 y 軸位置。

matrix.a

適用版本
Flash MX 2004。

用法
```
matrix.a
```

說明
屬性：在變形矩陣中，指定 (0,0) 元素的浮點數值。這個值代表物件 x 軸的縮放比例因數。

範例
矩陣中的 a 和 d 屬性代表縮放。在下列範例中，它們的值分別設定為 2 和 3，將選取物件的寬度放大兩倍，高度放大三倍：

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 2;
mat.d = 3;
fl.getDocumentDOM().selection[0].matrix = mat;
```

您可以旋轉物件，方法是以相對於彼此的方式設定 a、b、c 和 d 矩陣屬性，也就是 a = d 且 b = -c。例如，0.5、0.8、-0.8 和 0.5 等值會將物件旋轉 60 度：

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.a = 0.5;
mat.b = 0.8;
mat.c = 0.8*(-1);
mat.d = 0.5;
fl.getDocumentDOM().selection[0].matrix = mat;
```

您可以設定 a = d = 1 且 c = b = 0，將物件重設回原始形狀。

matrix.b

適用版本
Flash MX 2004。

用法
```
matrix.b
```

說明
屬性；在矩陣中，指定 (0,1) 元素的浮點數值。這個值代表形狀的垂直傾斜；它可以讓 Flash 沿著垂直軸來移動形狀的右邊緣。

矩陣中的 matrix.b 和 matrix.c 屬性代表傾斜 (請參閱 matrix.c)。

範例
在下列範例中，您可以分別將 b 和 c 設定為 -1 和 0；這些設定會讓物件垂直傾斜 45 度：

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.b = -1;
mat.c = 0;
fl.getDocumentDOM().selection[0].matrix = mat;
```

若要將物件傾斜回原始形狀，您可以將 b 和 c 設定為 0。

請參閱 matrix.a 的範例。

matrix.c

適用版本
Flash MX 2004。

用法
```
matrix.c
```

說明
屬性；在矩陣中，指定 (1,0) 元素的浮點數值。這個值可以讓 Flash 沿著水平軸來移動物件下邊緣，以傾斜物件。

矩陣中的 matrix.b 和 matrix.c 屬性代表傾斜。

範例
請參閱 matrix.b 的範例。

matrix.d

適用版本
Flash MX 2004。

用法
```
matrix.d
```

說明
屬性；在矩陣中，指定 (1,1) 元素的浮點數值。這個值代表物件 y 軸的縮放比例因數。

範例
請參閱 matrix.a 的範例。

matrix.tx

適用版本
Flash MX 2004。

用法
```
matrix.tx
```

說明
屬性；浮點值，用以指定元件註冊點 (以及 「原點」或 「零點」) 或形狀中心的 x 軸位置。它會定義變形的 x 轉譯。

您可以設定 matrix.tx 和 matrix.ty 屬性來移動物件 (請參閱 matrix.ty)。

範例
在下列範例中，會將 tx 和 ty 設定為 0，以將物件的註冊點移至文件中的點 0,0：

```
var mat = fl.getDocumentDOM().selection[0].matrix;
mat.tx = 0;
mat.ty = 0;
fl.getDocumentDOM().selection[0].matrix = mat;
```

matrix.ty

適用版本
Flash MX 2004。

用法

```
matrix.ty
```

說明

屬性；浮點值，用以指定元件註冊點或形狀中心的 y 軸位置。它會定義變形的 y 轉譯。

您可以設定 matrix.tx 和 matrix.ty 屬性來移動物件。

範例

請參閱 matrix.tx 的範例。

第 29 章 outputPanel 物件

適用版本
Flash MX 2004。

說明
這個物件代表 「輸出」面板，用來顯示語法錯誤等疑難排解資訊。若要存取這個物件，請使用 fl.outputPanel (或 flash.outputPanel)。請參閱 fl.outputPanel。

方法摘要
outputPanel 物件會使用下列方法：

方法	說明
outputPanel.clear()	清除 「輸出」面板的內容。
outputPanel.save()	將 「輸出」面板的內容儲存至本機的文字檔。
outputPanel.trace()	新增行到 「輸出」面板的內容中，並以換行字元結束。

outputPanel.clear()

適用版本
Flash MX 2004。

用法
```
outputPanel.clear()
```

參數
無。

傳回值
無。

說明
方法：清除 「輸出」面板的內容。您可以使用這個方法，在批次處理的應用程式中，清除錯誤清單，或將這個方法與 outputPanel.save() 搭配使用，以遞增的方式來儲存錯誤。

範例
下列範例會清除 「輸出」面板目前的內容：

```
fl.outputPanel.clear();
```

outputPanel.save()

適用版本
Flash MX 2004；在 Flash 8 中已加入 bUseSystemEncoding 參數。

用法
```
outputPanel.save(fileURI [, bAppendToFile [ , bUseSystemEncoding]])
```

參數
fileURI 字串；指定本機檔案包含 「輸出」面板的內容，表示為 file:/// URI。

bAppendToFile 選擇性的 Boolean 值。若為 true，此方法會將 「輸出」面板的內容附加到輸出檔案，若為 false，則覆寫已經存在的輸出檔案。預設值為 false。

bUseSystemEncoding 選擇性的 Boolean 值。若為 true，會使用系統編碼儲存 「輸出」面板文字；若為 false，則使用 UTF-8 編碼儲存 「輸出」面板文字，且文字開頭有 「位元組順序標記」字元。預設值為 false。

傳回值
無。

說明
方法；以覆寫檔案或附加至檔案的方式，將 「輸出」面板的內容儲存至本機文字檔。

如果 fileURI 無效或未指定，則會報告錯誤。

在批次處理時，這個方法很有用。例如，您可以建立一個 JSFL 檔來編譯數個組件。任何編譯錯誤都會出現在 「輸出」面板中，而且您可以使用這個方法，將產生的錯誤儲存至文字檔中，使用中的建置系統可以自動解析這個文字檔。

範例
下列範例會將 「輸出」面板的內容儲存到 /tests 資料夾中的 batch.log 檔，覆寫已經存在的 batch.log 檔：

```
fl.outputPanel.save("file:///c|/tests/batch.log");
```

outputPanel.trace()

適用版本
Flash MX 2004。

用法
```
outputPanel.trace(message)
```

參數
message 包含要新增至 「輸出」面板文字的字串。

傳回值
無。

說明
方法；傳送以新的一行結束的文字字串到 「輸出」面板，如果仍不可見，便會顯示 「輸出」面板。此方法與 fl.trace() 相同，運作方式與 ActionScript 中的 trace() 陳述式相同。

若要傳送空白行，請使用 outputPanel.trace("") 或 outputPanel.trace("\n")。您可以使用後行內命令，讓 \n 成為 message 字串的一部分。

範例

下列範例會在「輸出」面板中顯示數行文字：

```
fl.outputPanel.clear();
fl.outputPanel.trace("Hello World!!!");
var myPet = "cat";
fl.outputPanel.trace("\nI have a " + myPet);
fl.outputPanel.trace("");
fl.outputPanel.trace("I love my " + myPet);
fl.outputPanel.trace("Do you have a " + myPet +"?");
```

第 30 章 Oval 物件

繼承 Element 物件 > Shape 物件 > Oval 物件

適用版本
Flash CS3 Professional。

說明
Oval 物件是使用 「基本橢圓形」工具繪製的形狀。若要判斷某個項目是否為 Oval 物件，請使用 shape.isOvalObject。

屬性摘要
除了 Shape 物件屬性，Oval 物件還可以搭配下列屬性使用。若要設定 Oval 物件的屬性，請使用 document.setOvalObjectProperty()。

屬性	說明
OvalObject.closePath	唯讀；指定是否選取 「屬性」檢測器中 「封閉路徑」核取方塊的 Boolean 值。
OvalObject.endAngle	唯讀；指定 Oval 物件結束角度的浮點值。
OvalObject.innerRadius	唯讀；指定 Oval 物件內半徑為百分比的浮點值。
OvalObject.startAngle	唯讀；指定 Oval 物件開始角度的浮點值。

OvalObject.closePath

適用版本
Flash CS3 Professional。

用法
```
OvalObject.closePath
```

說明
唯讀屬性；指定是否選取 「屬性」檢測器中 「封閉路徑」核取方塊的 Boolean 值。如果物件的開始角度和結束角度都相同，在這些值變更之前，設定這項屬性將沒有任何作用。

若要設定此值，請使用 document.setOvalObjectProperty()。

範例
下列範例會取消選取 OvalObject.closePath 屬性：

```
fl.getDocumentDOM().setOvalObjectProperty("closePath",false);
```

請參閱
document.setOvalObjectProperty()、shape.isOvalObject

OvalObject.endAngle

適用版本
Flash CS3 Professional。

用法
```
OvalObject.endAngle
```

說明
唯讀屬性：指定 Oval 物件結束角度的浮點值。可接受的值從 0 到 360。

若要設定此值，請使用 document.setOvalObjectProperty()。

範例
下列範例會將已選取 Oval 物件的結束角度設定為 270。

```
fl.getDocumentDOM().setOvalObjectProperty("endAngle",270);
```

請參閱
document.setOvalObjectProperty()、OvalObject.startAngle、shape.isOvalObject

OvalObject.innerRadius

適用版本
Flash CS3 Professional。

用法
```
OvalObject.innerRadius
```

說明
唯讀屬性：指定 Oval 物件內半徑為百分比的浮點值。可接受的值從 0 到 99。

若要設定此值，請使用 document.setOvalObjectProperty()。

範例
下列範例會將已選取 Oval 物件的內半徑設定為百分之 50：

```
fl.getDocumentDOM().setOvalObjectProperty("innerRadius",50);
```

請參閱
document.setOvalObjectProperty()、shape.isOvalObject

OvalObject.startAngle

適用版本
Flash CS3 Professional。

用法

```
OvalObject.startAngle
```

說明

唯讀屬性：指定 Oval 物件開始角度的浮點值。可接受的值從 0 到 360。

若要設定此值，請使用 document.setOvalObjectProperty()。

範例

下列範例會將已選取 Oval 物件的開始角度設定為 270：

```
fl.getDocumentDOM().setOvalObjectProperty("startAngle",270);
```

請參閱

document.setOvalObjectProperty()、OvalObject.endAngle、shape.isOvalObject

第 31 章 Parameter 物件

適用版本
Flash MX 2004。

說明
Parameter 物件類型存取自 componentInstance.parameters 陣列 (此陣列對應至編寫工具中的組件 「屬性」檢測器)。

方法摘要
Parameter 物件可使用的方法如下：

方法	說明
parameter.insertItem()	插入項目到清單、物件或陣列。
parameter.removeItem()	移除螢幕或組件參數的清單、物件或陣列類型元素。

屬性摘要
Parameter 物件可使用的屬性如下：

屬性	說明
parameter.category	字串，指定 screen 參數或 componentInstance 參數的 category 屬性。
parameter.listIndex	指定選取清單項目值的整數。
parameter.name	唯讀；指定參數名稱的字串。
parameter.value	與 「組件檢測器」的 「參數」索引標籤、「屬性」檢測器或螢幕 「屬性」檢測器的 「參數」索引標籤中的 「值」欄位相對應。
parameter.valueType	唯讀；指定螢幕或組件參數類型的字串。
parameter.verbose	指定顯示參數的位置。

parameter.category

適用版本
Flash MX 2004。

用法
```
parameter.category
```

說明
屬性；字串，指定 screen 參數或 componentInstance 參數的 category 屬性。這個屬性提供另一種呈現參數清單的方式。無法透過 Flash 使用者介面使用這個功能。

parameter.insertItem()

適用版本
Flash MX 2004。

用法
```
parameter.insertItem(index, name, value, type)
```

參數
index 從零開始的整數索引，指出要在清單、物件或陣列中插入項目的位置。如果這個索引是 0，則項目會被插入清單的開頭。如果索引值大於清單的大小，則新項目會被插入陣列的結尾。

name 字串，指定要插入的項目名稱。這是 object 參數的必要參數。

value 指定插入項目值的字串。

type 字串，指定要插入的項目類型。

傳回值
無。

說明
方法：插入項目到清單、物件或陣列。如果參數是清單、物件或陣列，則 value 屬性是陣列。

範例
下列範例會將 New Value 值插入 labelPlacement 參數中：

```
// Select an instance of a Button component on the Stage.
var parms = fl.getDocumentDOM().selection[0].parameters;
parms[2].insertItem(0, "name", "New Value", "String");
var values = parms[2].value;
for(var prop in values){
    fl.trace("labelPlacement parameter value = " + values[prop].value);
}
```

parameter.listIndex

適用版本
Flash MX 2004。

用法
```
parameter.listIndex
```

說明
屬性：所選取清單項目的值。這個屬性只有在 parameter.valueType 為 "List" 時才有效。

範例
下列範例會為「幻燈片」設定第一個參數，也就是 autoKeyNav 參數。若要將這個參數設定為其中一個可接受的值 (true、false 或 inherit)，請將 parameter.listIndex 設定為清單中的項目索引 (0 代表 true、1 代表 false、2 代表 inherit)。

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;
parms[0].listIndex = 1;
```

parameter.name

適用版本
Flash MX 2004。

用法
```
parameter.name
```

說明
唯讀屬性：指定參數名稱的字串。

範例
下列範例會顯示選取組件的第五個參數名稱：

```
var parms = fl.getDocumentDOM().selection[0].parameters;
fl.trace("name: " + parms[4].name);
```

下列範例會顯示指定螢幕的第五個參數名稱：

```
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters; fl.trace("name: " + parms[4].name);
```

parameter.removeItem()

適用版本
Flash MX 2004。

用法
```
parameter.removeItem(index)
```

參數
index　要從螢幕或組件屬性中移除之項目的整數索引 (從零開始)。

傳回值
無。

說明
方法：移除螢幕或組件參數的清單、物件或陣列類型元素。

範例
下列範例會從組件的 labelPlacement 參數中移除索引 1 的元素：

```
// Select an instance of a Button component on the Stage.
var parms = fl.getDocumentDOM().selection[0].parameters;
var values = parms[2].value;
fl.trace("--Original--");
for(var prop in values){
fl.trace("labelPlacement value = " + values[prop].value);
}
parms[2].removeItem(1);

var newValues = parms[2].value;
fl.trace("--After Removing Item--");
for(var prop in newValues){
fl.trace("labelPlacement value = " + newValues[prop].value);
}
```

下列範例會從螢幕的 autoKeyNav 參數中移除索引 1 的元素：

```
// Open a presentation document.
var parms = fl.getDocumentDOM().screenOutline.screens[1].parameters;
var values = parms[0].value;
fl.trace("--Original--");
for(var prop in values){
fl.trace("autoKeyNav value = " + values[prop].value);
}
parms[0].removeItem(1);

var newValues = parms[0].value;
fl.trace("--After Removing Item--");
for(var prop in newValues){
fl.trace("autoKeyNav value = " + newValues[prop].value);
}
```

parameter.value

適用版本

Flash MX 2004。

用法

```
parameter.value
```

說明

屬性；與「組件檢測器」的「參數」索引標籤、「屬性」檢測器或螢幕「屬性」檢測器的「參數」索引標籤中的「值」欄位相對應。value 屬性的類型是根據參數的 valueType 屬性判斷 (請參閱 parameter.valueType)。

parameter.valueType

適用版本

Flash MX 2004。

用法

```
parameter.valueType
```

說明
唯讀屬性：指定螢幕或組件參數類型的字串。類型可以是下列其中一個值："Default"、"Array"、"Object"、"List"、"String"、"Number"、"Boolean"、"Font Name"、"Color"、"Collection"、"Web Service URL" 或 "Web Service Operation"。

請參閱
parameter.value

parameter.verbose

適用版本
Flash MX 2004。

用法
```
parameter.verbose
```

說明
屬性；指定顯示參數的位置。如果這個屬性的值為 0 (非詳細內容)，則參數只會顯示在 「組件檢測器」中。如果值為 1 (詳細內容)，則參數會顯示在 「組件檢測器」中以及 「屬性」檢測器的 「參數」索引標籤中。

第 32 章 Path 物件

適用版本
Flash MX 2004。

說明
Path 物件定義連續線段 (直線、曲線或兩者)，通常用於建立可擴充工具。下列範例會顯示從 Flash 物件傳回的 Path 物件實體：

```
path = fl.drawingLayer.newPath();
```

請參閱 drawingLayer 物件。

方法摘要
Path 物件可使用的方法如下：

方法	說明
path.addCubicCurve()	將三次方貝茲曲線線段附加至路徑中。
path.addCurve()	將二次方貝茲曲線線段附加至路徑中。
path.addPoint()	將點加入至路徑中。
path.clear()	移除路徑中所有的點。
path.close()	在路徑中第一個點的位置附加一個點，並將路徑延伸到該點來封閉路徑。
path.makeShape()	使用目前的筆畫和填色設定，在「舞台」上建立形狀。
path.newContour()	在路徑中開始新的輪廓。

屬性摘要
Path 物件可使用的屬性如下：

屬性	說明
path.nPts	唯讀；代表路徑中所含點數的整數。

path.addCubicCurve()

適用版本
Flash MX 2004。

用法
```
path.addCubicCurve(xAnchor, yAnchor, x2, y2, x3, y3, x4, y4)
```

參數
xAnchor 浮點數，指定第一個控制點的 x 位置。

yAnchor 浮點數，指定第一個控制點的 y 位置。

x2 浮點數，指定第二個控制點 x 位置。

y2 浮點數，指定第二個控制點的 y 位置。

x3 浮點數，指定第三個控制點的 x 位置。

y3 浮點數，指定第三個控制點的 y 位置。

x4 浮點數，指定第四個控制點的 x 位置。

y4 浮點數，指定第四個控制點的 y 位置。

傳回值
無。

說明
方法；將三次方貝茲曲線線段附加至路徑中。

範例
下列範例會建立新路徑，將新路徑儲存在 myPath 變數中，並將曲線指定到這個路徑：

```
var myPath = fl.drawingLayer.newPath();
myPath.addCubicCurve(0, 0, 10, 20, 20, 20, 30, 0);
```

path.addCurve()

適用版本
Flash MX 2004。

用法
```
path.addCurve(xAnchor, yAnchor, x2, y2, x3, y3)
```

參數
xAnchor 浮點數，指定第一個控制點的 x 位置。

yAnchor 浮點數，指定第一個控制點的 y 位置。

x2 浮點數，指定第二個控制點 x 位置。

y2 浮點數，指定第二個控制點的 y 位置。

x3 浮點數，指定第三個控制點的 x 位置。

y3 浮點數，指定第三個控制點的 y 位置。

傳回值
無。

說明
方法；將二次方貝茲曲線線段附加至路徑中。

範例
下列範例會建立新路徑，將新路徑儲存在 myPath 變數中，並將曲線指定到這個路徑：

```
var myPath = fl.drawingLayer.newPath();
myPath.addCurve(0, 0, 10, 20, 20, 0);
```

path.addPoint()

適用版本
Flash MX 2004。

用法
```
path.addPoint(x, y)
```

參數
x 浮點數，指定點的 x 位置。

y 浮點數，指定點的 y 位置。

傳回值
無。

說明
方法；將點加入至路徑中。

範例
下列範例會建立新路徑，將新路徑儲存在 myPath 變數中，並將新的點指定至這個路徑：

```
var myPath = fl.drawingLayer.newPath();
myPath.addPoint(10, 100);
```

path.clear()

適用版本
Flash MX 2004。

用法
```
path.clear()
```

參數
無。

傳回值
無。

說明
方法；移除路徑中所有的點。

範例
下列範例會從 myPath 變數中儲存的路徑，移除所有的點：

```
var myPath = fl.drawingLayer.newPath();
myPath.clear();
```

path.close()

適用版本
Flash MX 2004。

用法
```
path.close()
```

參數
無。

傳回值
無。

說明
方法：在路徑中第一個點的位置附加一個點，並將路徑延伸到該點來封閉路徑。如果這個路徑沒有點，則不會加入任何點。

範例
下列範例會建立封閉路徑：

```
var myPath = fl.drawingLayer.newPath();
myPath.close();
```

path.makeShape()

適用版本
Flash MX 2004。

用法
```
path.makeShape([bSupressFill [, bSupressStroke]])
```

參數
bSuppressFill Boolean 值，如果設定為 true，形狀便不會套用填色。預設值為 false。這個參數是選擇性參數。

bSupressStroke Boolean 值，如果設定為 true，形狀便不會套用筆畫。預設值為 false。這個參數是選擇性參數。

傳回值
無。

說明
方法：使用目前的筆畫和填色設定，在「舞台」上建立形狀。會在建立形狀之後清除路徑。這個方法有兩個選擇性參數，產生的形狀物件便不會套用填色和筆畫。如果您忽略這些參數，或將它們設定為 false，則會使用填色和筆畫目前的值。

範例
下列範例會使用目前的填色來建立形狀，而不使用筆畫：

```
var myPath = fl.drawingLayer.newPath();
myPath.makeShape(false, true);
```

path.newContour()

適用版本
Flash MX 2004。

用法
```
path.newContour()
```

參數
無。

傳回值
無。

說明
方法：在路徑中開始新的輪廓。

範例
下列範例會建立空心方形：

```
var myPath = fl.drawingLayer.newPath();
myPath.addPoint(0, 0);
myPath.addPoint(0, 30);
myPath.addPoint(30, 30);
myPath.addPoint(30, 0);
myPath.addPoint(0, 0);

myPath.newContour();
myPath.addPoint(10, 10);
myPath.addPoint(10, 20);
myPath.addPoint(20, 20);
myPath.addPoint(20, 10);
myPath.addPoint(10, 10);

myPath.makeShape();
```

path.nPts

適用版本
Flash MX 2004。

用法
```
path.nPts
```

說明

唯讀屬性：代表路徑中所含點數的整數。新路徑包含點數為 0。

範例

下列範例會使用「輸出」面板來顯示 myPath 變數所參照路徑中的點數：

```
var myPath = fl.drawingLayer.newPath();
var numOfPoints = myPath.nPts;
fl.trace("Number of points in the path: " + numOfPoints);
// Displays: Number of points in the path: 0
```

第 33 章 presetItem 物件

適用版本
Flash CS4 Professional。

說明
presetItem 物件代表「移動預設效果」面板 (「視窗 > 移動預設效果」) 中的項目 (預設效果或資料夾)。presetItem 物件陣列是 presetPanel 物件 (presetPanel.items) 的屬性。

presetItem 物件的所有屬性都是唯讀屬性。若要執行如刪除、重新命名或移動項目之類的工作，請使用 presetPanel 物件的方法。

屬性摘要
presetItem 物件可以搭配下列屬性使用：

屬性	說明
presetItem.isDefault	指定項目是隨 Flash 一起安裝的項目，還是已經建立的自訂項目。
presetItem.isFolder	指定「移動預設效果」面板中的項目是預設效果還是資料夾。
presetItem.level	「移動預設效果」面板資料夾結構中項目的階層。
presetItem.name	預設效果或資料夾的名稱，不含路徑資訊。
presetItem.open	指定「移動預設效果」面板中的資料夾目前是否展開。
presetItem.path	「移動預設效果」面板資料夾樹狀結構中的項目路徑和項目名稱。

presetItem.isDefault

適用版本
Flash CS4 Professional。

用法
```
presetItem.isDefault
```

說明
唯讀屬性：Boolean 值，指定項目是隨 Flash 一起安裝的 (true) 還是由您或他人所建立的自訂項目 (false)。如果此值為 true，您可以將它視為「唯讀」項目：無法移動、刪除它，或對它套用任何類似作業。

範例
下列範例會顯示「移動預設效果」面板的內容，並指出項目是否隨 Flash 一起安裝：

```
fl.outputPanel.clear();
var presetItemArray=fl.presetPanel.items;
for (i=0;i<presetItemArray.length; i++){
    var presetItem = presetItemArray[i];
    fl.trace(presetItem.name +", default =" + presetItem.isDefault);
}
```

presetItem.isFolder

適用版本
Flash CS4 Professional。

用法
```
presetItem.isFolder
```

說明
唯讀屬性：Boolean 值，指定 「移動預設效果」面板中的項目是資料夾 (true) 還是預設效果 (false)。

範例
下列範例會示範 「移動預設效果」面板中的第一個項目是資料夾，第二個項目是預設效果：

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].isFolder);
fl.trace(presetItemArray[1].isFolder);
```

presetItem.level

適用版本
Flash CS4 Professional。

用法
```
presetItem.level
```

說明
唯讀屬性：整數，指定項目在 「移動預設效果」面板資料夾結構中的層級。「預設的預設效果」和 「自訂的預設效果」資料夾是層級 0。

範例
下列範例會示範 「移動預設效果」面板中的第一個項目是層級 0，第二個項目是層級 1：

```
var presetItemArray=fl.presetPanel.items;
fl.trace(presetItemArray[0].level);
fl.trace(presetItemArray[1].level);
```

presetItem.name

適用版本
Flash CS4 Professional。

用法
```
presetItem.name
```

說明
唯讀屬性：字串，代表預設效果或資料夾的名稱 (不含路徑資訊)。

範例
請參閱 presetItem.path。

presetItem.open

適用版本
Flash CS4 Professional。

用法
```
presetItem.open
```

說明
唯讀屬性：指定 「移動預設效果」面板中的資料夾目前是 (true) 否 (false) 已展開。

如果項目不是資料夾，這個屬性就是 true。若要判斷項目是資料夾還是預設效果，請使用 presetItem.isFolder。

範例
下列範例會顯示有關 「移動預設效果」面板中的資料夾是展開還是收合的資訊：

```
fl.outputPanel.clear();
var presetItemArray=fl.presetPanel.items;
for (i=0;i<presetItemArray.length; i++){
    var presetItem = presetItemArray[i];
    if (presetItem.isFolder) {
        var status = presetItem.open ? "Open" : "Closed"
        fl.trace(presetItem.level + "-" + presetItem.name +" folder is " + status);
    }
}
```

presetItem.path

適用版本
Flash CS4 Professional。

用法
```
presetItem.path
```

說明
唯讀屬性：字串，代表 「移動預設效果」面板資料夾樹狀結構中的項目路徑和項目名稱。

範例
下列範例會示範 presetItem.name 和 presetItem.path 屬性值之間的差異。

```
fl.outputPanel.clear();
var presetItemArray=fl.presetPanel.items;
for (i=0;i<presetItemArray.length; i++){
    var presetItem = presetItemArray[i];
    fl.trace("Name: " + presetItem.name + "\n" + "Path: " + presetItem.path);
    fl.trace("");
}
```

第 34 章 presetPanel 物件

適用版本
Flash CS4 Professional。

說明
presetPanel 物件代表 「移動預設效果」面板 (「視窗 > 移動預設效果」)。它是 flash 物件 (fl.presetPanel) 的屬性。

方法摘要
presetPanel 物件可以搭配下列方法使用：

方法	說明
presetPanel.addNewItem()	如果 「舞台」上目前選取單一移動補間動畫，會將該移動加入至 「移動預設效果」面板。
presetPanel.applyPreset()	將指定的或目前選取的預設效果套用至 「舞台」上目前選取的項目。
presetPanel.deleteFolder()	從 「移動預設效果」面板的資料夾樹狀結構刪除指定的資料夾及其任何子資料夾。
presetPanel.deleteItem()	從 「移動預設效果」面板刪除指定的預設效果。
presetPanel.expandFolder()	在 「移動預設效果」面板中，展開或收合目前選取的一個或多個資料夾。
presetPanel.exportItem()	將目前選取或指定的預設效果匯出至 XML 檔。
presetPanel.findItemIndex()	傳回整數，代表項目在 「移動預設效果」面板中的索引位置。
presetPanel.getSelectedItems()	傳回 presetItem 物件的陣列，這些物件對應至 「移動預設效果」面板中目前選取的項目。
presetPanel.importItem()	將預設效果從指定的 XML 檔加入至 「移動預設效果」面板。
presetPanel.moveToFolder()	將指定的項目移到指定的資料夾。
presetPanel.newFolder()	在 「移動預設效果」面板的資料夾樹狀結構中建立資料夾。
presetPanel.renameItem()	將目前選取的預設效果或資料夾重新命名為指定的名稱。
presetPanel.selectItem()	選取或取消選取 「移動預設效果」面板中的項目。

屬性摘要
presetPanel 物件可以搭配下列屬性使用：

屬性	說明
presetPanel.items	「移動預設效果」面板中的 presetItem 物件陣列。

presetPanel.addNewItem()

適用版本
Flash CS4 Professional。

用法

```
fl.presetPanel.addNewItem( [namePath] );
```

參數

namePath 字串，指定要增加至 「移動預設效果」面板中的項目路徑和名稱。這個參數是選擇性參數。

傳回值

Boolean 值：如果成功增加項目，會傳回 true；否則便傳回 false。

說明

方法；如果 「舞台」上目前選取單一移動補間動畫，會將該移動以指定名稱加入至 「移動預設效果」面板中的指定資料夾。namePath 中指定的路徑必須已存在於面板。

如果符合 namePath 的預設效果存在，這個方法便無效，而且會傳回 false。

如果您未傳遞值給 namePath，項目便會加到 「自訂的預設效果」資料夾並命名為 「自訂預設效果 n」，其中 n 會在每次您以此方式增加項目時遞增。

範例

假設 「舞台」上已選取單一移動補間動畫，下列程式碼會將名為 Bouncing Ball 的預設效果加入至 「自訂的預設效果」資料夾：

```
fl.presetPanel.addNewItem("Custom Presets/Bouncing Ball");
```

請參閱

presetPanel.newFolder()

presetPanel.applyPreset()

適用版本

Flash CS4 Professional。

用法

```
presetPanel.applyPreset( [presetPath] )
```

參數

presetPath 字串，指定出現在 「移動預設效果」面板中要套用之預設效果的完整路徑和名稱。這是選擇性的參數；如果您未傳遞值，則會套用目前選取的預設效果。

傳回值

Boolean 值：如果成功套用預設效果，會傳回 true，否則便傳回 false。

說明

方法；將指定的或目前選取的預設效果套用至 「舞台」上目前選取的項目。該項目必須是移動補間動畫、元件或可轉換為元件的項目。如果項目是移動補間動畫，它的目前移動會被取代為選取的預設效果，而不會要求使用者確認。

在下列情況下，這個方法會失敗：

- 指定為 presetPath 的路徑不存在。
- 未傳遞值給 presetPath，而且未選取預設效果。

- 未傳遞值給 presetPath，而且選取多個預設效果。
- 「舞台」上選取的項目不是元件，也無法轉換為元件。

範例
下列範例會將 aDribble 預設效果套用至 「舞台」上目前選取的項目：

```
var result = fl.presetPanel.applyPreset("Custom Presets/Bounces/aDribble");
fl.trace(result);
```

presetPanel.deleteFolder()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.deleteFolder( [folderPath])
```

參數
folderPath 字串，指定要從 「移動預設效果」面板刪除的資料夾。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功刪除一或多個資料夾，會傳回 true；否則便傳回 false。

說明
方法：從 「移動預設效果」面板的資料夾樹狀結構刪除指定的資料夾及其任何子資料夾。同時也會刪除資料夾中的任何預設效果。您無法從 「預設的預設效果」資料夾刪除資料夾。

如果您未傳遞值給 folderPath，則會刪除目前選取的任何資料夾。

備註：刪除資料夾時不會要求使用者確認，而且無法還原此動作。

範例
下列程式碼會刪除 「自訂的預設效果」資料夾下名為 Bouncing 的資料夾，同時也會刪除 Bouncing 的任何子資料夾：

```
fl.presetPanel.deleteFolder("Custom Presets/Bouncing");
```

請參閱
presetPanel.deleteItem()

presetPanel.deleteItem()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.deleteItem( [namePath] )
```

參數
namePath 字串，指定要從 「移動預設效果」面板刪除的項目路徑和名稱。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功刪除一或多個項目，會傳回 true；否則便傳回 false。

說明
方法：從 「移動預設效果」面板刪除指定的預設效果。如果您未傳遞值給 namePath，則會刪除目前選取的任何預設效果。您無法從 「預設的預設效果」資料夾刪除項目。

備註：刪除項目時不會要求使用者確認，而且無法還原此動作。

範例
下列程式碼會從 「自訂的預設效果」資料夾刪除名為 aDribble 的預設效果：

```
fl.presetPanel.deleteItem("Custom Presets/aDribble");
```

請參閱
presetPanel.deleteFolder()

presetPanel.expandFolder()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.expandFolder( [bExpand [, bRecurse [, folderPath] ] ] )
```

參數
bExpand Boolean 值，指定展開資料夾 (true) 或收合資料夾 (false)。這是選擇性的參數，預設值為 true。

bRecurse Boolean 值，指定會展開或收合資料夾的子資料夾 (true) 還是不會 (false)。這是選擇性的參數，預設值為 false。

folderPath 字串，指定要展開或收合之資料夾的路徑。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功展開或收合一或多個資料夾，會傳回 true；否則便傳回 false。

說明
方法：在 「移動預設效果」面板中，展開或收合目前選取的一個或多個資料夾。若要展開或收合目前選取以外的其它資料夾，請傳遞值給 folderPath。

範例
下列範例會展開 「自訂的預設效果」資料夾，但不會展開它的子資料夾：

```
fl.presetPanel.expandFolder(true, false, "Custom Presets");
```

下列範例會展開 「自訂的預設效果」資料夾及其所有子資料夾：

```
fl.presetPanel.expandFolder(true, true, "Custom Presets");
```

presetPanel.exportItem()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.exportItem(fileURI [, namePath] )
```

參數
fileURI 字串；指定匯出檔案的路徑或檔案名稱，表示為 file:/// URI。如需詳細資訊，請參閱下面的 「說明」。

namePath 字串，指定要從 「移動預設效果」面板選取的項目路徑和名稱。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功匯出預設效果，會傳回 true，否則便傳回 false。

說明
方法；將目前選取或指定的預設效果匯出至 XML 檔。只能匯出預設效果，如果您嘗試匯出資料夾，此方法會失敗。如果您嘗試覆寫磁碟上的檔案，此方法也會失敗。

如果未指定檔案名稱為 fileURI 的一部分 (也就是說，如果 fileURI 的最後一個字元是斜線 (/))，則會以匯出預設效果的相同名稱來儲存匯出的檔案。如果未指定值給 namePath，則會匯出目前選取的預設效果。請參閱以下範例。

範例
下列範例示範當傳遞不同參數至此方法時會建立哪些檔案，並通知您是否成功建立指定的檔案。執行此範例之前，請選取 「預設的預設效果」資料夾中的 「從左邊飛入」 (fly-in-left) 預設效果，並在磁碟上建立 My Presets 資料夾。

```
//Exports fly-in-left to C:\My Presets\fly-in-left.xml
fl.presetPanel.exportItem("file:///C|/My Presets/");
//Exports fly-in-left to C:\My Presets\myFavoritePreset.xml
fl.presetPanel.exportItem("file:///C|/My Presets/myFavoritePreset.xml");
// Exports the "pulse" preset to C:\My Presets\pulse.xml
fl.presetPanel.exportItem("file:///C|/My Presets/", "Default Presets/pulse");
// Exports the "pulse" preset to C:\My Presets\thePulsePreset.xml
fl.presetPanel.exportItem("file:///C|/My Presets/thePulsePreset.xml", "Default Presets/pulse");
```

請參閱
presetPanel.importItem()

presetPanel.findItemIndex()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.findItemIndex([presetName])
```

參數
presetName 字串，指定要傳回其索引值的預設效果名稱。這個參數是選擇性參數。

傳回值

整數，代表指定的預設效果在 presetPanel.items 陣列中的索引。如果您未傳遞值給 presetName，則會傳回目前指定之預設效果的索引。在下列情況下，這個方法會傳回 -1：

- 未傳遞值給 presetName，而且未選取預設效果。
- 未傳遞值給 presetName，而且選取多個預設效果。
- 傳遞給 presetName 的值未對應於面板中的項目。

說明

方法；傳回整數，代表項目在 「移動預設效果」面板中的索引位置。

範例

下列程式碼會顯示目前選取之預設效果的索引值和完整路徑名稱：

```
// Select one preset in the Motions Preset panel before running this code
var selectedPreset = fl.presetPanel.findItemIndex();
fl.trace(selectedPreset);
fl.trace(fl.presetPanel.items[selectedPreset].path);
```

presetPanel.getSelectedItems()

適用版本

Flash CS4 Professional。

用法

```
presetPanel.getSelectedItems()
```

參數

無。

傳回值

presetItem 物件的陣列。

說明

方法；傳回 presetItem 物件的陣列，這些物件對應至 「移動預設效果」面板中目前選取的項目 (請參閱 presetItem 物件)。陣列中的每個項目代表資料夾或預設效果。

範例

下列程式碼會顯示 「移動預設效果」面板中目前選取之項目的完整路徑名稱：

```
var itemArray = fl.presetPanel.getSelectedItems();
var length =  itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

請參閱

presetPanel.items

presetPanel.importItem()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.importItem(fileURI [,namePath ])
```

參數
fileURI 字串，指定要匯入為 「移動預設效果」面板中之預設效果的 XML 檔，表示為 file:/// URI。

namePath 字串，指定要放置匯入檔案的資料夾以及檔案名稱。這個參數是選擇性參數。

傳回值
Boolean 值：如果檔案成功匯入，會傳回 true，否則便傳回 false。

說明
方法；將預設效果從指定的 XML 檔加入至 「移動預設效果」面板。namePath 中指定的路徑必須已存在於面板。

若要建立可匯入的 XML 檔，請使用 presetPanel.exportItem()。

如果您未傳遞值給 namePath，則會將匯入的預設效果放在「自訂的預設效果」資料夾，並將它指定為與匯入檔案相同的名稱 (不含 XML 副檔名)。

範例
下列範例會將預設效果匯入至 Custom Presets/Pulse 資料夾，並將它命名為 fastPulse。

```
fl.presetPanel.importItem("file:///C|/My Presets/thePulsePreset.xml", "Custom Presets/Pulse/fastPulse");
```

請參閱
presetPanel.exportItem()

presetPanel.items

適用版本
Flash CS4 Professional。

用法
```
presetPanel.items
```

說明
屬性；「移動預設效果」面板中的 presetItem 物件陣列 (請參閱 presetItem 物件)。陣列中的每個項目代表資料夾或預設效果。

範例
下列程式碼會顯示 「移動預設效果」面板中項目的完整路徑名稱：

```
var itemArray = fl.presetPanel.items;
var length =  itemArray.length
for (x=0; x<length; x++) {
    fl.trace(itemArray[x].path);
}
```

請參閱
presetPanel.getSelectedItems()

presetPanel.moveToFolder()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.moveToFolder(folderPath [, namePath] )
```

參數
folderPath 字串，指定「移動預設效果」面板中移動一或多個項目的目標資料夾路徑。

namePath 字串，指定要移動之項目的路徑和名稱。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功移動項目，會傳回 true；否則便傳回 false。

說明
方法：將指定的項目移到指定的資料夾。

如果您傳遞空字串 ("") 給 folderPath，則會將項目移至「自訂的預設效果」資料夾。如果未傳遞值給 namePath，則會移動目前選取的項目。

您無法移動「預設的預設效果」資料夾中的項目。

範例
在下列範例中，目前選取的項目會移至 Custom Presets/Bouncing 資料夾，接著，Fast Bounce 預設效果會移至相同的資料夾：

```
fl.presetPanel.moveToFolder("Custom Presets/Bouncing");
fl.presetPanel.moveToFolder("Custom Presets/Bouncing" , "Custom Presets/Fast Bounce");
```

presetPanel.newFolder()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.newFolder( [folderPath] )
```

參數
folderPath 字串，指定要在 「移動預設效果」面板中增加新資料夾的位置，以及新資料夾的名稱。這個參數是選擇性參數。

傳回值
Boolean 值：如果成功增加資料夾，會傳回 true；否則便傳回 false。

說明
方法：在 「移動預設效果」面板的資料夾樹狀結構中建立資料夾。使用這個方法只能建立一個新資料夾層級。也就是說，如果您傳遞 「Custom Presets/My First Folder/My Second Folder」給 folderPath，「Custom Presets/My First Folder」必須存在於資料夾樹狀結構中。

如果未傳遞值給 folderPath，則會在 「自訂的預設效果」下的第一個層級建立名為 「未命名資料夾 n」的資料夾，其中 n 會在每次以此方式增加資料夾時遞增。

備註：您無法將資料夾增加至 「預設的預設效果」資料夾。

範例
下列範例會在 「自訂的預設效果」資料夾下加入名為 Bouncing 的資料夾：

```
fl.presetPanel.newFolder("Custom Presets/Bouncing");
```

請參閱
presetPanel.addNewItem()

presetPanel.renameItem()

適用版本
Flash CS4 Professional。

用法
```
presetPanel.renameItem(newName)
```

參數
newName 字串，指定預設效果或資料夾的新名稱。

傳回值
Boolean 值：如果成功重新命名預設效果或資料夾，會傳回 true；否則便傳回 false。

說明
方法：將目前選取的預設效果或資料夾重新命名為指定的名稱。只有在選取 「自訂的預設效果」資料夾中的單一預設效果或資料夾時，這個方法才會成功。在下列情況下，這個方法會失敗：

- 未選取任何項目。
- 選取多個項目。
- 選取的項目位於 「預設的預設效果」資料夾。
- 與選取項目相同的位置上，有一個名為 newName 的項目。

範例

下列範例會將 「自訂的預設效果」資料夾中目前選取的預設效果重新命名為 Bounce Faster。

```
var renamed = fl.presetPanel.renameItem("Bounce Faster");
fl.trace(renamed);
```

presetPanel.selectItem()

適用版本

Flash CS4 Professional。

用法

```
presetPanel.selectItem(namePath [, bReplaceCurrentSelection [, bSelect] ])
```

參數

namePath 字串，指定要從 「移動預設效果」面板選取的項目路徑和名稱。

bReplaceCurrentSelection Boolean 值，指定特定項目會取代任何目前的選取範圍 (true) 還是加入至目前的選取範圍 (false)。這是選擇性的參數，預設值為 true。

bSelect Boolean 值，指定選取項目 (true) 或取消選取項目 (false)。這是選擇性的參數，預設值為 true。如果傳遞 false 給 bSelect，則會忽略 bReplaceCurrentSelection 的值。

傳回值

Boolean 值：如果成功選取或取消選取項目，會傳回 true；否則便傳回 false。

說明

方法：選取或取消選取 「移動預設效果」面板中的項目，並選擇性地取代目前選取的任何項目。

範例

下列程式碼會將 「從右邊模糊飛入」 (fly-in-blur-right) 預設效果加入至 「移動預設效果」面板中目前選取的預設效果 (如果有的話)：

```
fl.presetPanel.selectItem("Default Presets/fly-in-blur-right", false);
```

第 35 章 Rectangle 物件

繼承 Element 物件 > Shape 物件 > Rectangle 物件

適用版本
Flash CS3 Professional。

說明
Rectangle 物件是使用「基本矩形」工具繪製的形狀。若要判斷某個項目是否為 Rectangle 物件，請使用 shape.isRectangleObject。

屬性摘要
除了 Shape 物件屬性，Rectangle 物件還可以搭配下列屬性使用。若要設定 Rectangle 物件的屬性，請使用 document.setRectangleObjectProperty()。

屬性	說明
RectangleObject.bottomLeftRadius	唯讀；設定 Rectangle 物件左下角半徑的浮點值。
RectangleObject.bottomRightRadius	唯讀；設定 Rectangle 物件右下角半徑的浮點值。
RectangleObject.lockFlag	唯讀；Boolean 值，判斷矩形的不同角落是否可以具有不同的半徑值。
RectangleObject.topLeftRadius	唯讀；設定 Rectangle 物件所有角半徑或僅設定 Rectangle 物件左上角半徑的浮點值。
RectangleObject.topRightRadius	唯讀；設定 Rectangle 物件右上角半徑的浮點值。

RectangleObject.bottomLeftRadius

適用版本
Flash CS3 Professional。

用法
```
RectangleObject.bottomLeftRadius
```

說明
唯讀屬性；設定 Rectangle 物件左下角半徑的浮點值。如果 RectangleObject.lockFlag 為 true，嘗試設定這個值就沒有任何作用。

若要設定此值，請使用 document.setRectangleObjectProperty()。

請參閱
document.setRectangleObjectProperty()、RectangleObject.bottomRightRadius、RectangleObject.lockFlag、RectangleObject.topLeftRadius、RectangleObject.topRightRadius

RectangleObject.bottomRightRadius

適用版本
Flash CS3 Professional。

用法
```
RectangleObject.bottomRightRadius
```

說明
唯讀屬性；設定 Rectangle 物件右下角半徑的浮點值。如果 RectangleObject.lockFlag 為 true，嘗試設定這個值就沒有任何作用。

若要設定此值，請使用 document.setRectangleObjectProperty()。

請參閱
document.setRectangleObjectProperty()、RectangleObject.bottomLeftRadius、RectangleObject.lockFlag、RectangleObject.topLeftRadius、RectangleObject.topRightRadius

RectangleObject.lockFlag

適用版本
Flash CS3 Professional。

用法
```
RectangleObject.lockFlag
```

說明
唯讀屬性；Boolean 值，判斷矩形的不同角落是否可以具有不同的半徑值。如果這個值為 true，所有角落都具有指派給 RectangleObject.topLeftRadius 的值。如果它是 false，就可以分別設定每個圓角半徑。

若要設定此值，請使用 document.setRectangleObjectProperty()。

請參閱
document.setRectangleObjectProperty()、RectangleObject.bottomLeftRadius、RectangleObject.bottomRightRadius、RectangleObject.topLeftRadius、RectangleObject.topRightRadius

RectangleObject.topLeftRadius

適用版本
Flash CS3 Professional。

用法
```
RectangleObject.topLeftRadius
```

說明
唯讀屬性：浮點值，設定矩形所有角落的半徑 (如果 RectangleObject.lockFlag 為 true) 或僅設定左上角半徑 (如果 RectangleObject.lockFlag 為 false)。

若要設定此值，請使用 document.setRectangleObjectProperty()。

請參閱
document.setRectangleObjectProperty()、RectangleObject.bottomLeftRadius、RectangleObject.bottomRightRadius、RectangleObject.lockFlag、RectangleObject.topRightRadius

RectangleObject.topRightRadius

適用版本
Flash CS3 Professional。

用法
```
RectangleObject.topRightRadius
```

說明
唯讀屬性：設定 Rectangle 物件右上角半徑的浮點值。如果 RectangleObject.lockFlag 為 true，嘗試設定這個值就沒有任何作用。

若要設定此值，請使用 document.setRectangleObjectProperty()。

請參閱
document.setRectangleObjectProperty()、RectangleObject.bottomLeftRadius、RectangleObject.bottomRightRadius、RectangleObject.lockFlag、RectangleObject.topLeftRadius

第 36 章 Shape 物件

繼承 Element 物件 > Shape 物件

適用版本

Flash MX 2004。

說明

Shape 物件為 Element 物件的子類別。相較於繪圖 API，Shape 物件對在「舞台」上處理或建立幾何的控制更精確。這項控制是指令碼建立有用的特效和其它繪圖命令必要的控制 (請參閱 Element 物件)。

用來變更形狀或任何形狀附屬部分的所有 Shape 方法和屬性，都必須放置在 shape.beginEdit() 和 shape.endEdit() 呼叫之間，才能正確運作。

方法摘要

除了 Element 物件方法，Shape 物件還可以搭配下列方法使用：

方法	說明
shape.getCubicSegmentPoints()	傳回定義三次方曲線的點陣列。
shape.beginEdit()	定義編輯工作階段的開始。
shape.deleteEdge()	刪除指定的邊緣。
shape.endEdit()	定義形狀編輯工作階段的結束。

屬性摘要

除了 Element 物件屬性，Shape 物件還可以使用下列屬性：

屬性	說明
shape.contours	唯讀：形狀的 Contour 物件陣列 (請參閱 Contour 物件)。
shape.edges	唯讀：Edge 物件的陣列 (請參閱 Edge 物件)。
shape.isDrawingObject	唯讀：如果是 true，這個形狀會是繪圖物件。
shape.isGroup	唯讀：如果是 true，這個形狀會是群組。
shape.isOvalObject	唯讀，如果是 true，這個形狀會是基本 Oval 物件 (使用「橢圓形」工具建立)。
shape.isRectangleObject	唯讀，如果是 true，這個形狀會是基本 Rectangle 物件 (使用「矩形」工具建立)。
shape.members	目前選取群組中的物件陣列。
shape.numCubicSegments	唯讀：形狀中的三次方線段數目。
shape.vertices	唯讀：Vertex 物件的陣列 (請參閱 Vertex 物件)。

shape.beginEdit()

適用版本
Flash MX 2004。

用法
```
shape.beginEdit()
```

參數
無。

傳回值
無。

說明
方法；定義編輯工作階段的起點。這個方法必須在發出任何命令變更 Shape 物件或其附屬部分之前使用。

範例
下列範例會從目前選取的形狀中，移除邊緣陣列中的第一個邊緣：

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

shape.contours

適用版本
Flash MX 2004。

用法
```
shape.contours
```

說明
唯讀屬性；形狀的 Contour 物件陣列 (請參閱 Contour 物件)。

範例
下列範例會將輪廓陣列中的第一個輪廓儲存於 c 變數，然後將這個輪廓的 HalfEdge 物件儲存於 he 變數：

```
var c = fl.getDocumentDOM().selection[0].contours[0];
var he = c.getHalfEdge();
```

shape.deleteEdge()

適用版本
Flash MX 2004。

用法
```
shape.deleteEdge(index)
```

參數
index 從零開始的索引，指定要從 shape.edges 陣列中刪除的邊緣。這個方法會變更 shape.edges 陣列的長度。

傳回值
無。

說明
方法：刪除指定的邊緣。使用這個方法之前，您必須呼叫 shape.beginEdit()。

範例
下列範例會從目前選取的形狀中，移除邊緣陣列中的第一個邊緣：

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

shape.edges

適用版本
Flash MX 2004。

用法
```
shape.edges
```

說明
唯讀屬性：Edge 物件的陣列 (請參閱 Edge 物件)。

shape.endEdit()

適用版本
Flash MX 2004。

用法
```
shape.endEdit()
```

參數
無。

傳回值
無。

說明
方法：定義形狀的編輯工作階段結尾。對 Shape 物件或其任何附屬部分所做的所有變更，都會套用到形狀。這個方法必須在發出任何命令變更 Shape 物件或其附屬部分之後使用。

範例
下列範例會從目前選取的形狀中，移除邊緣陣列中的第一個邊緣：

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
shape.deleteEdge(0);
shape.endEdit();
```

shape.getCubicSegmentPoints()

適用版本
Flash CS4 Professional。

用法
```
shape.getCubicSegmentPoints(cubicSegmentIndex)
```

參數
cubicSegmentIndex 整數，指定要傳回點的三次方線段。

傳回值
點陣列，針對 Edge 物件定義對應於指定之 cubicSegmentIndex 的三次方曲線 (請參閱 edge.cubicSegmentIndex)。

說明
方法：傳回定義三次方曲線的點陣列。

範例
下列範例會顯示選取範圍第一個邊緣三次方曲線上各點的 x 和 y 值：

```
var elem = fl.getDocumentDOM().selection[0];
var index = elem.edges[0].cubicSegmentIndex;
var cubicPoints = elem.getCubicSegmentPoints(index);
for (i=0; i<cubicPoints.length; i++) {
    fl.trace("index " + i +" x: " + cubicPoints[i].x  + " y: " + cubicPoints[i].y);
}
```

shape.isDrawingObject

適用版本
Flash 8。

用法
```
shape.isDrawingObject
```

說明
唯讀屬性：如果是 true，這個形狀會是繪圖物件。

範例
下列範例會將第一個選取的物件儲存於 sel 變數，然後使用 element.elementType 和 shape.isDrawingObject 屬性來判斷選取的項目是否為繪圖物件：

```
var sel = fl.getDocumentDOM().selection[0];
var shapeDrawingObject = (sel.elementType == "shape") && sel.isDrawingObject;
fl.trace(shapeDrawingObject);
```

請參閱
document.crop()、document.deleteEnvelope()、document.intersect()、document.punch()、document.union()、shape.isGroup

shape.isGroup

適用版本
Flash MX 2004。

用法
```
shape.isGroup
```

說明
唯讀屬性：如果是 true，這個形狀會是群組。群組可以包含不同類型的元素，如文字元素和元件。不過，群組本身會被視為形狀，無論群組包含哪種類型的元素，您都可以使用 shape.isGroup 屬性。

範例
下列範例會將第一個選取的物件儲存於 sel 變數，然後使用 element.elementType 和 shape.isGroup 屬性判斷選取的項目是否為群組：

```
var sel = fl.getDocumentDOM().selection[0];
var shapeGroup = (sel.elementType == "shape") && sel.isGroup;
fl.trace(shapeGroup);
```

請參閱
shape.isDrawingObject

shape.isOvalObject

適用版本
Flash CS3 Professional。

用法
```
shape.isOvalObject
```

說明
唯讀屬性：如果為 true，此形狀就是基本 Oval 物件 (使用 「基本橢圓形」工具建立)。

範例
如果第一個選取的項目是基本 Oval 物件，下列範例就會顯示 "true"；如果不是，它就會顯示 "false"：

```
var sel = fl.getDocumentDOM().selection[0];
fl.trace(sel.isOvalObject);
```

請參閱
shape.isRectangleObject

shape.isRectangleObject

適用版本
Flash CS3 Professional。

用法
```
shape.isRectangleObject
```

說明
唯讀屬性：如果為 true，此形狀就是基本 Rectangle 物件 (使用 「基本矩形」工具建立)。

範例
如果第一個選取的項目是基本 Rectangle 物件，下列範例就會顯示 "true"；如果不是，它就會顯示 "false"：

```
var sel = fl.getDocumentDOM().selection[0];
fl.trace(sel.isRectangleObject);
```

請參閱
shape.isOvalObject

shape.members

適用版本
Flash CS4 Professional。

用法
```
shape.members
```

說明
唯讀屬性：目前選取群組中的物件陣列。只有在 shape.isGroup 的值為 true 時，才可以使用此屬性。shape.members 陣列中不包含群組中的原始形狀。

例如，如果群組包含三個繪圖物件和三個原始形狀，shape.members 陣列會包含三個項目，每個繪圖物件一個項目。如果群組只包含原始形狀，陣列為空白。

範例
下列程式碼會顯示目前選取的陣列中每個繪圖物件的三次方線段數目：

```
var shapesArray = fl.getDocumentDOM().selection[0].members;
for (i=0; i<shapesArray.length; i++) {
    fl.trace(shapesArray[i].numCubicSegments);
}
```

請參閱
shape.isGroup

shape.numCubicSegments

適用版本
Flash CS4 Professional。

用法
```
shape.numCubicSegments
```

說明
唯讀屬性：形狀中的三次方線段數目。

範例
假設已選取方形或矩形形狀，下列程式碼會在「輸出」面板中顯示「4」：

```
var theShape = fl.getDocumentDOM().selection[0];
fl.trace(theShape.numCubicSegments);
```

shape.vertices

適用版本
Flash MX 2004。

用法
```
shape.vertices
```

說明
唯讀屬性：Vertex 物件的陣列 (請參閱 Vertex 物件)。

範例
下列範例會將第一個選取的物件儲存於 someShape 變數，然後在「輸出」面板中顯示這個物件的頂點數：

```
var someShape = fl.getDocumentDOM().selection[0];
fl.trace("The shape has " + someShape.vertices.length + " vertices.");
```

第 37 章 SoundItem 物件

繼承　Item 物件 > SoundItem 物件

適用版本
Flash MX 2004。

說明
SoundItem 物件為 Item 物件的子類別。代表某個用來建立聲音的元件庫項目。請參閱 frame.soundLibraryItem 和 Item 物件。

方法摘要
除了 Item 物件方法，SoundItem 物件還有下列方法：

屬性	說明
soundItem.exportToFile()	將指定項目匯出至 Macintosh 上的 QuickTime 檔或 Windows 上的 WAV 或 QT 檔。

屬性摘要
除了 Item 物件屬性，SoundItem 物件還可以使用下列屬性：

屬性	說明
soundItem.bitRate	字串，指定元件庫中的聲音位元速率。只有 MP3 壓縮類型才能使用。
soundItem.bits	字串，指定元件庫 (具有 ADPCM 壓縮) 中的聲音位元值。
soundItem.compressionType	字串，指定元件庫中的聲音壓縮類型。
soundItem.convertStereoToMono	Boolean 值，只有 MP3 和 Raw 壓縮類型才能使用。
soundItem.fileLastModifiedDate	唯讀；包含十六進位數字的字串，代表在 1970 年 1 月 1 日和原始檔案 (在磁碟上) 匯入至元件庫時的修改日期之間經過的秒數。
soundItem.originalCompressionType	唯讀；字串，指定特定項目是否已匯入為 MP3 檔。
soundItem.quality	字串，指定元件庫中的聲音播放品質。只有 MP3 壓縮類型才能使用。
soundItem.sampleRate	字串，指定音效片段的取樣頻率。
soundItem.sourceFileExists	唯讀；Boolean 值，指定匯入至元件庫中的檔案是否仍然存在於匯入來源位置。
soundItem.sourceFileIsCurrent	唯讀；Boolean 值，指定元件庫項目的檔案修改日期是否與磁碟上已匯入的原始檔案的修改日期相同。
soundItem.sourceFilePath	唯讀；字串，代表已匯入至元件庫中之檔案的路徑和名稱，表示方式為 file:/// URI。
soundItem.useImportedMP3Quality	Boolean 值；如果是 true，就會忽略所有其它屬性，並使用匯入的 MP3 品質。

soundItem.bitRate

適用版本
Flash MX 2004。

用法
```
soundItem.bitRate
```

說明
屬性；字串；指定元件庫中的聲音位元速率。只有 MP3 壓縮類型才能使用這個屬性。可接受的值為 "8 kbps"、"16 kbps"、"20 kbps"、"24 kbps"、"32 kbps"、"48 kbps"、"56 kbps"、"64 kbps"、"80 kbps"、"112 kbps"、"128 kbps" 和 "160 kbps"。以 8 Kbps 或 16 Kbps 匯出的立體聲會轉換成單聲道。如果是其它壓縮類型，這個屬性會是 undefined。

如果您要指定這個屬性的值，請將 soundItem.useImportedMP3Quality 設定為 false。

範例
下列範例會在元件庫中的指定項目具有 MP3 壓縮類型時，將 bitRate 值顯示於 「輸出」面板：

```
alert(fl.getDocumentDOM().library.items[0].bitRate);
```

請參閱
soundItem.compressionType、soundItem.convertStereoToMono

soundItem.bits

適用版本
Flash MX 2004。

用法
```
soundItem.bits
```

說明
屬性；字串；指定元件庫 (具有 ADPCM 壓縮) 中的聲音位元值。可接受的值為 "2 bit"、"3 bit"、"4 bit" 和 "5 bit"。

如果您要指定這個屬性的值，請將 soundItem.useImportedMP3Quality 設定為 false。

範例
下列範例會在元件庫中的目前選取項目具有 ADPCM 壓縮類型時，將位元值顯示於 「輸出」面板：

```
alert(fl.getDocumentDOM().library.items[0].bits);
```

請參閱
soundItem.compressionType

soundItem.compressionType

適用版本
Flash MX 2004。

用法
```
soundItem.compressionType
```

說明
屬性；字串；指定元件庫中的聲音壓縮類型。可接受的值為 "Default"、"ADPCM"、"MP3"、"Raw" 和 "Speech"。

如果您要指定這個屬性的值，請將 soundItem.useImportedMP3Quality 設定為 false。

範例
下列範例會將元件庫中的項目壓縮類型變更為 Raw：

```
fl.getDocumentDOM().library.items[0].compressionType = "Raw";
```

下列範例會將選取的元件庫項目的壓縮類型變更為 Speech：

```
fl.getDocumentDOM().library.getSelectedItems().compressionType = "Speech";
```

請參閱
soundItem.originalCompressionType

soundItem.convertStereoToMono

適用版本
Flash MX 2004。

用法
```
soundItem.convertStereoToMono
```

說明
屬性；Boolean 值；只有 MP3 和 Raw 壓縮類型才能使用。將這個屬性設定為 true，會將立體聲轉換為單聲道；如果是 false 則會維持立體聲。如果是 MP3 壓縮類型，且 soundItem.bitRate 小於 20 Kbps，則會忽略這個屬性並強制為 true (請參閱 soundItem.bitRate)。

如果您要指定這個屬性的值，請將 soundItem.useImportedMP3Quality 設定為 false。

範例
下列範例只會在項目具有 MP3 或 Raw 壓縮類型的情況下，將元件庫中的項目轉換為單聲道：

```
fl.getDocumentDOM().library.items[0].convertStereoToMono = true;
```

請參閱
soundItem.compressionType

soundItem.exportToFile()

適用版本
Flash CS4 Professional。

用法
```
soundItem.exportToFile(fileURI)
```

參數
fileURI 字串；指定匯出檔案的路徑和名稱，表示為 file:/// URI。

傳回值
如果檔案成功匯出，就會傳回 Boolean 值 true，否則便傳回 false。

說明
方法；將指定項目匯出為 Macintosh 上的 QuickTime 檔或 Windows 上的 WAV 或 QT 檔。匯出的 QuickTime 或 QT 檔只包含音效；不會匯出視訊。匯出設定是根據匯出的項目而定的。

範例
假設元件庫中的第一個項目是聲音項目，下列程式碼會將它匯出為 WAV 檔：

```
var soundFileURL = "file:///C|/out.wav";
var libItem = fl.getDocumentDOM().library.items[0];
libItem.exportToFile(soundFileURL);
```

soundItem.fileLastModifiedDate

適用版本
Flash CS4 Professional。

用法
```
soundItem.fileLastModifiedDate
```

說明
唯讀屬性；包含十六進位數字的字串，代表在 1970 年 1 月 1 日和匯入至元件庫之原始檔案修改日期之間經過的秒數。如果檔案已不存在，此值為 「00000000」。

範例
假設元件庫中的第一個項目是聲音項目，下列程式碼會將它顯示為十六進位數字 (如上述)。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

請參閱
soundItem.sourceFileExists、soundItem.sourceFileIsCurrent、soundItem.sourceFilePath、FLfile.getModificationDate()

soundItem.originalCompressionType

適用版本
Flash CS4 Professional。

用法
```
soundItem.originalCompressionType
```

說明
唯讀屬性：字串，指定特定項目是否已匯入為 MP3 檔。這個屬性的可能值 "RAW" 和 "MP3"。

範例
假設元件庫中的第一個項目是聲音項目，下列程式碼會在檔案已匯入為元件庫中的 MP3 檔時顯示 「MP3」，否則顯示「RAW」：

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("Imported compression type = "+ libItem.originalCompressionType);
```

請參閱
soundItem.compressionType

soundItem.quality

適用版本
Flash MX 2004。

用法
```
soundItem.quality
```

說明
屬性；字串；指定元件庫中的聲音播放品質。只有 MP3 壓縮類型才能使用這個屬性。可接受的值為 "Fast"、"Medium" 和 "Best"。

如果您要指定這個屬性的值，請將 soundItem.useImportedMP3Quality 設定為 false。

範例
下列範例會在項目具有 MP3 壓縮類型時，將元件庫中的項目播放品質設定為 Best：

```
fl.getDocumentDOM().library.items[0].quality = "Best";
```

請參閱
soundItem.compressionType

soundItem.sampleRate

適用版本
Flash MX 2004。

用法
```
soundItem.sampleRate
```

說明
屬性；字串；指定音效片段的取樣頻率。只有 ADPCM、Raw 和 Speech 壓縮類型才能使用此屬性。可接受的值為 "5 kHz"、"11 kHz"、"22 kHz" 和 "44 kHz"。

如果您要指定這個屬性的值，請將 soundItem.useImportedMP3Quality 設定為 false。

範例
下列範例會在項目具有 ADPCM、Raw 或 Speech 壓縮類型時，將元件庫中的項目取樣頻率設定為 5 kHz：
```
fl.getDocumentDOM().library.items[0].sampleRate = "5 kHz";
```

請參閱
soundItem.compressionType

soundItem.sourceFileExists

適用版本
Flash CS4 Professional。

用法
```
soundItem.sourceFileExists
```

說明
唯讀屬性：如果匯入至元件庫中的檔案仍然位於匯入來源位置，會傳回 Boolean 值 true，否則便傳回 false。

範例
假設元件庫中的第一個項目是聲音項目，下列程式碼會在匯入至元件庫中的檔案仍然存在時顯示 "true"。
```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

請參閱
soundItem.sourceFileIsCurrent、soundItem.sourceFilePath

soundItem.sourceFileIsCurrent

適用版本
Flash CS4 Professional。

用法
```
soundItem.sourceFileIsCurrent
```

說明
唯讀屬性：如果元件庫項目的檔案修改日期和磁碟上已匯入的原始檔案的修改日期相同，會傳回 Boolean 值 true，否則便傳回 false。

範例
假設元件庫中的第一個項目是聲音項目，下列程式碼會在匯入的原始檔案自匯入後即未曾在磁碟上修改時顯示 "true"。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

請參閱
soundItem.fileLastModifiedDate、soundItem.sourceFilePath

soundItem.sourceFilePath

適用版本
Flash CS4 Professional。

用法
```
soundItem.sourceFilePath
```

說明
唯讀屬性：字串，代表已匯入至元件庫中之檔案的路徑和名稱，表示為 file:/// URI。

範例
下列範例會顯示元件庫中任何類型為 "sound" 的項目之名稱和來源檔案路徑：

```
for (idx in fl.getDocumentDOM().library.items) {
if (fl.getDocumentDOM().library.items[idx].itemType == "sound") {
    var myItem = fl.getDocumentDOM().library.items[idx];
    fl.trace(myItem.name + " source is " + myItem.sourceFilePath);
    }
}
```

請參閱
soundItem.sourceFileExists

soundItem.useImportedMP3Quality

適用版本
Flash MX 2004。

用法
```
soundItem.useImportedMP3Quality
```

說明
屬性；Boolean 值。如果是 true，就會忽略所有其它屬性，並使用匯入的 MP3 品質。

範例

下列範例會設定元件庫中的項目使用匯入的 MP3 品質：

```
fl.getDocumentDOM().library.items[0].useImportedMP3Quality = true;
```

請參閱

soundItem.compressionType

第 38 章 Stroke 物件

適用版本
Flash MX 2004。

說明
Stroke 物件包含筆畫的所有設定 (包括自訂設定)。這個物件代表 「屬性」檢測器中包含的資訊。將 Stroke 物件和 document.setCustomStroke() 方法搭配使用，可以讓您變更 「工具」面板、「屬性」檢測器和目前選取範圍的筆畫設定。您也可以使用 document.getCustomStroke() 方法，取得 「工具」面板和 「屬性」檢測器的筆畫設定，或目前選取範圍的筆畫設定。

這個物件一定具有下列四個屬性：style、thickness、color 和 breakAtCorners (在 Flash CS3 中，breakAtCorners 屬性已不建議使用，請改用 stroke.joinType)。其它屬性也可以根據 stroke.style 屬性的值進行設定。

屬性摘要
Stroke 物件可使用的屬性如下：

屬性	說明
stroke.breakAtCorners	Boolean 值，與 「筆畫樣式」對話方塊中的 「銳角」設定相同。
stroke.capType	字串；指定筆畫的端點類型。
stroke.color	字串，代表筆畫顏色的十六進位值或整數。
stroke.curve	字串；指定筆畫的花紋類型。
stroke.dash1	整數；指定虛線中實線部分的長度。
stroke.dash2	整數；指定虛線中空白部分的長度。
stroke.density	字串；指定點刻線的密度。
stroke.dotSize	字串；指定點刻線的圓點大小。
stroke.dotSpace	整數，指定虛線中的點間隔。
stroke.hatchThickness	字串；指定花紋線的粗細。
stroke.jiggle	字串；指定花紋線的微動屬性。
stroke.joinType	字串；指定筆畫的加入類型。
stroke.length	字串；指定花紋線的長度。
stroke.miterLimit	浮點值；指定斜接面 (miter) 端點將遭線段截斷的最小限制角度。
stroke.pattern	字串；指定鋸齒線的圖樣。
stroke.rotate	字串；指定花紋線的旋轉。
stroke.scaleType	字串，指定套用至筆畫的縮放類型。
stroke.shapeFill	Fill 物件，代表筆畫填色設定。
stroke.space	字串；指定花紋線的間距。
stroke.strokeHinting	Boolean 值；指定是否在筆畫上設定筆畫提示。
stroke.style	字串；說明筆畫樣式。

屬性	說明
stroke.thickness	整數；指定筆畫大小。
stroke.variation	字串；指定點刻線的變化。
stroke.waveHeight	字串；指定鋸齒線的波形高度。
stroke.waveLength	字串；指定鋸齒線的波形長度。

stroke.breakAtCorners

適用版本

Flash MX 2004。不建議在 Flash CS3 中使用，請改用 stroke.joinType。

用法

```
stroke.breakAtCorners
```

說明

屬性；Boolean 值。這個屬性與自訂「筆畫樣式」對話方塊中的「銳角」設定相同。

範例

下列範例會將 breakAtCorners 屬性設定為 true：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.breakAtCorners = true;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.capType

適用版本

Flash 8。

用法

```
stroke.capType
```

說明

屬性；字串；指定筆畫的端點類型。可接受的值為 "none"、"round" 和 "square"。

範例

下列範例會將筆畫的端點類型設定為 round：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.capType = "round";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.color

適用版本
Flash MX 2004。在 Flash 8 和更新版本中，這個屬性已不建議使用，請改用 stroke.shapeFill.color。

用法
```
stroke.color
```

說明
屬性；筆畫的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

範例
下列範例會設定筆畫顏色：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.color = "#000000";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

請參閱
stroke.shapeFill

stroke.curve

適用版本
Flash MX 2004。

用法
```
stroke.curve
```

說明
屬性；字串，指定筆畫的花紋類型。只有在 stroke.style 屬性設定為 "hatched" 時，才能設定這個屬性 (請參閱 stroke.style)。可接受的值為 "straight"、"slight curve"、"medium curve" 和 "very curved"。

範例
下列範例會對具有 hatched 樣式的筆畫，設定其曲線屬性和其它屬性：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dash1

適用版本
Flash MX 2004。

用法
```
stroke.dash1
```

說明
屬性；整數；指定虛線中實線部分的長度。只有在 stroke.style 屬性設定為 dashed 時，才能使用這個屬性 (請參閱 stroke.style)。

範例
下列範例會為 dashed 筆畫樣式，設定 dash1 和 dash2 屬性：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "dashed";
myStroke.dash1 = 1;
myStroke.dash2 = 2;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dash2

適用版本
Flash MX 2004。

用法
```
stroke.dash2
```

說明
屬性；整數，指定虛線中空白部分的長度。只有在 stroke.style 屬性設定為 dashed 時，才能使用這個屬性 (請參閱 stroke.style)。

範例
請參閱 stroke.dash1。

stroke.density

適用版本
Flash MX 2004。

用法
```
stroke.density
```

說明
屬性；字串；指定點刻線的密度。只有在 stroke.style 屬性設定為 stipple 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "very dense"、"dense"、"sparse" 和 "very sparse"。

範例

下列範例會將 stipple 筆畫樣式的 density 屬性設定為 sparse：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dotSize

適用版本

Flash MX 2004。

用法

```
stroke.dotSize
```

說明

屬性；字串；指定點刻線的圓點大小。只有在 stroke.style 屬性設定為 stipple 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "tiny"、"small"、"medium" 和 "large"。

下列範例會將 stipple 筆畫樣式的 dotSize 屬性設定為 tiny：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.dotsize = "tiny";
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.dotSpace

適用版本

Flash MX 2004。

用法

```
stroke.dotSpace
```

說明

屬性；整數；指定虛線中的點間隔。只有在 stroke.style 屬性設定為 dotted 時，才能使用這個屬性。請參閱 stroke.style。

範例

下列範例會將 dotted 筆畫樣式的 dotSpace 屬性設定為 3：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "dotted";
myStroke.dotSpace= 3;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.hatchThickness

適用版本
Flash MX 2004。

用法
```
stroke.hatchThickness
```

說明
屬性；字串；指定花紋線的粗細。只有在 stroke.style 屬性設定為 hatched 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "hairline"、"thin"、"medium" 和 "thick"。

範例
下列範例會將 hatched 筆畫樣式的 hatchThickness 屬性，設定為 thin：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.jiggle

適用版本
Flash MX 2004。

用法
```
stroke.jiggle
```

說明
屬性；字串；指定花紋線的微動屬性。只有在 stroke.style 屬性設定為 hatched 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "none"、"bounce"、"loose" 和 "wild"。

範例
下列範例會將 hatched 筆畫樣式的 jiggle 屬性，設定為 wild：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.joinType

適用版本
Flash 8。

用法
```
stroke.joinType
```

說明
屬性；字串；指定筆畫的加入類型。可接受的值為 "miter"、"round" 和 "bevel"。

請參閱
stroke.capType

stroke.length

適用版本
Flash MX 2004。

用法
```
stroke.length
```

說明
屬性；字串；指定花紋線的長度。只有在 stroke.style 屬性設定為 hatched 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "equal"、"slight variation"、"medium variation" 及 "random"。("random" 值實際上對應至 "medium variation"。)

範例
下列範例會將 hatched 筆畫樣式的 length 屬性，設定為 slight：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight variation";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.miterLimit

適用版本
Flash 8。

用法
```
stroke.miterLimit
```

說明
屬性；浮點值；指定斜接面 (miter) 端點將遭線段截斷的最小限制角度。也就是說，只有在斜接面角度大於 miterLimit 的值時，斜接面才會遭截斷。

範例
下列範例會將筆畫設定的斜接面限制設定為 3。如果斜接面的角度大於 3，則斜接面會遭截斷。

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.miterLimit = 3;
var myStroke = fl.getDocumentDOM().setCustomStroke();
```

stroke.pattern

適用版本
Flash MX 2004。

用法
```
stroke.pattern
```

說明
屬性；字串；指定鋸齒線的圖樣。只有在 stroke.style 屬性設定為 ragged 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "solid"、"simple"、"random"、"dotted"、"random dotted"、"triple dotted" 和 "random triple dotted"。

範例
下列範例會將 ragged 筆畫樣式的 pattern 屬性，設定為 random：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
myStroke.pattern = "random";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.rotate

適用版本
Flash MX 2004。

用法
```
stroke.rotate
```

說明
屬性；字串；指定花紋線的旋轉。只有在 stroke.style 屬性設定為 hatched 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "none"、"slight"、"medium" 和 "free"。

範例
下列範例會將 hatched 筆畫樣式的 rotate 屬性設定為 free：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
```

stroke.scaleType

適用版本

Flash 8。

用法

```
stroke.scaleType
```

說明

屬性；字串；指定套用至筆畫的縮放類型。可接受的值為 "normal"、"horizontal"、"vertical" 和 "none"。

範例

下列範例會將筆畫的縮放類型設定為 horizontal：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.scaleType = "horizontal";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.shapeFill

適用版本

Flash 8。

用法

```
stroke.shapeFill
```

說明

屬性；Fill 物件；代表筆畫的填色設定。

範例

下列範例會指定填色設定，並將這些設定套用至筆畫：

```
var fill = fl.getDocumentDOM().getCustomFill();
fill.linearGradient = true;
fill.colorArray = [ 00ff00, ff0000, fffff ];
var stroke = fl.getDocumentDOM().getCustomStroke();
stroke.shapeFill = fill;
fl.getDocumentDOM().setCustomStroke(stroke);
```

stroke.space

適用版本
Flash MX 2004。

用法
```
stroke.space
```

說明
屬性；字串；指定花紋線的間距。只有在 stroke.style 屬性設定為 hatched 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "very close"、"close"、"distant" 和 "very distant"。

範例
下列範例會將 hatched 筆畫樣式的 space 屬性，設定為 close：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "hatched";
myStroke.curve = "straight";
myStroke.space = "close";
myStroke.jiggle = "wild";
myStroke.rotate = "free";
myStroke.length = "slight";
myStroke.hatchThickness = "thin";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.strokeHinting

適用版本
Flash 8。

用法
```
stroke.strokeHinting
```

說明
屬性；Boolean 值；指定是否在筆畫上設定筆畫提示。

範例
下列範例會啟用筆畫的筆畫提示：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.strokeHinting = true;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.style

適用版本
Flash MX 2004。

用法
```
stroke.style
```

說明
屬性；字串；說明筆畫樣式。可接受的值為 "noStroke"、"solid"、"dashed"、"dotted"、"ragged"、"stipple" 和 "hatched"。其中某些值需要設定 Stroke 物件的其它屬性，如下列清單所示：

- 如果值是 "solid" 或 "noStroke"，表示沒有其它屬性。
- 如果值是 "dashed"，表示另外有兩個屬性：dash1 和 dash2。
- 如果值是 "dotted"，表示另外有一個屬性：dotSpace。
- 如果值是 "ragged"，表示另外有三個屬性：pattern、waveHeight 和 waveLength。
- 如果值是 "stipple"，表示另外有三個屬性：dotSize、variation 和 density。
- 如果值是 "hatched"，表示另外有六個屬性：hatchThickness、space、jiggle、rotate、curve 和 length。

範例
下列範例會將筆畫樣式設定為 ragged：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.thickness

適用版本
Flash MX 2004。

用法
```
stroke.thickness
```

說明
屬性；整數；指定筆畫大小。

範例
下列範例會將筆畫的 thickness 屬性，設定為值 2：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.thickness = 2;
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.variation

適用版本
Flash MX 2004。

用法
```
stroke.variation
```

說明

屬性；字串；指定點刻線的變化。只有在 stroke.style 屬性設定為 stipple 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "one size"、"small variation"、"varied sizes" 和 "random sizes"。

範例

下列範例會將 stipple 筆畫樣式的 variation 屬性，設定為 random sizes：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "stipple";
myStroke.dotSpace= 3;
myStroke.variation = "random sizes";
myStroke.density = "sparse";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveHeight

適用版本

Flash MX 2004。

用法

```
stroke.waveHeight
```

說明

屬性；字串；指定鋸齒線的波形高度。只有在 stroke.style 屬性設定為 ragged 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "flat"、"wavy"、"very wavy" 和 "wild"。

範例

下列範例會將 ragged 筆畫樣式的 waveHeight 屬性，設定為 flat：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
myStroke.pattern = "random";
myStroke.waveHeight = "flat";
myStroke.waveLength = "short";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

stroke.waveLength

適用版本

Flash MX 2004。

用法

```
stroke.waveLength
```

說明

屬性；字串，指定鋸齒線的波形長度。只有在 stroke.style 屬性設定為 ragged 時，才能使用這個屬性 (請參閱 stroke.style)。可接受的值為 "very short"、"short"、"medium" 和 "long"。

範例

下列範例會將 ragged 筆畫樣式的 waveLength 屬性，設定為 short：

```
var myStroke = fl.getDocumentDOM().getCustomStroke();
myStroke.style = "ragged";
myStroke.pattern = "random";
myStroke.waveHeight = 'flat";
myStroke.waveLength = "short";
fl.getDocumentDOM().setCustomStroke(myStroke);
```

第 39 章 swfPanel 物件

適用版本
Flash CS4 Professional。

說明
swfPanel 物件代表「視窗 SWF」面板。「視窗 SWF」面板是實作可從 Flash 編寫環境執行之應用程式的 SWF 檔案；這些面板位於「視窗 > 其他面板」選單中。根據預設，「視窗 SWF」面板會儲存於 Configuration 資料夾的子資料夾中 (請參閱第 2 頁 「儲存 JSFL 檔」)。例如，在 Windows XP 中，資料夾位於 < 開機磁碟 >\Documents and Settings\< 使用者 >\Local Settings\Application Data\Adobe\Flash CS4\< 語言 >\Configuration\WindowSWF 中。如需使用樣本 「視窗 SWF」面板，請參閱第 12 頁 「轉換成向量圖樣本面版」。已註冊的 「視窗 SWF」面板陣列會儲存於 fl.swfPanels 屬性中。

方法摘要
swfPanel 物件可以搭配下列方法使用：

方法	說明
swfPanel.call()	與 ActionScript ExternalInterface.addCallback() 和 MMExecute() 方法一起運作，以便從編寫環境中與 SWF 面板通訊。
第 384 頁 「swfPanel.setFocus()」	將鍵盤焦點設定至指定的 SWF 面板。

屬性摘要
swfPanel 物件可以搭配下列屬性使用：

屬性	說明
swfPanel.name	唯讀；字串，代表指定 「視窗 SWF」面板的名稱。
swfPanel.path	唯讀；字串，代表指定 「視窗 SWF」面板中所使用 SWF 檔的路徑。

swfPanel.call()

適用版本
Flash CS4 Professional。

用法
```
swfPanel.call(request)
```

參數
request 要傳遞至函數的參數 (請參閱下面的 「說明」和 「範例」)。

傳回值
null 或是函數呼叫所傳回的字串。函數結果可能是空字串。

說明
方法；與 ActionScript ExternalInterface.addCallback() 和 MMExecute() 方法一起運作，以便從編寫環境中與 SWF 面板通訊。

範例

下列範例說明如何使用 ActionScript 和 JavaScript 程式碼建立 「視窗 SWF」面板，以及如何從編寫環境中與此面板通訊。

1 建立 ActionScript 3.0 FLA 檔、將顏色設定為淺灰色，以及將大小設定為 400 像素寬和 250 像素高。

2 將動態文字方塊置於 「舞台」中央、將其實體名稱設定為 myTextField，以及在文字方塊中輸入 「Status」一字。

3 設定其它文字方塊屬性，您可以參考下列方式：

- 置中對齊
- 355 像素寬和 46 像素高
- Times New Roman 字體、28 點、紅色

4 請加入下列 ActionScript 程式碼：

```
// Here's the callback function to be called from JSAPI
function callMeFromJavascript(arg:String):void
{
    try {
        var name:String = String(arg);
        myTextField.text = name;
    } catch (e:Error) {
    }
}

// Expose the callback function as "callMySWF"
ExternalInterface.addCallback("callMySWF", callMeFromJavascript);

// run the JSAPI to wire up the callback
MMExecute("fl.runScript( fl.configURI + \"WindowSWF/fileOp.jsfl\" );");

MMExecute("fl.trace(\"AS3 File Status Panel Initialized\");");
```

5 將檔案儲存為 fileStatus.fla，然後使用預設的發佈設定來發佈 SWF 檔。

6 關閉 Flash。

7 將 fileStatus.swf 檔複製到 WindowSWF 資料夾，後者為 Configuration 資料夾的子資料夾 (請參閱第 2 頁「儲存 JSFL 檔」)。例如，在 Windows XP 中，資料夾位於 開機磁碟 \Documents and Settings\< 使用者 >\Local Settings\Application Data\Adobe\Flash CS4\< 語言 >\Configuration\WindowSWF 中。

8 啟動 Flash。

9 使用下列程式碼建立 JSFL 檔：

```
function callMyPanel(panelName, arg)
{
    if(fl.swfPanels.length > 0){
        for(x = 0; x < fl.swfPanels.length; x++){
            // look for a SWF panel of the specified name, then call the specified AS3 function
            // in this example, the panel is named "test" and the AS3 callback is "callMySWF"
            if(fl.swfPanels[x].name == panelName) // name busted?
            {
                fl.swfPanels[x].call("callMySWF",arg);
                break;
            }
        }
    }
    else
        fl.trace("no panels");
}


// define the various handlers for events
documentClosedHandler = function () { callMyPanel("fileStatus", "Document Closed");};
fl.addEventListener("documentClosed", documentClosedHandler );

var dater = "New Document";
documentNewHandler = function () { callMyPanel("fileStatus", dater );};
fl.addEventListener("documentNew", documentNewHandler );

documentOpenedHandler = function () { callMyPanel("fileStatus", "Document Opened");};
fl.addEventListener("documentOpened", documentOpenedHandler );
```

10 以名稱 fileOp.jsfl，將 JSFL 檔儲存在與 SWF 檔相同的目錄中。

11 選擇「視窗 > 其他面板 > fileStatus」。

現在，當您建立、開啟和關閉 FLA 檔時，「fileStatus」面板就會顯示指出您所執行之動作的訊息。

swfPanel.name

適用版本

Flash CS4 Professional。

用法

```
swfPanel.name
```

說明

唯讀屬性；字串；代表指定視窗 SWF 面板的名稱。

範例

下列程式碼會在「輸出」面板中顯示第一個註冊視窗 SWF 面板的名稱：

```
fl.trace(fl.swfPanels[0].name);
```

請參閱

swfPanel.path、fl.swfPanels

swfPanel.path

適用版本
Flash CS4 Professional。

用法
```
swfPanel.path
```

說明
唯讀屬性；字串；代表指定視窗 SWF 面板中所使用 SWF 檔的路徑。

範例
下列程式碼會在 「輸出」面板中顯示第一個註冊視窗 SWF 面板中所使用 SWF 檔的路徑：

```
fl.trace(fl.swfPanels[0].path);
```

請參閱
swfPanel.name、fl.swfPanels

swfPanel.setFocus()

適用版本
Flash CS5.5 Professional。

用法
```
swfPanel.setFocus()
```

說明
方法：將鍵盤焦點設定至指定的 SWF 面板。

範例
下列程式碼會將焦點設定至名為 「專案」的 SWF 面板：

在執行這個命令之前，請先依照下列步驟執行：

1 移除塢接 「專案」面板，使它成為浮動面板。

2 從 「專案」面板開啟 「建立檔案」對話方塊，然後按一下 「舞台」。

3 按幾次 Tab 鍵，確認 「專案」面板沒有焦點。

4 從 「命令」功能表執行下列執令碼 (將包含下列程式碼的 JSFL 檔案放在 user/config/Commands 目錄底下)：

5 按下 Tab 鍵。您應該會在 「建立檔案」對話方塊的其中一個文字欄位中看到插入游標。

```
flash.getSwfPanel("Project").setFocus();
```

請參閱
swfPanel.name、fl.swfPanels

第 40 章 SymbolInstance 物件

繼承　Element 物件 > Instance 物件 > SymbolInstance 物件

適用版本
Flash MX 2004。

說明
SymbolInstance 是 Instance 物件的子類別，代表影格中的元件 (請參閱 Instance 物件」)。

屬性摘要
除了 Instance 物件屬性，SymbolInstance 物件還有下列屬性：

屬性	說明
symbolInstance.accName	字串；等於 「輔助功能」面板中的 「名稱」欄位。
symbolInstance.actionScript	字串；指定要指定給元件的動作。
symbolInstance.backgroundColor	選取不透明時，用來指定邊緣調合顏色的字串。
symbolInstance.bitmapRenderMode	用來指定元件實體之顯示類型的字串。
symbolInstance.blendMode	字串；指定要套用到影片片段元件的混合模式。
symbolInstance.buttonTracking	字串；設定屬性與 「屬性」檢測器中的 「按鈕形式」或 「選單項目形式」彈出式選單相同，限用於按鈕元件。
symbolInstance.cacheAsBitmap	指定執行階段點陣圖快取是否啟用的 Boolean 值。
symbolInstance.colorAlphaAmount	整數；為實體顏色轉換的一部分，用來指定 「進階效果」的 Alpha 設定；等於使用 「屬性」檢測器中的 「顏色 > 進階」設定，並調整對話方塊右方的控制項。
symbolInstance.colorAlphaPercent	整數；指定實體顏色轉換的一部分；等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定 (對話方塊左方的百分比控制項)。
symbolInstance.colorBlueAmount	整數；為實體顏色轉換的一部分；等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定。
symbolInstance.colorBluePercent	整數；為實體顏色轉換的一部分；等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定 (對話方塊左邊的百分比控制項)。
symbolInstance.colorGreenAmount	整數；為實體顏色轉換的一部分；等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定。允許的值從 -255 到 255。
symbolInstance.colorGreenPercent	實體顏色轉換的一部分；等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定 (對話方塊左邊的百分比控制項)。
symbolInstance.colorMode	字串；根據元件 「屬性」檢測器中的 「顏色」彈出式選單指定顏色模式。
symbolInstance.colorRedAmount	整數；為實體顏色轉換的一部分；等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定。
symbolInstance.colorRedPercent	實體顏色轉換的一部分；等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定 (對話方塊左邊的百分比控制項)。
symbolInstance.description	字串；等於 「輔助功能」面板的 「說明」欄位。
symbolInstance.filters	Filter 物件的陣列 (請參閱 Filter 物件)。

屬性	說明
symbolInstance.firstFrame	從零開始的整數；指定圖像時間軸中出現的第一個影格。
symbolInstance.forceSimple	Boolean 值；啟用和停用物件子系的存取；等於 「輔助功能」面板 「讓子物件支援輔助功能」設定的反轉邏輯。
symbolInstance.loop	字串；用於圖像元件，設定屬性與 「屬性」檢測器中的 「重複」彈出式選單相同。
symbolInstance.shortcut	字串；等於與元件相關的快速鍵；等於 「輔助功能」面板中的 「快速鍵」欄位。
symbolInstance.silent	Boolean 值；啟用和停用物件的存取；等於 「輔助功能」面板 「讓物件支援輔助功能」設定的反轉邏輯。
symbolInstance.symbolType	字串；指定元件的類型；等於 「建立新元件」和 「轉換成元件」中的 「行為指令」值。
symbolInstance.tabIndex	整數；等於 「輔助功能」面板中的 「定位鍵索引」欄位。
第 396 頁 「symbolInstance.usesBackgroundColor」	用來指定顯示格式的布林值。
第 396 頁 「symbolInstance.visible」	用來指定顯示或隱藏實體的布林值。

symbolInstance.accName

適用版本
Flash MX 2004。

用法
```
symbolInstance.accName
```

說明
屬性；字串；等於 「輔助功能」面板中的 「名稱」欄位。螢幕朗讀程式將大聲唸出物件名稱來識別物件。圖像元件無法使用這個屬性。

範例
下列範例會將物件的 「輔助功能」面板名稱值儲存於 theName 變數：

```
var theName = fl.getDocumentDOM().selection[0].accName;
```

下列範例會將物件的 「輔助功能」面板名稱值設定為 Home Button：

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

symbolInstance.actionScript

適用版本
Flash MX 2004。

用法
```
symbolInstance.actionScript
```

說明

屬性；字串；指定要指定給元件的動作。只能套用至影片片段和按鈕實體。如果是圖像元件實體，這個值就會傳回 undefined。

範例

下列範例會將 onClipEvent 動作指定給時間軸中，第一個圖層的第一個影格的第一個項目：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].actionScript
    = "onClipEvent(enterFrame) {trace('movie clip enterFrame');}";
```

symbolInstance.backgroundColor

適用版本

Flash CS5.5 Professional。

用法

```
symbolInstance.backgroundColor
```

說明

屬性；為實體選取 24 位元模式時，用來指定邊緣調合顏色的字串。這是一個十六進位 #rrggbb 格式的字串或包含值的整數。

範例

下列範例會指定黑色作為元件實體的背景顏色：

```
var bitmapInstance = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0];
bitmapInstance.backgroundColor = "#000000";
```

symbolInstance.bitmapRenderMode

適用版本

Flash CS5.5 Professional。

用法

```
symbolInstance.bitmapRenderMode
```

說明

屬性；設定元件顯示類型的字串。

可接受的值包括：

- “none”
- "cache" - 將元件設定為在執行階段由 Flash Player 快取為點陣圖。
- "export" - 將元件設定為在編譯 SWF 時匯出為點陣圖。

舊版 第 389 頁「symbolInstance.cacheAsBitmap」屬性類似於此屬性，但提供的選擇較少，因為它是布林。未來可能不建議使用 cacheAsBitmap 屬性，因此，使用者應切換至此新屬性。布林 cacheAsBitmap 屬性中的 true/false 選項，等於此新屬性的 "cache" / "none" 值。

範例
下列範例是將元件的 bitmapRenderMode 指定為 "export"：

```
var symbol = fl.getDocumentDOM().selection[0];
fl.trace(symbol.bitmapRenderMode);
symbol.bitmapRenderMode = "export";
```

symbolInstance.blendMode

適用版本
Flash 8。

用法
```
symbolInstance.blendMode
```

說明
屬性；字串；指定要套用到影片片段元件的混合模式。可接受的值為 "normal"、"layer"、"multiply"、"screen"、"overlay"、"hardlight"、"lighten"、"darken"、"difference"、"add"、"subtract"、"invert"、"alpha" 和 "erase"。

範例
下列範例會將第一個階層上，第一個影格中第一個影片片段元件的混合模式設定為 add：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].blendMode = "add";
```

請參閱
document.setBlendMode()

symbolInstance.buttonTracking

適用版本
Flash MX 2004。

用法
```
symbolInstance.buttonTracking
```

說明
屬性；字串；設定屬性與 「屬性」檢測器中的 「按鈕形式」或 「選單項目形式」彈出式選單相同，限用於按鈕元件。其它類型的元件會忽略這個屬性。可接受的值為 "button" 或 "menu"。

範例
下列範例會在元件為按鈕時，將時間軸中第一個圖層的第一個影格的第一個元件，設定為 「選單項目形式」：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].buttonTracking = "menu";
```

symbolInstance.cacheAsBitmap

適用版本
Flash 8。

用法
```
symbolInstance.cacheAsBitmap
```

說明
屬性；指定執行階段點陣圖快取是否啟用的 Boolean 值。

備註：自 Flash Professional CS5.5 起，使用者應切換為使用 第 387 頁「symbolInstance.bitmapRenderMode」屬性，而非此屬性。

範例
下列範例會啟用第一個圖層上，第一個影格中第一個元素的執行階段點陣圖快取：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].cacheAsBitmap = true;
```

symbolInstance.colorAlphaAmount

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorAlphaAmount
```

說明
屬性；整數；為實體顏色轉換的一部分，用來指定「進階效果」的 Alpha 設定。這個屬性等於使用「屬性」檢測器中的「顏色 > 進階」設定，並調整對話方塊右邊的控制項。這個值會以常數值來增減著色和 Alpha 值。這個值會加入至目前值。這個屬性與 symbolInstance.colorAlphaPercent 搭配使用時會特別有用。允許的值為 -255 到 255。

範例
下列範例會從選取元件實體的 Alpha 設定中減去 100：

```
fl.getDocumentDOM().selection[0].colorAlphaAmount = -100;
```

symbolInstance.colorAlphaPercent

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorAlphaPercent
```

說明
屬性；整數；指定實體顏色轉換的一部分。這個屬性等於使用實體「屬性」檢測器中的「顏色 > 進階」設定 (對話方塊左邊的百分比控制項)。這個值會將著色和 Alpha 值變更為指定的百分比。允許的值為 -100 到 100。請參閱 symbolInstance.colorAlphaAmount。

範例
下列範例會將選取元件實體的 colorAlphaPercent 設定為 80：

```
fl.getDocumentDOM().selection[0].colorAlphaPercent = 80;
```

symbolInstance.colorBlueAmount

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorBlueAmount
```

說明
屬性；整數，屬於實體顏色轉換的一部分。這個屬性等於使用實體「屬性」檢測器中的「顏色 > 進階」設定。允許的值為 -255 到 255。

symbolInstance.colorBluePercent

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorBluePercent
```

說明
屬性；整數，屬於實體顏色轉換的一部分。這個屬性等於使用實體「屬性」檢測器中的「顏色 > 進階」設定 (對話方塊左邊的百分比控制項)。這個值會將藍色值設定為指定的百分比。允許的值為 -100 到 100。

範例
下列範例會將選取元件實體的 colorBluePercent 設定為 80：

```
fl.getDocumentDOM().selection[0].colorBluePercent = 80;
```

symbolInstance.colorGreenAmount

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorGreenAmount
```

說明
屬性；整數，屬於實體顏色轉換的一部分。這個屬性等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定。允許的值為 -255 到 255。

symbolInstance.colorGreenPercent

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorGreenPercent
```

說明
屬性；實體顏色轉換的一部分。這個屬性等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定 (對話方塊左邊的百分比控制項)。這個值會將綠色值設定為指定的百分比。允許的值為 -100 到 100。

範例
下列範例會將選取元件實體的 colorGreenPercent 設定為 70：
```
fl.getDocumentDOM().selection[0].colorGreenPercent = 70;
```

symbolInstance.colorMode

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorMode
```

說明
屬性；字串；根據元件 「屬性」檢測器中的 「顏色」彈出式選單指定顏色模式。可接受的值為 "none"、"brightness"、"tint"、"alpha" 和 "advanced"。

範例
下列範例會將時間軸第一個圖層中第一個影格的第一個元素的 colorMode 屬性，變更為 alpha：
```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].colorMode = "alpha";
```

symbolInstance.colorRedAmount

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorRedAmount
```

說明
屬性；整數，屬於實體顏色轉換的一部分。這個屬性等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定。允許的值為 -255 到 255。

範例
下列範例會將選取元件實體的 colorRedAmount 設定為 255：

```
fl.getDocumentDOM().selection[0].colorRedAmount = 255;
```

symbolInstance.colorRedPercent

適用版本
Flash MX 2004。

用法
```
symbolInstance.colorRedPercent
```

說明
屬性；實體顏色轉換的一部分。這個屬性等於使用實體 「屬性」檢測器中的 「顏色 > 進階」設定 (對話方塊左邊的百分比控制項)。這個值會將紅色值設定為指定的百分比。允許的值為 -100 到 100。

範例
下列範例會將選取元件實體的 colorRedPercent 設定為 10：

```
fl.getDocumentDOM().selection[0].colorRedPercent = 10;
```

symbolInstance.description

適用版本
Flash MX 2004。

用法
```
symbolInstance.description
```

說明
屬性；字串；等於 「輔助功能」面板中的 「說明」欄位。這項說明是由螢幕朗讀程式唸出。圖像元件無法使用這個屬性。

範例
下列範例會將物件的 「輔助功能」面板說明值儲存於 theDescription 變數：

```
var theDescription = fl.getDocumentDOM().selection[0].description;
```

下列範例會將 「輔助功能」面板說明值設定為 Click the home button to go to home：

```
fl.getDocumentDOM().selection[0].description= "Click the home button to go to home";
```

symbolInstance.filters

適用版本
Flash 8。

用法
```
symbolInstance.filters
```

說明
屬性；Filter 物件陣列 (請參閱 Filter 物件)。若要修改濾鏡屬性，請不要直接寫入這個陣列。請改為擷取陣列、設定個別屬性，然後再設定陣列以反應新的屬性。

範例
下列範例會追蹤索引 0 上濾鏡的名稱。如果此濾鏡為 「光暈」濾鏡，其 blurX 屬性會設定為 100，新值會寫入濾鏡陣列。

```
var filterName = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters[0].name;
fl.trace(filterName);
var filterArray = fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters;
if (filterName == 'glowFilter'){
    filterArray[0].blurX = 100;
}
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].filters = filterArray;
```

symbolInstance.firstFrame

適用版本
Flash MX 2004。

用法
```
symbolInstance.firstFrame
```

說明
屬性；從零開始的整數；指定圖像時間軸中出現的第一個影格。這個屬性只會套用至圖像元件，設定屬性與 「屬性」檢測器中的 「第一個」欄位相同。如果是其它類型的元件，這個屬性會是 undefined。

範例
下列範例會將影格 10 指定為指定元素的時間軸中出現的第一個影格：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].firstFrame = 10;
```

symbolInstance.forceSimple

適用版本
Flash MX 2004。

用法
```
symbolInstance.forceSimple
```

說明
屬性；Boolean 值；啟用和停用物件子系的輔助功能。這個屬性等於 「輔助功能」面板 「讓子物件支援輔助功能」設定的反轉邏輯。例如，如果 forceSimple 為 true，則等於未選取 「讓子物件支援輔助功能」選項。如果 forceSimple 為 false，則等於選取 「讓子物件支援輔助功能」選項。

這個屬性只能用於 MovieClip 物件。

範例
下列範例會檢查物件的子系是否可以存取；傳回值 false 代表可以存取子系：

```
var areChildrenAccessible = fl.getDocumentDOM().selection[0].forceSimple;
```

下列範例會允許物件的子系變成可存取：

```
fl.getDocumentDOM().selection[0].forceSimple = false;
```

symbolInstance.loop

適用版本
Flash MX 2004。

用法
```
symbolInstance.loop
```

說明
屬性；字串；用於圖像元件，設定屬性與 「屬性」檢測器中的 「重複」彈出式選單相同。如果是其它類型的元件，這個屬性會是 undefined。可接受的值為 "loop"、"play once" 和 "single frame"，可據此設定圖像動畫。

範例
下列範例會在元件為圖像時，將時間軸第一個圖層中第一個影格的第一個元件，設定為 single frame (顯示圖像時間軸的一個指定影格)：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].loop = 'single frame';
```

symbolInstance.shortcut

適用版本
Flash MX 2004。

用法
```
symbolInstance.shortcut
```

說明
屬性；字串；等於元件相關的快速鍵。這個屬性等於 「輔助功能」面板中的 「快速鍵」欄位。這項按鍵是由螢幕朗讀程式唸出。圖像元件無法使用這個屬性。

範例
下列範例會將物件的快速鍵值儲存於 theShortcut 變數：

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
```

下列範例會將物件的快速鍵設定為 Ctrl+i：

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

symbolInstance.silent

適用版本
Flash MX 2004。

用法
```
symbolInstance.silent
```

說明
屬性；Boolean 值；用來啟用或停用物件的輔助功能。這個屬性等於 「輔助功能」面板 「讓物件支援輔助功能」設定的反轉邏輯。例如，如果 silent 為 true，則等於未選取 「讓物件支援輔助功能」選項。如果 silent 為 false，則等於選取 「讓物件支援輔助功能」選項。

圖像物件無法使用這個屬性。

範例
下列範例會檢查物件是否可以存取；傳回值 false 代表可以存取物件：

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

下列範例會將物件設定為可存取：

```
fl.getDocumentDOM().selection[0].silent = false;
```

symbolInstance.symbolType

適用版本
Flash MX 2004。

用法
```
symbolInstance.symbolType
```

說明
屬性；字串；指定元件類型。這個屬性等於 「建立新元件」和 「轉換成元件」對話方塊中的 「行為指令」值。可接受的值為 "button"、"movie clip" 和 "graphic"。

範例
下列範例會將目前文件的時間軸中，第一個影格的第一個元件設定為圖像元件：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].symbolType = "graphic";
```

symbolInstance.tabIndex

適用版本
Flash MX 2004。

用法
```
symbolInstance.tabIndex
```

說明
屬性；相等於 「輔助功能」面板中 「定位鍵索引」欄位的整數。建立在使用者按下 Tab 鍵時存取物件的定位鍵順序。圖像元件無法使用這個屬性。

範例
下列範例會將 mySymbol 物件的 tabIndex 屬性設定為 3，並在 「輸出」面板中顯示這個值：

```
var mySymbol = fl.getDocumentDOM().selection[0];
mySymbol.tabIndex = 3;
fl.trace(mySymbol.tabIndex);
```

symbolInstance.usesBackgroundColor

適用版本
Flash CS5.5 Professional。

用法
```
symbolInstance.usesBackgroundColor
```

說明
屬性；用來指示要在實體使用 24 位元模式 (true) 還是含 Alpha 的 32 位元模式 (false) 的布林值。若為 true，就指定在實體使用 backgroundColor。

範例
下列範例是將實體的 usesBackgroundColor 屬性設定為 true：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].useTransparentBackground = true;
```

symbolInstance.visible

適用版本
Flash CS5.5 Professional。

用法
```
symbolInstance.visible
```

說明

屬性：將物件的 Visible 屬性設定為開啟 (true) 或關閉 (false) 的布林值。

範例

下列範例是將第一個圖層之第一個影格中的第一個項目的可見性設定為 false：

```
fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].visible = false;
```

第 41 章 SymbolItem 物件

繼承　Item 物件 > SymbolItem 物件

適用版本
Flash MX 2004。

說明
SymbolItem 物件為 Item 物件的子類別。

方法摘要
除了 Item 物件方法之外，SymbolItem 物件還可以搭配下列方法使用：

方法	說明
symbolItem.convertToCompiledClip()	將元件庫中的元件項目轉換成編譯後的影片片段。
symbolItem.exportSWC()	將元件項目匯出至 SWC 檔。
symbolItem.exportSWF()	將元件項目匯出至 SWF 檔。

屬性摘要
除了 Item 物件屬性，SymbolItem 物件還可以使用下列屬性：

屬性	說明
symbolItem.scalingGrid	Boolean 值；指定是否啟用項目的 9 分割縮放。
symbolItem.scalingGridRect	指定四條 9 分割導引線所在位置的 Rectangle 物件。
symbolItem.sourceAutoUpdate	Boolean 值；指定 FLA 檔發佈時是否更新項目。
symbolItem.sourceFilePath	字串；指定來源 FLA 檔的路徑，表示為 file:/// URI。
symbolItem.sourceLibraryName	字串；指定來源檔案元件庫中的項目名稱。
symbolItem.symbolType	字串；指定元件類型。
symbolItem.timeline	唯讀；Timeline 物件。

symbolItem.convertToCompiledClip()

適用版本
Flash MX 2004。

用法
```
symbolItem.convertToCompiledClip()
```

參數
無。

傳回值
無。

說明
方法；將元件庫中的元件項目轉換成編譯後的影片片段。

範例
下列範例會將元件庫中的項目轉換成編譯後的影片片段：

```
fl.getDocumentDOM().library.items[3].convertToCompiledClip();
```

symbolItem.exportSWC()

適用版本
Flash MX 2004。

用法
```
symbolItem.exportSWC(outputURI)
```

參數
outputURI 字串；指定方法要對其匯出元件的 SWC 檔，表示為 file:/// URI。outputURI 必須參考本機檔案。如果 outputURI 不存在，Flash 不會建立資料夾。

傳回值
無。

說明
方法；將元件項目匯出至 SWC 檔。

範例
下列範例會將元件庫中的項目匯出至 tests 資料夾中名為 mySymbol.swc 的 SWC 檔：

```
fl.getDocumentDOM.library.selectItem("mySymbol");
var currentSelection = fl.getDocumentDOM().library.getSelectedItems();
currentSelection[0].exportSWC("file:///Macintosh HD/SWCDirectory/mySymbol.swc");
```

symbolItem.exportSWF()

適用版本
Flash MX 2004。

用法
```
symbolItem.exportSWF(outputURI)
```

參數
outputURI 字串；指定方法要對其匯出元件的 SWF 檔，表示為 file:/// URI。outputURI 必須參考本機檔案。如果 outputURI 不存在，Flash 不會建立資料夾。

傳回值
無。

說明
方法；將元件項目匯出至 SWF 檔。

範例
下列範例會將元件庫中的項目匯出至 tests 資料夾中的 my.swf 檔：

```
fl.getDocumentDOM().library.items[0].exportSWF("file:///c|/tests/my.swf");
```

symbolItem.scalingGrid

適用版本
Flash 8。

用法
```
symbolItem.scalingGrid
```

說明
屬性；Boolean 值；指定是否啟用項目的 9 分割縮放。

範例
下列範例會啟用元件庫中項目的 9 分割縮放：

```
fl.getDocumentDOM().library.items[0].scalingGrid = true;
```

請參閱
symbolItem.scalingGridRect

symbolItem.scalingGridRect

適用版本
Flash 8。

用法
```
symbolItem.scalingGridRect
```

說明
屬性；指定四條 9 分割導引線所在位置的 Rectangle 物件。如需有關矩形格式的詳細資訊，請參閱 document.addNewRectangle()。

範例
下列範例指定 9 分割導引線的位置：

```
fl.getDocumentDOM().library.items[0].scalingGridRect = {left:338, top:237, right:3859, bottom:713};
```

請參閱
symbolItem.scalingGrid

symbolItem.sourceAutoUpdate

適用版本
Flash MX 2004。

用法
```
symbolItem.sourceAutoUpdate
```

說明
屬性；Boolean 值；指定 FLA 檔發佈時是否更新項目。預設值為 false。用於共享元件庫元件。

範例
下列範例會設定元件庫項目的 sourceAutoUpdate 屬性：

```
fl.getDocumentDOM().library.items[0].sourceAutoUpdate = true;
```

symbolItem.sourceFilePath

適用版本
Flash MX 2004。

用法
```
symbolItem.sourceFilePath
```

說明
屬性；字串；指定來源 FLA 檔的路徑，表示為 file:/// URI。路徑必須是絕對路徑，而不是相對路徑。此屬性用於共享元件庫元件。

範例
下列範例會在「輸出」面板中顯示 sourceFilePath 屬性的值：

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceFilePath);
```

symbolItem.sourceLibraryName

適用版本
Flash MX 2004。

用法
```
symbolItem.sourceLibraryName
```

說明
屬性；字串；指定來源檔案元件庫中的項目名稱。此屬性用於共享元件庫。

範例
下列範例會在「輸出」面板中顯示 sourceLibraryName 屬性的值：

```
fl.trace(fl.getDocumentDOM().library.items[0].sourceLibraryName);
```

symbolItem.symbolType

適用版本
Flash MX 2004。

用法
```
symbolItem.symbolType
```

說明
屬性；字串；指定元件類型。可接受的值為 "movie clip"、"button" 和 "graphic"。

範例
下列範例會顯示 symbolType 屬性的目前值、將值變更為 button，並再次顯示這個值：

```
alert(fl.getDocumentDOM().library.items[0].symbolType);
fl.getDocumentDOM().library.items[0].symbolType = "button";
alert(fl.getDocumentDOM().library.items[0].symbolType);
```

symbolItem.timeline

適用版本
Flash MX 2004。

用法
```
symbolItem.timeline
```

說明
唯讀屬性；Timeline 物件。

範例
下列範例會取得並顯示元件庫中選取的影片片段所包含的圖層數：

```
var tl = fl.getDocumentDOM().library.getSelectedItems()[0].timeline;
alert(tl.layerCount);
```

第 42 章 Text 物件

繼承 Element 物件 > Text 物件

適用版本

Flash MX 2004。

說明

Text 物件代表文件中的單一文字項目。文字的所有屬性會與整個文字區塊有關。

若要在文字欄位內設定連續文字的屬性，請參閱 TextAttrs 物件的屬性摘要。若要在文字欄位內變更選擇範圍的屬性，您可以使用 document.setElementTextAttr() 並指定文字範圍，或使用目前的選擇範圍。

若要設定選取的文字欄位的一般屬性，請使用 document.setElementProperty()。下列範例會將選取的文字欄位註冊點的 x 值設定為 50：

```
fl.getDocumentDOM().setElementProperty("x", 50);
```

方法摘要

除了 Element 物件方法，Text 物件還可以使用下列方法：

方法	說明
text.getTextAttr()	擷取選擇性 *startIndex* 和 *endIndex* 參數識別的文字的指定特質。
text.getTextString()	擷取指定範圍的文字。
text.setTextAttr()	設定 *startIndex* 和 *endIndex* 識別的文字的相關指定特質。
text.setTextString()	變更這個 Text 物件內的文字字串。

屬性摘要

除了 Element 物件屬性，Text 物件還可以使用下列屬性：

屬性	說明
text.accName	字串；等於「輔助功能」面板中的「名稱」欄位。
text.antiAliasSharpness	浮點值；指定文字的消除鋸齒清晰度。
text.antiAliasThickness	浮點值；指定文字的消除鋸齒濃密度。
text.autoExpand	Boolean 值；控制靜態文字欄位的外框寬度擴展，或是動態或輸入文字的外框寬度和高度擴展。
text.border	Boolean 值；控制 Flash 顯示 (true) 或隱藏 (false) 動態或輸入文字的邊框。
text.description	字串；等於「輔助功能」面板的「說明」欄位。
text.embeddedCharacters	字串；指定要內嵌的字元。等於在「字元內嵌」對話方塊中輸入文字。
text.embedRanges	字串；由分隔的整數組成，這些整數與「字元內嵌」對話方塊中可選取的項目相對應。
text.embedVariantGlyphs	Boolean 值，指定是否啟用變化字形內嵌。
text.fontRenderingMode	字串；指定文字的顯示模式。
text.length	唯讀；整數，代表 Text 物件中的字元數。

屬性	說明
text.lineType	字串；會將字行類型設定為 "single line"、"multiline"、"multiline no wrap" 或 "password"。
text.maxCharacters	整數；指定使用者可以在這個 Text 物件中輸入的最大字元數。
text.orientation	字串；指定文字欄位的方向。
text.renderAsHTML	Boolean 值；控制 Flash 是否將文字繪製為 HTML，並解譯內嵌的 HTML 標籤。
text.scrollable	Boolean 值；控制文字可以捲動 (true) 或不可以捲動 (false)。
text.selectable	Boolean 值；控制文字可以選取 (true) 或不可以選取 (false)。輸入的文字一定可以選取。
text.selectionEnd	從零開始的整數；指定文字細部選取結尾的偏移值。
text.selectionStart	從零開始的整數；指定文字細部選取起始位置的偏移值。
text.shortcut	字串；等於 「輔助功能」面板的 「快速鍵」欄位。
text.silent	Boolean 值；指定物件是否可存取。
text.tabIndex	整數；等於 「輔助功能」面板中的 「定位鍵索引」欄位。
text.textRuns	唯讀；TextRun 物件的陣列。
text.textType	字串；指定文字欄位的類型。可接受的值為 "static"、"dynamic" 和 "input"。
text.useDeviceFonts	Boolean 值。true 值會讓 Flash 使用裝置字體繪製文字。
text.variableName	字串，包含 Text 物件的內容。

text.accName

適用版本
Flash MX 2004。

用法
```
text.accName
```

說明
屬性；字串；等於 「輔助功能」面板中的 「名稱」欄位。螢幕朗讀程式將大聲唸出物件名稱來識別物件。這個屬性無法與動態文字搭配使用。

範例
下列範例會擷取物件的名稱：

```
var doc = fl.getDocumentDOM();
var theName = doc.selection[0].accName;
```

下列範例會設定目前選取物件的名稱：

```
fl.getDocumentDOM().selection[0].accName = "Home Button";
```

text.antiAliasSharpness

適用版本
Flash 8。

用法
```
text.antiAliasSharpness
```

說明
屬性；浮點值；指定文字的消除鋸齒清晰度。這個屬性控制文字繪製的清晰度；較高的值指定較為銳利 (清晰) 的文字。0 值則指定一般銳利度。只有當 text.fontRenderingMode 設定為 customThicknessSharpness 時，才能使用這個屬性。

範例
請參閱 text.fontRenderingMode。

請參閱
text.antiAliasThickness、text.fontRenderingMode

text.antiAliasThickness

適用版本
Flash 8。

用法
```
text.antiAliasThickness
```

說明
屬性；浮點值；指定文字的消除鋸齒濃密度。這個屬性控制文字繪製的粗細，較高的值指定較粗的文字。0 值則指定一般粗細。只有當 text.fontRenderingMode 設定為 customThicknessSharpness 時，才能使用這個屬性。

範例
請參閱 text.fontRenderingMode。

請參閱
text.antiAliasSharpness、text.fontRenderingMode

text.autoExpand

適用版本
Flash MX 2004。

用法
```
text.autoExpand
```

說明
屬性；Boolean 值。如果是靜態文字欄位，true 值會展開外框寬度，以顯示所有的文字。如果是動態或輸入文字欄位，true 值會展開外框寬度和高度，以顯示所有的文字。

範例
下列範例會將 autoExpand 屬性設定為 true 值：

```
fl.getDocumentDOM().selection[0].autoExpand = true;
```

text.border

適用版本
Flash MX 2004。

用法
```
text.border
```

說明
屬性；Boolean 值。true 值會讓 Flash 顯示文字的邊框。

範例
下列範例會將 border 屬性設定為 true 值：

```
fl.getDocumentDOM().selection[0].border = true;
```

text.description

適用版本
Flash MX 2004。

用法
```
text.description
```

說明
屬性；字串；等於 「輔助功能」面板中的 「說明」欄位。這項說明是由螢幕朗讀程式唸出。

範例
下列範例會擷取物件的說明：

```
var doc = fl.getDocumentDOM();
var desc = doc.selection[0].description;
```

下列範例會設定物件的說明：

```
var doc = fl.getDocumentDOM();
doc.selection[0].description= "Enter your name here";
```

text.embeddedCharacters

適用版本
Flash MX 2004。

用法
```
text.embeddedCharacters
```

說明
屬性；字串；指定要內嵌的字元。等於在 「字元內嵌」對話方塊中輸入文字。

這個屬性只能與動態和輸入文字搭配使用，如果與其它文字類型搭配使用，則會產生警告。

備註：從 Flash Professional CS5 開始，會在文件層級而不是文字物件層級控制字體內嵌。請使用 第 258 頁「fontItem.embeddedCharacters」屬性，而不是 text.embeddedCharacters 屬性。

範例
下列範例假設在目前選取範圍中的第一個項目或唯一項目是傳統文字物件，並將 embeddedCharacters 屬性設定為 abc：

```
fl.getDocumentDOM().selection[0].embeddedCharacters = "abc";
```

text.embedRanges

適用版本
Flash MX 2004。

用法
```
text.embedRanges
```

說明
屬性；字串；由分隔的整數組成，這些整數與 「字元內嵌」對話方塊中可選取的項目相對應。這個屬性只能與動態或輸入文字搭配使用，如果與靜態文字搭配使用，則會忽略這個屬性。

這個屬性與 Configuration/Font Embedding 資料夾中的 XML 檔相對應。

備註：從 Flash Professional CS5 開始，會在文件層級而不是文字物件層級控制字體內嵌。請使用 第 258 頁「fontItem.embedRanges」屬性，而不是 text.embedRanges 屬性。

範例
下列範例假設目前選取範圍中的第一個項目或唯一項目是傳統文字物件，並將 embedRanges 屬性設定為 "1|3|7"：

```
var doc = fl.getDocumentDOM();
doc.selection[0].embedRanges = "1|3|7";
```

下列範例會重設屬性：

```
var doc = fl.getDocumentDOM();
doc.selection[0].embedRanges = "";
```

text.embedVariantGlyphs

適用版本
Flash CS4 Professional。

用法
```
text.embedVariantGlyphs
```

說明
屬性；Boolean 值，指定是 (true) 否 (false) 啟用變化字形內嵌。這個屬性只能與動態或輸入文字搭配使用，如果與靜態文字搭配使用，則會忽略這個屬性。預設值為 false。

備註：從 Flash Professional CS5 開始，會在文件層級而不是文字物件層級控制字體內嵌。請使用 第 259 頁「fontItem.embedVariantGlyphs」屬性，而不是 text.embedVariantGlyphs 屬性。在 Flash Professional CS5 中，text.embedVariantGlyphs 屬性將不再有任何作用，因為 Flash 永遠會在 TLF 文字中內嵌變化字形，但絕不會在傳統文字中內嵌變化字形。

範例
下列範例會讓變化字形內嵌在選取的 Text 物件中：

```
fl.getDocumentDOM().selection[0].embedVariantGlyphs = true;
```

請參閱
fontItem.embedVariantGlyphs

text.fontRenderingMode

適用版本
Flash 8。

用法
```
text.fontRenderingMode
```

說明
屬性；字串；判斷文字的顯示模式。這個屬性會同時影響在「舞台」和 Flash Player 中文字顯示的方式。可接受的值如下表說明：

屬性值	文字的顯示方式
device	以裝置字體顯示文字。
bitmap	將鋸齒狀的文字呈現為點陣圖，或如同像素字體的外觀。
standard	使用 Flash MX 2004 所使用的標準消除鋸齒方法來顯示文字。這是動畫、超大型或傾斜文字最適用的設定。
advanced	使用 Flash 8 中配備的先進消除鋸齒字體呈現技術呈現文字；該項技術可以產生較佳的消除鋸齒效果，並改善易讀性，特別是小型文字。
customThicknessSharpness	讓您在使用 Flash 8 中配備的先進消除鋸齒字體呈現技術時，指定文字銳利度和粗細的自訂設定。

範例

下列範例會顯示如何使用 customThicknessSharpness 值，指定文字的清晰度和粗細：

```
fl.getDocumentDOM().setElementProperty("fontRenderingMode", "customThicknessSharpness");
fl.getDocumentDOM().setElementProperty("antiAliasSharpness", 400);
fl.getDocumentDOM().setElementProperty("antiAliasThickness", -200);
```

請參閱

text.antiAliasSharpness、text.antiAliasThickness

text.getTextAttr()

適用版本

Flash MX 2004。

用法

```
text.getTextAttr(attrName [, startIndex [, endIndex]])
```

參數

attrName 字串：指定傳回的 TextAttrs 物件屬性名稱。如需 ***attrName*** 的可能值清單，請參閱 TextAttrs 物件的屬性摘要。

startIndex 整數：為第一個字元的索引。這個參數是選擇性參數。

endIndex 整數：指定文字範圍的結尾，這個文字範圍會以 startIndex 做為開頭，並延續至 (但不包含) endIndex。這個參數是選擇性參數。

傳回值

attrName 參數中指定的特質值。

說明

方法：擷取選擇性 startIndex 和 endIndex 參數識別文字的特質 (以 attrName 參數指定)。如果特質和指定的範圍不一致，Flash 會傳回 undefined。如果您略過選擇性參數 startIndex 和 endIndex，則這個方法會使用整個文字範圍。如果您只指定 startIndex，則使用範圍是這個位置的單一字元。如果您指定 startIndex 和 endIndex 兩者，則範圍會從 startIndex 開始，並延續至 (但不包含) endIndex。

範例

下列範例會取得目前選取文字欄位的字體大小並顯示：

```
var TheTextSize = fl.getDocumentDOM().selection[0].getTextAttr("size");
fl.trace(TheTextSize);
```

下列範例會取得選取文字欄位的文字填色顏色：

```
var TheFill = fl.getDocumentDOM().selection[0].getTextAttr("fillColor");
fl.trace(TheFill);
```

下列範例會取得第三個字元的大小：

```
var Char3 = fl.getDocumentDOM().selection[0].getTextAttr("size", 2);
fl.trace(Char3);
```

下列範例會從選取文字欄位中的第三到第八個字元取得顏色：

```
fl.getDocumentDOM().selection[0].getTextAttr("fillColor", 2, 8);
```

text.getTextString()

適用版本
Flash MX 2004。

用法
```
text.getTextString([startIndex [, endIndex]])
```

參數
startIndex 整數；指定第一個字元的索引 (從零開始)。這個參數是選擇性參數。

endIndex 整數；指定文字範圍的結尾，這個文字範圍會從 startIndex 開始，並延續至 (但不包含) endIndex。這個參數是選擇性參數。

傳回值
指定範圍中的文字字串。

說明
方法；擷取指定範圍的文字。如果您忽略選擇性參數 startIndex 和 endIndex，則會傳回整個文字字串。如果您只指定 startIndex，這個方法會傳回開始於索引位置並結束於欄位結尾的字串。如果您指定 startIndex 和 endIndex 兩者，這個方法會傳回從 startIndex 開始且延續至 (但不包含) endIndex 的字串。

範例
下列範例會取得從第五個字元至選取文字欄位結尾之間的字元：

```
var myText = fl.getDocumentDOM().selection[0].getTextString(4);
fl.trace(myText);
```

下列範例從選取文字欄位開始，取得第四至第九個字元：

```
var myText = fl.getDocumentDOM().selection[0].getTextString(3, 9);
fl.trace(myText);
```

text.length

適用版本
Flash MX 2004。

用法
```
text.length
```

說明
唯讀屬性；整數，代表 Text 物件中的字元數。

範例
下列範例會傳回選取文字中的字元數：

```
var textLength = fl.getDocumentDOM().selection[0].length;
```

text.lineType

適用版本
Flash MX 2004。

用法
```
text.lineType
```

說明
屬性；設定字行類型的字串。可接受的值為 "single line"、"multiline"、"multiline no wrap" 和 "password"。

這個屬性只能與動態或輸入文字搭配使用，如果與靜態文字搭配使用，則會產生警告。"password" 值只能用於輸入文字。

範例
下列範例會將 lineType 屬性設定為值 multiline no wrap：

```
fl.getDocumentDOM().selection[0].lineType = "multiline no wrap";
```

text.maxCharacters

適用版本
Flash MX 2004。

用法
```
text.maxCharacters
```

說明
屬性；整數，指定使用者可以在 Text 物件中輸入的最大字元數。

這個屬性只能與輸入文字搭配使用，如果與其它文字類型搭配使用，則這個屬性會產生警告。

範例
下列範例會將 maxCharacters 屬性的值設定為 30：

```
fl.getDocumentDOM().selection[0].maxCharacters = 30;
```

text.orientation

適用版本
Flash MX 2004。

用法
```
text.orientation
```

說明
屬性；字串；指定文字欄位的方向。可接受的值為 "horizontal"、"vertical left to right" 和 "vertical right to left"。

這個屬性只能與靜態文字搭配使用，如果與其它文字類型搭配使用，則會產生警告。

範例
下列範例會將方向屬性設定為 vertical right to left：

```
fl.getDocumentDOM().selection[0].orientation = "vertical right to left";
```

text.renderAsHTML

適用版本
Flash MX 2004。

用法
```
text.renderAsHTML
```

說明
屬性；Boolean 值。如果這個值是 true，Flash 會將文字繪製為 HTML，並解譯內嵌的 HTML 標籤。

這個屬性只能與動態和輸入文字搭配使用，如果與其它文字類型搭配使用，則會產生警告。

範例
下列範例會將 renderAsHTML 屬性設定為 true：

```
fl.getDocumentDOM().selection[0].renderAsHTML = true;
```

text.scrollable

適用版本
Flash MX 2004。

用法
```
text.scrollable
```

說明
屬性；Boolean 值。如果這個值是 true，則可以捲動文字。

這個屬性只能與動態或輸入文字搭配使用，如果與靜態文字搭配使用，則會產生警告。

範例
下列範例會將 scrollable 屬性設定為 false：

```
fl.getDocumentDOM().selection[0].scrollable = false;
```

text.selectable

適用版本
Flash MX 2004。

用法
```
text.selectable
```

說明
屬性；Boolean 值。如果這個值是 true，則可以選取文字。

輸入的文字一定可以選取。如果這個屬性設定為 false 並與輸入文字搭配使用，Flash 會產生警告。

範例
下列範例會將 selectable 屬性設定為 true：

```
fl.getDocumentDOM().selection[0].selectable = true;
```

text.selectionEnd

適用版本
Flash MX 2004。

用法
```
text.selectionEnd
```

說明
屬性；從零開始的整數；指定文字細部選取的結尾。如需詳細資訊，請參閱 text.selectionStart。

text.selectionStart

適用版本
Flash MX 2004。

用法
```
text.selectionStart
```

說明
屬性；從零開始的整數；指定文字細部選取的開頭。這個屬性可以搭配 text.selectionEnd 使用，選取某個範圍的字元。字元最多會選取至 (但不包含) text.selectionEnd。請參閱 text.selectionEnd。

- 如果有插入點或沒有選取範圍，text.selectionEnd 就等於 text.selectionStart。
- 如果 text.selectionStart 的設定值大於 text.selectionEnd 的設定值，則 text.selectionEnd 會設定為 text.selectionStart，而且不會選取文字。

範例
下列範例會將文字細部選取的開頭設定為第六個字元：

```
fl.getDocumentDOM().selection[0].selectionStart = 5;
```

下列範例會從包含文字 My name is Barbara 的文字欄位中，選取字元 Barbara，並將其格式設定為粗體和綠色：

```
fl.getDocumentDOM().selection[0].selectionStart = 11;
fl.getDocumentDOM().selection[0].selectionEnd = 18;
var s = fl.getDocumentDOM().selection[0].selectionStart;
var e = fl.getDocumentDOM().selection[0].selectionEnd;
fl.getDocumentDOM().setElementTextAttr('bold', true, s, e);
fl.getDocumentDOM().setElementTextAttr("fillColor", "#00ff00", s, e);
```

text.setTextAttr()

適用版本

Flash MX 2004。

用法

```
text.setTextAttr(attrName, attrValue [, startIndex [, endIndex]])
```

參數

attrName 字串；指定變更的 TextAttrs 物件屬性名稱。

attrValue TextAttrs 物件的屬性值。

如需 ***attrName*** 和 ***attrValue*** 的可能值清單，請參閱 TextAttrs 物件的屬性摘要。

startIndex 整數；陣列中第一個字元的索引 (從零開始)。這個參數是選擇性參數。

endIndex 整數；指定選取的文字字串中索引的結束點，它會以 startIndex 做為開頭，並延續至 (但不包含) endIndex。這個參數是選擇性參數。

傳回值

無。

說明

方法；將 startIndex 和 endIndex 識別文字有關的特質 (以 attrName 參數指定)，設定為 attrValue 指定的值。這個方法可以用來變更文字的特質，這些特質可能橫跨多個 TextRun 元素 (請參閱 TextRun 物件)，或是現有 TextRun 元素的一部分。使用這個方法時，可能會變更此物件之 text.textRuns 陣列中的 TextRun 元素位置和數目 (請參閱 text.textRuns)。

如果您省略選擇性的參數，此方法就會使用整個 Text 物件的字元範圍。如果您只指定 startIndex，則範圍是這個位置的單一字元。如果您指定 startIndex 和 endIndex 兩者，範圍會從 startIndex 開始，並延續至 (但不包含) 位於 endIndex 的字元。

範例

下列範例會將選取的文字欄位設定為斜體：

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

下列範例會將第三個字元的大小設定為 10：

```
fl.getDocumentDOM().selection[0].setTextAttr("size", 10, 2);
```

下列範例會將選取文字第三至第八個字元的顏色設定為紅色：

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

text.setTextString()

適用版本
Flash MX 2004。

用法
```
text.setTextString(text [, startIndex [, endIndex]])
```

參數
text 字串，由插入這個 Text 物件的字元所組成。

startIndex 整數；指定插入文字字串中的字元索引 (從零開始)。這個參數是選擇性參數。

endIndex 整數；指定選取文字字串中的端點索引。新文字會覆寫從 startIndex 至 (但不包括) endIndex 的文字。這個參數是選擇性參數。

傳回值
無。

說明
屬性；變更這個 Text 物件內的文字字串。如果您呼略這個選擇性的參數，則會取代整個 Text 物件。如果您只指定 startIndex，則指定的字串會插入 startIndex 位置。如果您指定 startIndex 和 endIndex 兩者，則指定的字串會取代從 startIndex 開始，並延續至 (但不包含) endIndex 的文字片段。

範例
下列範例會將字串 this is a string 指定給選取的文字欄位：

```
fl.getDocumentDOM().selection[0].setTextString("this is a string");
```

下列範例會從選取文字欄位的第五個字元開始，插入字串 abc：

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");
fl.getDocumentDOM().selection[0].setTextString("abc", 4);
// text field is now "0123abc4567890"
```

下列範例會將選取的文字字串的第三至第八個字元文字，取代為字串 abcdefghij，並且覆寫 startIndex 和 endIndex 之間的字元。自 endIndex 開始的字元，會在插入字串之後。

```
fl.getDocumentDOM().selection[0].setTextString("01234567890");
fl.getDocumentDOM().selection[0].setTextString("abcdefghij", 2, 8);
// text field is now "01abcdefghij890"
```

text.shortcut

適用版本
Flash MX 2004。

用法
```
text.shortcut
```

說明
屬性；字串；等於 「輔助功能」面板中的 「快速鍵」欄位。這項快速鍵是由螢幕朗讀程式唸出。這個屬性無法與動態文字搭配使用。

範例
下列範例會取得選取物件的快速鍵並顯示其值：

```
var theShortcut = fl.getDocumentDOM().selection[0].shortcut;
fl.trace(theShortcut);
```

下列範例會設定選取物件的快速鍵：

```
fl.getDocumentDOM().selection[0].shortcut = "Ctrl+i";
```

text.silent

適用版本
Flash MX 2004。

用法
```
text.silent
```

說明
屬性；Boolean 值；指定物件是否可存取。等於 「輔助功能」面板 「讓物件支援輔助功能」設定的反向邏輯。也就是說，如果 silent 是 true，則未選取 「讓物件支援輔助功能」。如果是 false，則選取 「讓物件支援輔助功能」。

範例
下列範例會判斷物件是否可存取 (false 值代表可存取)：

```
var isSilent = fl.getDocumentDOM().selection[0].silent;
```

下列範例會將物件設定為可存取：

```
fl.getDocumentDOM().selection[0].silent = false;
```

text.tabIndex

適用版本
Flash MX 2004。

用法
```
text.tabIndex
```

說明
屬性；相等於 「輔助功能」面板中 「定位鍵索引」欄位的整數。這個值可以讓您決定在使用者按下 Tab 鍵時，物件的存取順序。

範例
下列範例會取得目前選取物件的 tabIndex：

```
var theTabIndex = fl.getDocumentDOM().selection[0].tabIndex;
```

下列範例會設定目前選取物件的 tabIndex：

```
fl.getDocumentDOM().selection[0].tabIndex = 1;
```

text.textRuns

適用版本
Flash MX 2004。

用法
```
text.textRuns
```

說明
唯讀屬性；TextRun 物件陣列 (請參閱 TextRun 物件)。

範例
下列範例會將 textRuns 屬性的值儲存於 myTextRuns 變數：

```
var myTextRuns = fl.getDocumentDOM().selection[0].textRuns;
```

text.textType

適用版本
Flash MX 2004。

用法
```
text.textType
```

說明
屬性；字串；指定文字欄位的類型。可接受的值為 "static"、"dynamic" 和 "input"。

範例
下列範例會將 textType 屬性設定為 input：

```
fl.getDocumentDOM().selection[0].textType = "input";
```

text.useDeviceFonts

適用版本
Flash MX 2004。

用法
```
text.useDeviceFonts
```

說明
屬性；Boolean 值。true 值會讓 Flash 使用裝置字體繪製文字。

範例
下列範例會讓 Flash 在繪製文字時，使用裝置字體：

```
fl.getDocumentDOM().selection[0].useDeviceFonts = true;
```

text.variableName

適用版本
Flash MX 2004。

用法
```
text.variableName
```

說明
屬性；字串，其中包含與 Text 物件有關的變數名稱。這個屬性只能與動態和輸入文字搭配使用，如果與其它文字類型搭配使用，則會產生警告。

只有在 ActionScript 1.0 和 ActionScript 2.0 中，才會支援這個屬性。

範例
下列範例會將選取的文字方塊變數名稱設定為 firstName：

```
fl.getDocumentDOM().selection[0].variableName = "firstName";
```

第 43 章 TextAttrs 物件

適用版本
Flash MX 2004。

說明
TextAttrs 物件包含所有可以套用於細部選取的文字屬性。這個物件是 TextRun 物件的屬性 (textRun.textAttrs)。

屬性摘要
TextAttrs 物件可使用的屬性如下：

屬性	說明
textAttrs.aliasText	Boolean 值；指定 Flash 繪製文字時，應該使用增加小型文字易讀性的最佳化方法。
textAttrs.alignment	字串；指定段落齊行。可接受的值為 "left"、"center"、"right" 和 "justify"。
textAttrs.autoKern	Boolean 值；判斷 Flash 使用 (true) 或忽略 (false) 字體的字距微調資訊，對文字進行字距微調。
textAttrs.bold	Boolean 值。true 值會讓文字以字體的粗體顯示。
textAttrs.characterPosition	字串；判斷文字基線。
textAttrs.characterSpacing	不建議使用，請改用 textAttrs.letterSpacing。整數；代表字元間距。
textAttrs.face	字串；代表字體名稱，例如 "Arial"。
textAttrs.fillColor	字串，代表填色顏色的十六進位值或整數。
textAttrs.indent	整數；指定段落縮排。
textAttrs.italic	Boolean 值。true 值會讓文字以字體的斜體顯示。
textAttrs.leftMargin	整數；指定段落的左邊界。
textAttrs.letterSpacing	整數；代表字元間距。
textAttrs.lineSpacing	整數；指定段落的行間距 (行距)。
textAttrs.rightMargin	整數；指定段落的右邊界。
textAttrs.rotation	Boolean 值。true 值會使 Flash 將文字的字元旋轉 90 度。預設值為 false。
textAttrs.size	整數；指定字體大小。
textAttrs.target	字串；代表文字欄位的 target 屬性。
textAttrs.url	字串；代表文字欄位的 URL 屬性。

textAttrs.aliasText

適用版本
Flash MX 2004。

用法
```
textAttrs.aliasText
```

說明
屬性；Boolean 值；指定 Flash 繪製文字時，應該使用增加小型文字易讀性的最佳化方法。

範例
下列範例會將目前選取文字欄位中所有文字的 aliasText 屬性，設定為 true：

```
fl.getDocumentDOM().setElementTextAttr('aliasText', true);
```

textAttrs.alignment

適用版本
Flash MX 2004。

用法
```
textAttrs.alignment
```

說明
屬性；字串；指定段落齊行。可接受的值為 "left"、"center"、"right" 和 "justify"。

範例
下列範例會將索引 0 到 (但不包含) 索引 3 之間包含字元的段落，設定為齊行。如果指定範圍之外的字元位於同一段落，這些字元也會受到影響。

```
fl.getDocumentDOM().setTextSelection(0, 3);
fl.getDocumentDOM().setElementTextAttr("alignment", "justify");
```

textAttrs.autoKern

適用版本
Flash MX 2004。

用法
```
textAttrs.autoKern
```

說明
屬性；Boolean 值；判斷 Flash 使用 (true) 或忽略 (false) 字體的字距微調資訊，對文字進行字距微調。

範例
下列範例會選取索引 2 到 (但不包含) 索引 6 之間的字元，並將 autoKern 屬性設定為 true：

```
fl.getDocumentDOM().setTextSelection(3, 6);
fl.getDocumentDOM().setElementTextAttr('autoKern', true);
```

textAttrs.bold

適用版本
Flash MX 2004。

用法
```
textAttrs.bold
```

說明
屬性；Boolean 值。true 值會讓文字以字體的粗體顯示。

範例
下列範例會選取被選取 Text 物件的第一個字元，並將 bold 屬性設定為 true：

```
fl.getDocumentDOM().setTextSelection(0, 1);
fl.getDocumentDOM().setElementTextAttr('bold', true);
```

textAttrs.characterPosition

適用版本
Flash MX 2004。

用法
```
textAttrs.characterPosition
```

說明
屬性；字串；判斷文字基線。可接受的值為 "normal"、"subscript" 和 "superscript"。這個屬性只能套用至靜態文字。

範例
下列範例會選取所選文字欄位中索引 2 到 (但不包含) 索引 6 的字元，並將 characterPosition 屬性設定為 subscript：

```
fl.getDocumentDOM().setTextSelection(2, 6);
fl.getDocumentDOM().setElementTextAttr("characterPosition", "subscript");
```

textAttrs.characterSpacing

適用版本
Flash MX 2004。不建議在 Flash 8 中使用，請改用 textAttrs.letterSpacing。

用法
```
textAttrs.characterSpacing
```

說明
屬性；整數，代表字元間距。可接受的值從 -60 到 60。

這個屬性只能套用至靜態文字，如果與其它文字類型搭配使用，則會產生警告。

範例
下列範例會將選取文字欄位的字元間距設定為 10：

```
fl.getDocumentDOM().setElementTextAttr("characterSpacing", 10);
```

textAttrs.face

適用版本
Flash MX 2004。

用法
```
textAttrs.face
```

說明
屬性；字串；代表字體名稱，例如 "Arial"。

範例
下列範例會將所選文字欄位中索引 2 到 (但不包含) 索引 8 的字元字體，都設定為 Arial：

```
fl.getDocumentDOM().selection[0].setTextAttr("face", "Arial", 2, 8);
```

textAttrs.fillColor

適用版本
Flash MX 2004。

用法
```
textAttrs.fillColor
```

說明
屬性；填色的顏色，採用下列其中一種格式：

- 格式為 "#RRGGBB" 或 "#RRGGBBAA" 的字串
- 格式為 0xRRGGBB 的十六進位數字
- 整數，代表十六進位數字對等的十進位值

範例
下列範例會將選取文字欄位中索引 2 到索引 8 (但不包含) 的字元顏色，都設定為紅色：

```
fl.getDocumentDOM().selection[0].setTextAttr("fillColor", 0xff0000, 2, 8);
```

textAttrs.indent

適用版本
Flash MX 2004。

用法
```
textAttrs.indent
```

說明
屬性；整數；指定段落縮排。可接受的值從 -720 到 720。

範例
下列範例會將選取文字欄位中索引 2 到 (但不包含) 索引 8 的縮排，設定為 100。如果指定範圍之外的字元位於同一段落，這些字元也會受到影響。

```
fl.getDocumentDOM().selection[0].setTextAttr("indent", 100, 2, 8);
```

textAttrs.italic

適用版本
Flash MX 2004。

用法
```
textAttrs.italic
```

說明
屬性；Boolean 值。true 值會讓文字以字體的斜體顯示。

範例
下列範例會將選取的文字欄位設定為斜體：

```
fl.getDocumentDOM().selection[0].setTextAttr("italic", true);
```

textAttrs.leftMargin

適用版本
Flash MX 2004。

用法
```
textAttrs.leftMargin
```

說明
屬性；整數；指定段落的左邊界。可接受的值從 0 到 720。

範例
下列範例會將選取文字欄位中索引 2 到 (但不包含) 索引 8 的 leftMargin 屬性，設定為 100。如果指定範圍之外的字元位於同一段落，這些字元也會受到影響。

```
fl.getDocumentDOM().selection[0].setTextAttr("leftMargin", 100, 2, 8);
```

textAttrs.letterSpacing

適用版本
Flash 8。

用法
```
textAttrs.letterSpacing
```

說明
屬性；整數，代表字元間距。可接受的值從 -60 到 60。

這個屬性只能套用至靜態文字，如果與其它文字類型搭配使用，則會產生警告。

範例
下列的程式碼會選取索引 0 到索引 10 (不含) 的字元，並將字元間距設為 60：

```
fl.getDocumentDOM().setTextSelection(0, 10);
fl.getDocumentDOM().setElementTextAttr("letterSpacing", 60);
```

textAttrs.lineSpacing

適用版本
Flash MX 2004。

用法
```
textAttrs.lineSpacing
```

說明
屬性；整數；指定段落的行間距 (「行距」)。可接受的值從 -360 到 720。

範例
下列範例會將選取文字欄位的 lineSpacing 屬性設定為 100：

```
fl.getDocumentDOM().selection[0].setTextAttr("lineSpacing", 100);
```

textAttrs.rightMargin

適用版本
Flash MX 2004。

用法
```
textAttrs.rightMargin
```

說明
屬性；整數；指定段落的右邊界。可接受的值從 0 到 720。

範例
下列範例會將選取文字欄位中索引 2 到 (但不包含) 索引 8 的 rightMargin 屬性，設定為 100。如果指定範圍之外的字元位於同一段落，這些字元也會受到影響。

```
fl.getDocumentDOM().selection[0].setTextAttr("rightMargin", 100, 2, 8);
```

textAttrs.rotation

適用版本
Flash MX 2004。

用法
```
textAttrs.rotation
```

說明
屬性：Boolean 值。true 值會使 Flash 將文字的字元旋轉 90 度。預設值為 false。這個屬性只能套用至垂直方向的靜態文字，如果與其它文字類型搭配使用，則會產生警告。

範例
下列範例會將選取文字欄位的旋轉設定為 true：

```
fl.getDocumentDOM().setElementTextAttr("rotation", true);
```

textAttrs.size

適用版本
Flash MX 2004。

用法
```
textAttrs.size
```

說明
屬性：整數；指定字體大小。

範例
下列範例會擷取索引 2 的字元大小，然後在 「輸入」面板中顯示結果：

```
fl.outputPanel.trace(fl.getDocumentDOM().selection[0].getTextAttr("size", 2));
```

textAttrs.target

適用版本
Flash MX 2004。

用法
```
textAttrs.target
```

說明
屬性；字串；代表文字欄位的 target 屬性。這個屬性只能與靜態文字搭配使用。

範例
下列範例會在目前場景最上層圖層的第一個影格中，取得其文字欄位的 target 屬性，並在「輸出」面板中顯示：

```
fl.outputPanel.trace(fl.getDocumentDOM().getTimeline().layers[0].frames[0].elements[0].getTextAttr("targe
t"));
```

textAttrs.url

適用版本
Flash MX 2004。

用法
```
textAttrs.url
```

說明
屬性；字串；代表文字欄位的 URL 屬性。這個屬性只能與靜態文字搭配使用。

範例
下列範例會將所選文字欄位的 URL 設定為 http://www.adobe.com：

```
fl.getDocumentDOM().setElementTextAttr("url", "http://www.adobe.com");
```

第 44 章 TextRun 物件

適用版本
Flash MX 2004。

說明
TextRun 物件代表連續字元，所含特質與 TextAttrs 物件中的所有屬性皆相符。這個物件是 Text 物件的屬性 (text.textRuns)。

屬性摘要
除了與 Text 物件搭配使用的屬性之外，TextRun 物件還提供下列屬性：

屬性	說明
textRun.characters	字串；代表 TextRun 物件中包含的文字。
textRun.textAttrs	TextAttrs 物件；包含連續文字的特質。

textRun.textAttrs

適用版本
Flash MX 2004。

用法
```
textRun.textAttrs
```

說明
屬性；TextAttrs 物件；包含連續文字的特質。

範例
下列範例會在「輸出」面板的選取文字欄位中，顯示第一個連續字元的屬性。

```
var curTextAttrs = fl.getDocumentDOM().selection[0].textRuns[0].textAttrs;
for (var prop in curTextAttrs) {
    fl.trace(prop + " = " + curTextAttrs[prop]);
}
```

textRun.characters

適用版本
Flash MX 2004。

用法
```
textRun.characters
```

說明

屬性；TextRun 物件中包含的文字。

範例

下列範例會在「輸出」面板的選取文字欄位中，顯示組成第一個連續字元的字元：

```
fl.trace(fl.getDocumentDOM().selection[0].textRuns[0].characters);
```

第 45 章 Timeline 物件

適用版本

Flash MX 2004。

說明

Timeline 物件代表 Flash 時間軸，目前的文件可使用 fl.getDocumentDOM().getTimeline() 存取。這個方法會傳回目前正在編輯的場景或元件的時間軸。

您處理場景時，每個場景的時間軸都有索引值，可以讓目前的文件利用 fl.getDocumentDOM().timelines[i] 存取 (在這個範例中，i 是時間軸的索引值)。

當您使用 Timeline 物件的方法和屬性處理影格時，請記得，影格的值是從零開始的索引 (而非時間軸中連續影格的實際影格編號)。也就是說，第一個影格的影格索引為 0。

方法摘要

Timeline 物件可使用的方法如下：

方法	說明
timeline.addMotionGuide()	在目前圖層上面加入移動導引圖層，並將目前的圖層附加至新增的導引圖層。
timeline.addNewLayer()	將新圖層加入至文件，並將文件變成目前的圖層。
timeline.clearFrames()	從目前圖層的一個影格或影格範圍中，刪除所有內容。
timeline.clearKeyframes()	將關鍵影格轉換成一般影格，並在目前的圖層中刪除其內容
timeline.convertToBlankKeyframes()	在目前的圖層上，將影格轉換為空白關鍵影格。
timeline.convertToKeyframes()	在目前的圖層上，將影格範圍轉換為關鍵影格 (如果沒有指定影格，則轉換選取範圍)。
timeline.copyFrames()	在目前的圖層上，將影格範圍複製到剪貼簿。
timeline.copyLayers()	將 「時間軸」圖層的範圍複製到剪貼簿。
timeline.copyMotion()	從移動補間動畫或逐格動畫複製所選影格上的動畫，以套用至其它影格。
timeline.copyMotionAsAS3()	從移動補間動畫或逐格動畫複製所選影格上的動畫至剪貼簿，做為 ActionScript 3.0 程式碼。
timeline.createMotionObject()	在指定的開始與結束影格建立新的移動物件。
timeline.createMotionTween()	在目前的圖層上，將每個選取關鍵影格的 frame.tweenType 屬性設定為 motion，並於必要時，將每個影格的內容轉換為單一元件實體。
timeline.cutFrames()	從時間軸目前的圖層上剪下影格範圍，並將它們儲存到剪貼簿。
timeline.cutLayers()	將 「時間軸」圖層的範圍剪下並儲存到剪貼簿。
timeline.deleteLayer()	刪除圖層。
timeline.duplicateLayers()	重製選取的圖層或指定的圖層。
timeline.expandFolder()	展開或收合指定的資料夾。
timeline.findLayerIndex()	尋找具有指定名稱的圖層索引陣列。
timeline.getFrameProperty()	擷取選取影格的指定屬性值。

方法	說明
timeline.getGuidelines()	傳回 XML 字串，代表時間軸在水平導引線和垂直導引線的目前位置 (「檢視 > 導引線 > 顯示導引線」)。
timeline.getLayerProperty()	擷取選取圖層的指定屬性值。
timeline.getSelectedFrames()	擷取陣列中目前選取的影格。
timeline.getSelectedLayers()	擷取目前選取圖層的索引值 (從零開始)。
timeline.insertBlankKeyframe()	在指定的影格索引中插入空白關鍵影格；如果未指定索引，則會使用播放磁頭 / 選取範圍插入空白關鍵影格。
timeline.insertFrames()	在指定的影格編號上插入指定數目的影格。
timeline.insertKeyframe()	在指定的影格中插入關鍵影格。
timeline.pasteFrames()	將影格範圍從剪貼簿貼至指定的影格。
timeline.pasteLayers()	將複製的圖層貼到指定圖層索引上面的 「時間軸」。
timeline.pasteMotion()	將 timeline.copyMotion() 所擷取的移動影格範圍貼至時間軸。
timeline.removeFrames()	刪除影格。
timeline.removeMotionObject()	移除透過 timeline.createMotionObject() 建立的移動物件，並將影格轉換為靜態影格。
timeline.reorderLayer()	將第一個指定圖層移至第二個指定圖層的前面或後面。
timeline.reverseFrames()	反轉影格範圍。
timeline.selectAllFrames()	選取目前時間軸中的所有影格。
timeline.setFrameProperty()	設定選取影格的 Frame 物件屬性。
timeline.setGuidelines()	以指定的資訊取代時間軸的導引線。
timeline.setLayerProperty()	在所有選取的圖層上，將指定的屬性設定為指定的值。
timeline.setSelectedFrames()	在目前的圖層中選取影格範圍，或將選取的影格設定至傳遞到這個方法的選取範圍陣列。
timeline.setSelectedLayers()	設定要選取的圖層，並將指定的圖層變成目前的圖層。
timeline.showLayerMasking()	藉由鎖定遮色或被遮色的圖層，在編寫期間顯示圖層遮色片。
timeline.startPlayback()	如果目前未播放時間軸，則會啟動時間軸自動播放。
timeline.stopPlayback()	如果目前正在播放時間軸，則會停止時間軸自動播放。

屬性摘要

Timeline 物件可使用的屬性如下：

屬性	說明
timeline.currentFrame	目前播放磁頭位置上影格的索引 (從零開始)。
timeline.currentLayer	目前作用中圖層的索引 (從零開始)。
timeline.frameCount	唯讀；整數，代表這個時間軸內最長圖層中的影格數。
timeline.layerCount	唯讀；整數，代表指定時間軸中的圖層數。

屬性	說明
timeline.layers	唯讀：圖層物件的陣列。
timeline.libraryItem	唯讀屬性：指出時間軸是否屬於某個場景。
timeline.name	字串，代表目前時間軸的名稱。

timeline.addMotionGuide()

適用版本
Flash MX 2004。

用法
```
timeline.addMotionGuide()
```

參數
無。

傳回值
整數：代表新增導引圖層的索引 (從零開始)。如果目前圖層的類型不是 "Normal" 類型，Flash 會傳回 -1。

說明
方法：在目前圖層上面加入移動導引圖層，然後將目前的圖層附加到新增的導引圖層，並將目前的圖層轉換成類型為 "Guided" 的圖層。

這個方法只能在類型為 "Normal" 的圖層上產生作用。在類型為 "Folder"、"Mask"、"Masked"、"Guide" 或 "Guided" 的圖層，沒有任何作用。

範例
下列範例會在目前圖層上面加入移動導引圖層，並將目前的圖層轉換為 Guided：

```
fl.getDocumentDOM().getTimeline().addMotionGuide();
```

timeline.addNewLayer()

適用版本
Flash MX 2004。

用法
```
timeline.addNewLayer([name] [, layerType [, bAddAbove]])
```

參數
name 字串；指定新圖層的名稱。如果您忽略這個參數，則會將新的預設圖層名稱指定給新圖層 ("Layer n"，其中 n 為圖層的總數)。這個參數是選擇性參數。

layerType 字串，指定要新增的圖層類型。如果您省略這個參數，則會建立 "Normal" 類型圖層。這個參數是選擇性參數。可接受的值包括 "normal"、"guide"、"guided"、"mask"、"masked" 和 "folder"。

bAddAbove Boolean 值：如果設定為 true (預設值)，Flash 會將新圖層加到目前圖層上面；如果是 false，Flash 會將新圖層加到目前圖層下面。這個參數是選擇性參數。

傳回值
新增圖層的索引整數值 (從零開始)。

說明
方法；將新圖層加入至文件，並將文件變成目前的圖層。

範例
下列範例會使用 Flash 產生的預設名稱，將新圖層加入時間軸：

```
fl.getDocumentDOM().getTimeline().addNewLayer();
```

下列範例會將新圖層加入目前圖層的上面，並將它命名為 Folder1：

```
fl.getDocumentDOM().getTimeline().addNewLayer("Folder1", "folder", true);
```

timeline.clearFrames()

適用版本
Flash MX 2004。

用法

```
timeline.clearFrames([startFrameIndex [, endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引，定義要清除的影格範圍開頭。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引，定義要清除的影格範圍結尾。這個範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；從目前圖層的一個影格或影格範圍中，刪除所有內容。

範例
下列範例會清除影格 6 至 (但不包括) 影格 11 的影格 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().clearFrames(5, 10);
```

下列範例會清除影格 15：

```
fl.getDocumentDOM().getTimeline().clearFrames(14);
```

timeline.clearKeyframes()

適用版本
Flash MX 2004。

用法
```
timeline.clearKeyframes([startFrameIndex [, endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引，定義要清除的影格範圍開頭。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引，定義要清除的影格範圍結尾。這個範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；將關鍵影格轉換成一般影格，並在目前的圖層中刪除其內容

範例
下列範例會清除影格 5 至 (但不包括) 影格 10 的關鍵影格 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().clearKeyframes(4, 9);
```

下列範例會清除影格 15 中的關鍵影格，並將它轉換為一般影格：

```
fl.getDocumentDOM().getTimeline().clearKeyframes(14);
```

timeline.convertToBlankKeyframes()

適用版本
Flash MX 2004。

用法
```
timeline.convertToBlankKeyframes([startFrameIndex [, endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引；指定要轉換為關鍵影格的起始影格。如果您忽略 startFrameIndex，這個方法會轉換目前選取的影格。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定停止關鍵影格轉換的影格。轉換的影格範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；在目前的圖層上，將影格轉換為空白關鍵影格。

範例
下列範例會將影格 2 至 (但不包含) 影格 10 轉換成空白關鍵影格 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(1, 9);
```

下列範例會將影格 5 轉換為空白關鍵影格：

```
fl.getDocumentDOM().getTimeline().convertToBlankKeyframes(4);
```

timeline.convertToKeyframes()

適用版本
Flash MX 2004。

用法

```
timeline.convertToKeyframes([startFrameIndex [, endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引；指定要轉換為關鍵影格的第一個影格。如果您忽略 startFrameIndex，這個方法會轉換目前選取的影格。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定停止關鍵影格轉換的影格。轉換的影格範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；在目前的圖層上，將影格範圍轉換為關鍵影格 (如果沒有指定影格，則轉換選取範圍)。

範例
下列範例會將選取影格轉換為關鍵影格：

```
fl.getDocumentDOM().getTimeline().convertToKeyframes();
```

下列範例會將影格 2 至 (但不包含) 影格 10 的影格轉換為關鍵影格 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(1, 9);
```

下列範例會將影格 5 轉換為關鍵影格：

```
fl.getDocumentDOM().getTimeline().convertToKeyframes(4);
```

timeline.copyFrames()

適用版本
Flash MX 2004。

用法

```
timeline.copyFrames([startFrameIndex [, endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引；指定要複製的影格範圍開頭。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要停止複製的影格。複製的影格範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；在目前的圖層上，將影格範圍複製到剪貼簿。

範例
下列範例會將選取的影格複製到剪貼簿：

```
fl.getDocumentDOM().getTimeline().copyFrames();
```

下列範例會將影格 2 至 (但不包含) 影格 10 複製到剪貼簿 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().copyFrames(1, 9);
```

下列範例會將影格 5 複製到剪貼簿：

```
fl.getDocumentDOM().getTimeline().copyFrames(4);
```

timeline.copyLayers()

適用版本
Flash CS5.5 Professional。

用法
```
timeline.copyLayers([startLayerIndex [, endLayerIndex]])
```

參數
startLayerIndex 這是選擇性的。從零開始的索引；指定要複製的圖層範圍開頭。如果您忽略 startLayerIndex，這個方法會使用目前的選取範圍。

endLayerIndex 這是選擇性的。從零開始的索引；指定要停止複製的圖層。要複製的圖層範圍會延續至 (並包含) endLayerIndex。如果您只指定 startLayerIndex，endLayerIndex 會預設為 startLayerIndex 的值。

傳回值
無。

說明
方法；複製 「時間軸」中目前選取的圖層，或複製指定範圍內的圖層。可以提供選擇性引數，以便指定要複製的圖層或圖層範圍。

範例
下列範例是複製 「時間軸」中索引 2 到索引 7 的圖層：

```
fl.getDocumentDOM().getTimeline().copyLayers(2, 7);
```

請參閱
timeline.cutLayers()、timeline.pasteLayers()、timeline.duplicateLayers()

timeline.copyMotion()

適用版本
Flash CS3 Professional。

用法
```
timeline.copyMotion()
```

參數
無。

傳回值
無。

說明
方法：複製選取影格上的移動 (從移動補間動畫或從逐格動畫)。您可以接著使用 timeline.pasteMotion() 套用動畫至其它影格。

若要將動畫複製成可以貼入指令碼的文字 (程式碼)，請參閱 timeline.copyMotionAsAS3()。

範例
下列範例會從選取的單一影格或多個影格複製動畫：

```
fl.getDocumentDOM().getTimeline().copyMotion();
```

請參閱
timeline.copyMotionAsAS3()、timeline.pasteMotion()

timeline.copyMotionAsAS3()

適用版本
Flash CS3 Professional。

用法
```
timeline.copyMotionAsAS3()
```

參數
無。

傳回值
無。

說明
方法：從移動補間動畫或逐格動畫複製所選影格上的動畫至剪貼簿，做為 ActionScript 3.0 程式碼。然後就可以將此程式碼貼入指令碼。

若要以您可套用至其它影格的格式複製動畫，請參閱 timeline.copyMotion()。

範例
下列範例會從選取的單一影格或多個影格複製動畫至剪貼簿做為 ActionScript 3.0 程式碼：

```
fl.getDocumentDOM().getTimeline().copyMotionAsAS3();
```

請參閱
timeline.copyMotion()

timeline.createMotionObject()

適用版本
Flash Professional CS5。

用法
```
timeline.createMotionObject([startFrame [,endFrame])
```

參數
startFrame 指定建立移動物件的第一個影格。如果省略 startFrame，則此方法會使用目前的選取範圍；如果沒有任何選取範圍，則會移除所有圖層上目前播放磁頭的所有影格。這個參數是選擇性參數。

endFrame 指定停止建立移動物件的影格，影格的範圍最大可到 (但不含) endFrame。如果您僅指定 startFrame，則 endFrame 的預設值為 startFrame 值。這個參數是選擇性參數。

傳回值
無。

說明
方法：建立新的移動物件。此參數為選擇性參數。如有指定，則會在建立移動物件之前，將時間軸選取範圍設為指定的影格。

範例
下列範例會在最上層圖層的目前播放磁頭位置建立移動物件：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().createMotionObject();
```

下列範例會在第 5 個影格建立移動物件，且最多可延伸至 (但不含) 目前場景之最上層圖層的第 15 個影格：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().createMotionObject(5, 15);
```

timeline.createMotionTween()

適用版本
Flash MX 2004。

用法

```
timeline.createMotionTween([startFrameIndex [, endFrameIndex]])
```

參數

startFrameIndex 從零開始的索引；指定要建立移動補間動畫的起始影格。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要停止移動補間動畫的影格。影格範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值

無。

說明

方法；在目前的圖層上，將每個選取關鍵影格的 frame.tweenType 屬性設定為 motion，並於必要時，將每個影格的內容轉換為單一元件實體。這個屬性等於 Flash 編寫工具中的 「建立移動補間動畫」選單項目。

範例

下列範例會將影格 1 至 (但不包含) 影格 10 中的形狀轉換為圖像元件實體，並將 frame.tweenType 設定為 motion (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().createMotionTween(0, 9);
```

timeline.currentFrame

適用版本

Flash MX 2004。

用法

```
timeline.currentFrame
```

說明

屬性；目前播放磁頭位置上影格的索引 (從零開始)。

範例

下列範例會將目前時間軸的播放磁頭設定為影格 10 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().currentFrame = 9;
```

下列範例會將目前播放磁頭位置的值儲存於 curFrame 變數：

```
var curFrame = fl.getDocumentDOM().getTimeline().currentFrame;
```

timeline.currentLayer

適用版本

Flash MX 2004。

用法

```
timeline.currentLayer
```

說明

屬性；目前作用中圖層的索引 (從零開始)。值 0 指定最上層的圖層，值 1 指定最上層圖層下面的圖層，以此類推。

範例

下列範例會將最上層設定成作用中：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
```

下列範例會將目前作用中圖層的索引儲存於 curLayer 變數：

```
var curLayer = fl.getDocumentDOM().getTimeline().currentLayer;
```

timeline.cutFrames()

適用版本

Flash MX 2004。

用法

```
timeline.cutFrames([startFrameIndex [, endFrameIndex]])
```

參數

startFrameIndex 從零開始的索引；指定要剪下的影格範圍開頭。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要停止剪下的影格。影格範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值

無。

說明

方法；從時間軸目前的圖層上剪下影格範圍，並將它們儲存到剪貼簿。

範例

下列範例會從時間軸中剪下選取的影格並儲存到剪貼簿：

```
fl.getDocumentDOM().getTimeline().cutFrames();
```

下列範例會從時間軸中剪下影格 2 至 (但不包含) 影格 10，並將它們儲存到剪貼簿 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().cutFrames(1, 9);
```

下列範例會從時間軸中剪下影格 5 並儲存到剪貼簿：

```
fl.getDocumentDOM().getTimeline().cutFrames(4);
```

timeline.cutLayers()

適用版本
Flash CS5.5 Professional。

用法
```
timeline.cutLayers([startLayerIndex [, endLayerIndex]])
```

參數
startLayerIndex 這是選擇性的。從零開始的索引；指定要剪下的圖層範圍開頭。如果您忽略 startLayerIndex，這個方法會使用目前的選取範圍。

endLayerIndex 這是選擇性的。從零開始的索引；指定要停止剪下的圖層。要剪下的圖層範圍會延續至 (並包含) endLayerIndex。如果您只指定 startLayerIndex，endLayerIndex 會預設為 startLayerIndex 的值。

傳回值
無。

說明
方法；剪下 「時間軸」中目前選取的圖層，或複製指定範圍內的圖層。可以提供選擇性引數，以便指定要剪下的圖層或圖層範圍。

範例
下列範例是剪下 「時間軸」中索引 2 到索引 7 的圖層：

```
fl.getDocumentDOM().getTimeline().cutLayers(2, 7);
```

請參閱
timeline.copyLayers()、timeline.pasteLayers()、timeline.duplicateLayers()

timeline.deleteLayer()

適用版本
Flash MX 2004。

用法
```
timeline.deleteLayer([index])
```

參數
index 從零開始的索引值；指定要刪除的圖層。如果時間軸中只有一個圖層，這個方法不會有作用。這個參數是選擇性參數。

傳回值
無。

說明
方法；刪除圖層。如果這個圖層是資料夾，則會刪除這個資料夾內的所有圖層。如果您未指定圖層索引，Flash 會刪除目前選取的圖層。

範例
下列範例會刪除最上層起算的第二個圖層：

```
fl.getDocumentDOM().getTimeline().deleteLayer(1);
```

下列範例會刪除目前選取的圖層：

```
fl.getDocumentDOM().getTimeline().deleteLayer();
```

timeline.duplicateLayers()

適用版本
Flash CS5.5 Professional。

用法

```
timeline.duplicateLayers([startLayerIndex [, endLayerIndex]])
```

參數
startLayerIndex 這是選擇性的。從零開始的索引；指定要複製的圖層範圍開頭。它也會指定圖層，其上已貼上剪貼簿中的圖層。如果您忽略 startLayerIndex，這個方法會使用目前的圖層選取範圍。

endLayerIndex 這是選擇性的。從零開始的索引；指定要停止複製的圖層。要複製的圖層範圍會延續至 (並包含) endLayerIndex。如果您只指定 startLayerIndex，endLayerIndex 會預設為 startLayerIndex 的值。

傳回值
無。

說明
方法；重製 「時間軸」中目前選取的圖層，或複製指定範圍內的圖層。可以提供選擇性引數，以便指定要重製的圖層或圖層範圍。

範例
下列範例會重製 「時間軸」中目前選取的圖層：

```
fl.getDocumentDOM().getTimeline().duplicateLayers();
```

下列範例會重製圖層索引 2 上方，從索引 2 到索引 7 的圖層：

```
fl.getDocumentDOM().getTimeline().duplicatedLayers(2,7);
```

請參閱
timeline.copyLayers()、timeline.cutLayers()、timeline.pasteLayers()

timeline.expandFolder()

適用版本
Flash MX 2004。

用法

```
timeline.expandFolder(bExpand [, bRecurseNestedParents [, index]])
```

參數
bExpand Boolean 值；如果設定為 true，這個方法會展開資料夾；如果是 false，這個方法會收合資料夾。

bRecurseNestedParents Boolean 值；如果設定為 true，會根據 bExpand 參數，開啟或關閉指定資料夾內的所有圖層。這個參數是選擇性參數。

index 展開或收合的資料夾索引 (從零開始)。使用 -1 會套用至所有圖層 (您也必須將 bRecurseNestedParents 設定為 true)。這個屬性等於 Flash 編寫工具中的 「全部展開 / 全部收合」選單項目。這個參數是選擇性參數。

傳回值
無。

說明
方法；展開或收合指定的資料夾。如果您未指定圖層，這個方法會在目前的圖層上運作。

範例
下列範例會使用這個資料夾的結構：

```
Folder 1 ***
--layer 7
--Folder 2 ****
----Layer 5
```

下列範例只會展開資料夾 1：

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;
fl.getDocumentDOM().getTimeline().expandFolder(true);
```

下列範例只會展開資料夾 1 (假設資料夾 1 最後一次收合時，資料夾 2 是收合的；否則，資料夾 2 會是展開的)：

```
fl.getDocumentDOM().getTimeline().expandFolder(true, false, 0);
```

下列範例會在目前的時間軸中，收合所有的資料夾：

```
fl.getDocumentDOM().getTimeline().expandFolder(false, true, -1);
```

timeline.findLayerIndex()

適用版本
Flash MX 2004。

用法
```
timeline.findLayerIndex(name)
```

參數
name 字串，指定要搜尋的圖層名稱。

傳回值
指定圖層的索引值陣列。如果找不到指定的圖層，Flash 會傳回 undefined。

說明
方法；尋找具有指定名稱的圖層索引陣列。圖層索引是平面的，所以資料夾會被視為主索引的一部分。

範例

下列範例會在「輸出」面板中顯示所有名稱為 Layer 7 的圖層索引值：

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 7");
fl.trace(layerIndex);
```

下列範例會示範如何將這個方法的傳回值傳遞回 timeline.setSelectedLayers()：

```
var layerIndex = fl.getDocumentDOM().getTimeline().findLayerIndex("Layer 1");
fl.getDocumentDOM().getTimeline().setSelectedLayers(layerIndex[0], true);
```

timeline.frameCount

適用版本

Flash MX 2004。

用法

```
timeline.frameCount
```

說明

唯讀屬性：整數，代表這個時間軸內最長圖層中的影格數。

範例

下列範例會使用 countNum 變數儲存目前文件最長圖層中的影格數：

```
var countNum = fl.getDocumentDOM().getTimeline().frameCount;
```

timeline.getFrameProperty()

適用版本

Flash MX 2004。

用法

```
timeline.getFrameProperty(property [, startframeIndex [, endFrameIndex]])
```

參數

property 字串；指定要取得其值的屬性名稱。如需屬性的完整清單，請參閱 Frame 物件的屬性摘要。

startFrameIndex 從零開始的索引；指定要取得其值的起始影格編號。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要選取的影格範圍結尾。這個範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值

指定屬性的值；如果所有選取的影格都沒有相同的屬性值，則會傳回 undefined。

說明

方法；擷取選取影格的指定屬性值。

範例
下列範例會擷取目前文件最上層圖層中的第一個影格名稱，並在 「輸出」面板中顯示這個名稱：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 0, true);
var frameName = fl.getDocumentDOM().getTimeline().getFrameProperty("name");
fl.trace(frameName);
```

timeline.getGuidelines()

適用版本
Flash CS4 Professional。

用法
```
timeline.getGuidelines()
```

參數
無。

傳回值
XML 字串。

說明
方法：傳回 XML 字串，代表時間軸在水平導引線和垂直導引線的目前位置 (「檢視 > 導引線 > 顯示導引線」)。若要將這些導引線套用至時間軸，請使用 timeline.setGuidelines()。

範例
假設在第一個時間軸上有一些導引線，下列範例會在 「輸出」面板中將它們顯示為 XML 字串：

```
var currentTimeline = fl.getDocumentDOM().timelines[0];
fl.trace(currentTimeline.getGuidelines());
```

timeline.getLayerProperty()

適用版本
Flash MX 2004。

用法
```
timeline.getLayerProperty(property)
```

參數
property 字串；指定您要擷取其值的屬性名稱。如需屬性清單，請參閱 Frame 物件的屬性摘要。

傳回值
指定屬性的值。Flash 會檢視圖層的屬性來決定其類型。如果指定的圖層沒有相同的屬性值，Flash 會傳回 undefined。

說明
方法；擷取選取圖層的指定屬性值。

範例
下列範例會擷取目前文件的最上層圖層名稱，並在「輸出」面板中顯示名稱：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
var layerName = fl.getDocumentDOM().getTimeline().getLayerProperty("name");
fl.trace(layerName);
```

timeline.getSelectedFrames()

適用版本
Flash MX 2004。

參數
無。

傳回值
包含 3n 個整數的陣列，其中 n 代表選取的區域數。每個群組中的第一個整數是圖層索引，第二個整數是這個選取範圍開頭的起始影格，第三個整數指定這個選取範圍的結尾影格。這個選取範圍不包括結尾影格。

說明
方法；擷取陣列中目前選取的影格。

範例
下列範例會在最上層圖層為目前圖層的情況下，在「輸出」面板中顯示 0,5,10,0,20,25：

```
var timeline = fl.getDocumentDOM().getTimeline();
timeline.setSelectedFrames(5,10);
timeline.setSelectedFrames(20,25,false);
var theSelectedFrames = timeline.getSelectedFrames();
fl.trace(theSelectedFrames);
```

請參閱
timeline.setSelectedFrames()

timeline.getSelectedLayers()

適用版本
Flash MX 2004。

參數
無。

傳回值
選取圖層的索引值 (從零開始) 陣列。

說明
方法：取得目前選取圖層的索引值 (從零開始)。

範例
下列範例會在 「輸出」面板中顯示 1,0：

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
var layerArray = fl.getDocumentDOM().getTimeline().getSelectedLayers();
fl.trace(layerArray);
```

請參閱
timeline.setSelectedLayers()

timeline.insertBlankKeyframe()

適用版本
Flash MX 2004。

用法
```
timeline.insertBlankKeyframe([frameNumIndex])
```

參數
frameNumIndex 從零開始的索引：指定要插入關鍵影格的影格。如果您忽略 frameNumIndex，這個方法會使用目前的播放磁頭影格編號。這個參數是選擇性參數。

如果指定或選取的影格是一般影格，會在這個影格中插入關鍵影格。例如，如果影格範圍為 10 個影格，影格的編號是 1-10，而且您選取影格 5，這個方法會將影格 5 變成空白關鍵影格，而且這個影格範圍的長度仍然是 10 個影格。如果選取影格 5，而且影格 5 是關鍵影格，它旁邊的是一般影格，則這個方法會在影格 6 插入一個空白關鍵影格。如果影格 5 是關鍵影格，而且旁邊的影格已經是關鍵影格，則不會插入關鍵影格，但是播放磁頭會移至影格 6。

傳回值
無。

說明
方法：將空白關鍵影格插入指定的影格索引；如果未指定索引，這個方法會使用播放磁頭 / 選取範圍插入空白關鍵影格。請參閱 timeline.insertKeyframe()。

範例
下列範例會在影格 20 插入空白關鍵影格 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe(19);
```

下列範例會在目前選取的影格中插入空白關鍵影格 (如果未選取影格，則在播放磁頭的位置插入)：

```
fl.getDocumentDOM().getTimeline().insertBlankKeyframe();
```

timeline.insertFrames()

適用版本
Flash MX 2004。

用法
```
timeline.insertFrames([numFrames [, bAllLayers [, frameNumIndex]]])
```

參數
numFrames 整數；指定要插入的影格數目。如果您忽略這個參數，這個方法會在目前圖層的目前選取範圍中插入影格。這個參數是選擇性參數。

bAllLayers Boolean 值；如果設定為 true (預設值)，則這個方法會將 numFrames 參數中指定的影格數，插入所有圖層中；如果設定為 false，這個方法會將影格插入目前的圖層。這個參數是選擇性參數。

frameNumIndex 從零開始的索引；指定要插入新影格的影格。這個參數是選擇性參數。

傳回值
無。

說明
方法；在指定索引中插入指定的影格數。

如果未指定參數，這個方法的運作如下：

- 如果選取一個或多個影格，這個方法會在目前圖層第一個選取影格的位置中，插入選取的影格數。也就是說，如果選取影格 6 到 10 (總共五個影格)，這個方法會在包含選取影格的圖層上，將五個影格加入影格 6。
- 如果未選取影格，這個方法會在所有圖層的目前影格中插入一個影格。

  如果指定參數，這個方法的運作如下：

- 如果只指定 numFrames，會在目前圖層的目前影格中，插入指定的影格數。
- 如果指定 numFrames，而且 bAllLayers 是 true，會在所有圖層的目前影格中，插入指定的影格數。
- 如果三個參數都指定，會在指定索引中 (frameIndex) 插入指定的影格數；傳送給 bAllLayers 的值，會決定只將這些影格加入至目前的圖層，或加入所有的圖層。

  如果指定或選取的影格是一般影格，會在這個影格中插入影格。例如，如果影格範圍是 10 個影格，影格的編號是 1-10，而且您選取影格 5 (或將值 4 傳送給 frameIndex)，這個方法會在影格 5 加入影格，而且這個影格範圍的長度會變成 11 個影格。如果選取影格 5 而且它是關鍵影格，則無論影格 5 旁邊的影格是否為關鍵影格，這個方法都會在影格 6 插入影格。

範例
下列範例會在目前圖層的目前選取範圍插入一或數個影格，依照選取範圍而定：

```
fl.getDocumentDOM().getTimeline().insertFrames();
```

下列範例會在所有圖層的目前影格中，插入五個影格：

```
fl.getDocumentDOM().getTimeline().insertFrames(5);
```

備註：如果有數個圖層具有影格，而且使用上述命令在某個圖層中選取一個影格，則 Flash 只會在選取的圖層中插入影格。如果您有數個圖層，而且其中沒有影格，則 Flash 會在所有圖層中插入影格。

下列範例只會在目前的圖層中插入三個影格：

```
fl.getDocumentDOM().getTimeline().insertFrames(3, false);
```

下列範例會從第一個影格開始，在所有圖層中插入四個影格：

```
fl.getDocumentDOM().getTimeline().insertFrames(4, true, 0);
```

timeline.insertKeyframe()

適用版本
Flash MX 2004。

用法
```
timeline.insertKeyframe([frameNumIndex])
```

參數
frameNumIndex 從零開始的索引；指定要在目前圖層中插入關鍵影格的影格索引。如果您忽略 frameNumIndex，這個方法會使用目前播放磁頭或選取影格的影格編號。這個參數是選擇性參數。

傳回值
無。

說明
方法；在指定的影格中插入關鍵影格。如果您忽略這個參數，這個方法會使用播放磁頭或選取位置插入影格。

除了插入的關鍵影格包含其轉換的影格內容 (也就是說，關鍵影格不是空白的)，這個方法與 timeline.insertBlankKeyframe() 的作用相同。

範例
下列範例會在播放磁頭或選取位置中，插入一個關鍵影格：

```
fl.getDocumentDOM().getTimeline().insertKeyframe();
```

下列範例會在第二個圖層的影格 10 中，插入一個關鍵影格 (請記得，索引值與影格或圖層編號值不同)：

```
fl.getDocumentDOM().getTimeline().currentLayer = 1;
fl.getDocumentDOM().getTimeline().insertKeyframe(9);
```

timeline.layerCount

適用版本
Flash MX 2004。

用法
```
timeline.layerCount
```

說明
唯讀屬性；整數；代表指定時間軸中的圖層數。

範例
下列範例會使用 NumLayer 變數，將圖層數目儲存在目前的場景中：

```
var NumLayer = fl.getDocumentDOM().getTimeline().layerCount;
```

timeline.layers

適用版本
Flash MX 2004。

用法
```
timeline.layers
```

說明
唯讀屬性：圖層物件的陣列。

範例
下列範例會使用 currentLayers 變數，將圖層物件的陣列儲存在目前的文件中：

```
var currentLayers = fl.getDocumentDOM().getTimeline().layers;
```

timeline.libraryItem

適用版本
Flash Professional CS5。

用法
```
timeline.libraryItem
```

說明
唯讀屬性：如果時間軸的 libraryItem 屬性為空值，則時間軸屬於場景。如果它不是空值，您可以將它視為 LibraryItem 物件。

範例
下列範例將在 libraryItem 的值不是 Null 時，輸出 libraryItem 的名稱，並在 librayItem 為 Null 時，輸出場景的名稱：

```
var item = fl.getDocumentDOM().getTimeline().libraryItem;
if (item)
fl.trace("libraryItem name: " + item.name);
else
fl.trace("scene name: " + fl.getDocumentDOM().getTimeline().name);
```

timeline.name

適用版本
Flash MX 2004。

用法
```
timeline.name
```

說明
屬性：字串；指定目前時間軸的名稱。這個名稱是目前正在編輯的場景、螢幕 (幻燈片或表單) 或元件的名稱。

範例
下列範例會擷取第一個場景名稱：

```
var sceneName = fl.getDocumentDOM().timelines[0].name;
```

下列範例會將第一個場景名稱設定為 FirstScene：

```
fl.getDocumentDOM().timelines[0].name = "FirstScene";
```

timeline.pasteFrames()

適用版本
Flash MX 2004。

用法

```
timeline.pasteFrames([startFrameIndex [, endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引；指定要貼上的影格範圍開頭。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要停止貼上影格的影格。這個方法會貼至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；將影格範圍從剪貼簿貼至指定的影格。

範例
下列範例會將剪貼簿上的影格，貼至目前選取的影格或播放磁頭位置：

```
fl.getDocumentDOM().getTimeline().pasteFrames();
```

下列範例會將剪貼簿上的影格，從影格 2 貼至 (但不包含) 影格 10 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().pasteFrames(1, 9);
```

下列範例會貼上剪貼簿上從影格 5 開始的影格：

```
fl.getDocumentDOM().getTimeline().pasteFrames(4);
```

timeline.pasteLayers()

適用版本
Flash CS5.5 Professional。

用法

```
timeline.pasteLayers([layerIndex])
```

參數
layerIndex 這是選擇性的。從零開始的索引；指定圖層，其上已貼上剪貼簿中的圖層。如果您忽略 layerIndex，這個方法會使用目前的選取範圍。

傳回值
指示貼上之圖層的最低圖層索引的整數。

說明
方法；將之前剪下或複製的圖層貼到目前選取的圖層上方，或貼到指定之圖層索引的上方。如果指定的圖層是資料夾，圖層就會貼到資料夾中。傳回貼上之圖層的最低圖層索引。此動作不會影響系統剪貼簿。

範例
下列範例會將圖層從圖層剪貼簿貼到 「時間軸」中目前選取的圖層上方。

```
fl.getDocumentDOM().getTimeline().pasteLayers();
```

下列範例會在圖層索引 2 上方，從圖層剪貼簿貼上圖層：

```
fl.getDocumentDOM().getTimeline().pasteLayers(2);
```

請參閱
timeline.cutLayers(), timeline.copyLayers()、timeline.duplicateLayers()

timeline.pasteMotion()

適用版本
Flash CS3 Professional。

用法
```
timeline.pasteMotion()
```

參數
無。

傳回值
無。

說明
方法；將 timeline.copyMotion() 所擷取的移動影格範圍貼至 「時間軸」。必要時，現有的影格會置換 (移至右邊)，以挪出空間給貼上的影格。

範例
下列範例會將剪貼簿上的動畫，貼至目前選取的影格或播放磁頭位置，以置換貼上影格右方的影格：

```
fl.getDocumentDOM().getTimeline().pasteMotion();
```

請參閱
timeline.copyMotion()

timeline.removeFrames()

適用版本
Flash MX 2004。

用法
```
timeline.removeFrames([startFrameIndex [,endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引；指定要開始移除影格的第一個影格。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍；如果沒有選擇，則會移除所有圖層上目前播放磁頭的所有影格。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要停止移除影格的影格；這個影格範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；刪除影格。

範例
下列範例會刪除目前場景中，最上層圖層的影格 5 至 (但不包含) 影格 10 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().removeFrames(4, 9);
```

下列範例會刪除目前場景最上層圖層的影格 8：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().removeFrames(7);
```

timeline.removeMotionObject()

適用版本
Flash Professional CS5。

用法
```
timeline.removeMotionObject([startFrame [,endFrame])
```

參數
startFrame 指定開始移除移動物件的第一個影格。如果省略 startFrame，則此方法會使用目前的選取範圍；如果沒有任何選取範圍，則會移除所有圖層上目前播放磁頭的所有影格。這個參數是選擇性參數。

endFrame 指定停止移除移動物件的影格，影格的範圍最大可到 (但不含) endFrame。如果您僅指定 startFrame，則 endFrame 的預設值為 startFrame 值。這個參數是選擇性參數。

傳回值
無。

說明
方法；移除移動物件，並將影格轉換回靜態影格。此參數為選擇性參數。如有指定，則會在移除移動物件之前，將時間軸選取範圍設為指定的影格。

範例
下列範例會刪除所有移動物件，並在最上層圖層的目前播放磁頭位置將影格轉換回靜態影格：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().removeMotionObject();
```

下列範例會刪除從目前場景最上層圖層的第 5 個影格到 (但不含) 第 15 個影格的移動物件：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().removeMotionObject(5, 15);
```

請參閱
第 437 頁「timeline.createMotionObject()」

timeline.reorderLayer()

適用版本
Flash MX 2004。

用法
```
timeline.reorderLayer(layerToMove, layerToPutItBy [, bAddBefore])
```

參數
layerToMove 從零開始的索引；指定要移動的圖層。

layerToPutItBy 從零開始的索引；指定要移動圖層旁邊的圖層。例如，如果您將 layerToMove 指定為 1，而且將 layerToPutItBy 指定為 0，則第二個圖層會放置在第一個圖層旁邊。

bAddBefore 指定將圖層移至 layerToPutItBy 前面或後面。如果您指定 false，則圖層會移至 layerToPutItBy 後面。預設值為 true。這個參數是選擇性參數。

傳回值
無。

說明
方法；將第一個指定圖層移至第二個指定圖層的前面或後面。

範例
下列範例會將索引 2 的圖層，移至最上層 (索引 0 的圖層上面)：

```
fl.getDocumentDOM().getTimeline().reorderLayer(2, 0);
```

下列範例會將索引 3 的圖層，放置在索引 5 的圖層後面：

```
fl.getDocumentDOM().getTimeline().reorderLayer(3, 5, false);
```

timeline.reverseFrames()

適用版本
Flash MX 2004。

用法
```
timeline.reverseFrames([startFrameIndex [, endFrameIndex]])
```

參數
startFrameIndex 從零開始的索引；指定要開始反轉的第一個影格。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要停止反轉的影格；這個影格範圍會延續至 (但不包含) endFrameIndex。如果您只指定 startFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；反轉影格範圍。

範例
下列範例會反轉目前選取影格的位置：

```
fl.getDocumentDOM().getTimeline().reverseFrames();
```

下列範例會反轉影格 10 至 (但不包含) 影格 15 的影格 (請記得，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().reverseFrames(9, 14);
```

timeline.selectAllFrames()

適用版本
Flash MX 2004。

用法
```
timeline.selectAllFrames()
```

參數
無。

傳回值
無。

說明
方法；選取目前時間軸中的所有影格。

範例
下列範例會選取目前時間軸中的所有影格。

```
fl.getDocumentDOM().getTimeline().selectAllFrames();
```

timeline.setFrameProperty()

適用版本
Flash MX 2004。

用法
```
timeline.setFrameProperty(property, value [, startFrameIndex [, endFrameIndex]])
```

參數
property 字串；指定要修改的屬性名稱。如需屬性和值的完整清單，請參閱 Frame 物件的屬性摘要。

不得使用此方法設定唯讀屬性值，如 frame.duration 和 frame.elements。

value 指定您用來設定屬性的值。若要決定適合的值和類型，請參閱 Frame 物件的屬性摘要。

startFrameIndex 從零開始的索引；指定要修改的起始影格編號。如果您忽略 startFrameIndex，這個方法會使用目前的選取範圍。這個參數是選擇性參數。

endFrameIndex 從零開始的索引；指定要停止的第一個影格。影格範圍會延續至 (但不包含) endFrameIndex。如果您指定 startFrameIndex 但忽略 endFrameIndex，endFrameIndex 會預設為 startFrameIndex 的值。這個參數是選擇性參數。

傳回值
無。

說明
方法；設定選取影格的 Frame 物件屬性。

範例
下列範例會將 ActionScript stop() 命令，指定給目前文件最上層圖層的第一個影格：

```
fl.getDocumentDOM().getTimeline().currentLayer = 0;
fl.getDocumentDOM().getTimeline().setSelectedFrames(0,0,true);
fl.getDocumentDOM().getTimeline().setFrameProperty("actionScript", "stop();");
```

下列範例會從目前圖層的影格 2 至 (但不包含) 影格 5，設定補間動畫 (請記得，索引值與影格編號值不同)：

```
var doc = fl.getDocumentDOM();
doc.getTimeline().setFrameProperty("tweenType","motion",1,4);
```

timeline.setGuidelines()

適用版本
Flash CS4 Professional。

用法
```
timeline.setGuidelines(xmlString)
```

參數
xmlString XML 字串，包含要套用之導引線的資訊。

傳回值
如果成功套用導引線，就會傳回 Boolean 值 true，否則便傳回 false。

說明
方法：將時間軸的導引線 (「檢視 > 導引線 > 顯示導引線」) 取代為 xmlString 中指定的資訊。若要擷取可傳遞至此方法的 XML 字串，請使用 timeline.getGuidelines()。

若要檢視剛設定的導引線，您可能必須先隱藏導引線，然後再檢視它們。

範例
下列範例會將某個 FLA 檔中的導引線套用至另一個 FLA 檔：

```
var doc0 = fl.documents[0];
var guides0 = doc0.timelines[0].getGuidelines();
var doc1 = fl.documents[1];
doc1.timelines[0].setGuidelines(guides0);
```

timeline.setLayerProperty()

適用版本
Flash MX 2004。

用法
```
timeline.setLayerProperty(property, value [, layersToChange])
```

參數
property 字串；指定要設定的屬性。如需屬性清單，請參閱：第 296 頁 「Layer 物件」。

value 您要用來設定屬性的值。使用的值類型，要與在 Layer 物件上設定屬性的類型相同。

layersToChange 字串；可以識別應該修改的圖層。可接受的值為 "selected"、"all" 和 "others"。如果您忽略這個參數，預設的值為 "selected"。這個參數是選擇性參數。

傳回值
無。

說明
方法；在所有選取的圖層上，將指定的屬性設定為指定的值。

範例
下列範例會將選取的圖層變成不可見的：

```
fl.getDocumentDOM().getTimeline().setLayerProperty("visible", false);
```

下列範例會將選取圖層的名稱設定為 selLayer：

```
fl.getDocumentDOM().getTimeline().setLayerProperty("name", "selLayer");
```

timeline.setSelectedFrames()

適用版本
Flash MX 2004。

用法
```
timeline.setSelectedFrames(startFrameIndex, endFrameIndex [, bReplaceCurrentSelection])
timeline.setSelectedFrames(selectionList [, bReplaceCurrentSelection])
```

參數
startFrameIndex 從零開始的索引；指定要設定的起始影格。

endFrameIndex 從零開始的索引；指定選取範圍的結尾；endFrameIndex 是要選取的範圍中，最後一個影格之後的影格。

bReplaceCurrentSelection Boolean 值；如果設定為 true，則在選取指定的影格之前，會先取消選取目前已選取的影格。預設值為 true。

selectionList 三個整數的陣列；如 timeline.getSelectedFrames() 傳回值。

傳回值
無。

說明
方法；在目前的圖層中選取影格範圍，或將選取的影格設定至傳遞到這個方法的選取範圍陣列。

範例
下列範例會顯示兩種方法來選取最上層圖層的影格 1 至 (但不包含) 影格 10，然後在相同圖層上，將影格 12 至 (但不包含) 影格 15 加入至目前的選取範圍 (請注意，索引值與影格編號值不同)：

```
fl.getDocumentDOM().getTimeline().setSelectedFrames(0, 9);
fl.getDocumentDOM().getTimeline().setSelectedFrames(11, 14, false);
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 0, 9]);
fl.getDocumentDOM().getTimeline().setSelectedFrames([0, 11, 14], false);
```

下列範例會先將選取影格的陣列儲存於 savedSelectionList 變數，然後在命令或使用者互動變更選取範圍之後，在程式碼中使用這個陣列重新選取這些影格：

```
var savedSelectionList = fl.getDocumentDOM().getTimeline().getSelectedFrames();
// Do something that changes the selection.
fl.getDocumentDOM().getTimeline().setSelectedFrames(savedSelectionList);
```

請參閱
timeline.getSelectedFrames()

timeline.setSelectedLayers()

適用版本
Flash MX 2004。

用法
```
timeline.setSelectedLayers(index [, bReplaceCurrentSelection])
```

參數
index 要選取的圖層索引 (從零開始)。

bReplaceCurrentSelection Boolean 值；如果設定為 true，這個方法會取代目前的選取範圍；如果是 false，這個方法會擴充目前的選取範圍。預設值為 true。這個參數是選擇性參數。

傳回值
無。

說明
方法；設定要選取的圖層，並將指定的圖層變成目前的圖層。此外，選取圖層也代表這個圖層內的所有影格都會被選取。

範例
下列範例會選取最上層的圖層：

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(0);
```

下列範例會將下一個圖層加入選取範圍：

```
fl.getDocumentDOM().getTimeline().setSelectedLayers(1, false);
```

請參閱
timeline.getSelectedLayers()

timeline.showLayerMasking()

適用版本
Flash MX 2004。

用法
```
timeline.showLayerMasking([layer])
```

參數
layer 遮色或被遮色圖層的索引 (從零開始)；用以在編寫期間顯示遮色片。這個參數是選擇性參數。

傳回值
無。

說明
方法；藉由鎖定遮色或被遮色的圖層，在編寫期間顯示圖層遮色片。如果未指定圖層，這個方法會使用目前的圖層。如果在類型不是 Mask 或 Masked 的圖層上使用這個方法，Flash 會在 「輸出」面板中顯示錯誤。

範例
下列範例會指定在編寫期間應該顯示第一個圖層的圖層遮色片。

```
fl.getDocumentDOM().getTimeline().showLayerMasking(0);
```

timeline.startPlayback()

適用版本
Flash Professional CS5。

用法
```
timeline.startPlayback()
```

傳回值
無。

說明
方法；如果目前正在播放時間軸，則會啟動時間軸自動播放。此方法可以和 SWF 面板搭配使用，以控制編寫環境中的時間軸播放。

範例
下列範例會啟動時間軸的播放。

```
fl.getDocumentDOM().getTimeline().startPlayback();
```

timeline.stopPlayback()

適用版本
Flash Professional CS5。

用法
```
timeline.stopPlayback()
```

傳回值
無。

說明
方法；如果目前正在播放時間軸，則會停止時間軸自動播放。此方法可以和 SWF 面板搭配使用，以控制編寫環境中的時間軸播放。

範例
下列範例會停止時間軸的播放。

```
fl.getDocumentDOM().getTimeline().stopPlayback();
```

第 46 章 ToolObj 物件

適用版本
Flash MX 2004。

說明
ToolObj 物件代表 「工具」面板中的個別工具。若要存取 ToolObj 物件，請使用 Tools 物件的屬性：tools.toolObjs 陣列或 tools.activeTool。

方法摘要
ToolObj 物件可使用的方法如下。

備註：下列方法只能在建立可擴充工具時使用。

方法	說明
toolObj.enablePIControl()	啟用或停用 「屬性」檢測器中的指定控制項。只能在建立可擴充工具時使用。
toolObj.setIcon()	識別 PNG 檔，要在 「Flash 工具」面板中做為工具圖示。
toolObj.setMenuString()	設定彈出式選單中出現的字串，做為工具名稱。
toolObj.setOptionsFile()	將 XML 檔與工具建立關聯。
toolObj.setPI()	設定啟用工具時，要使用的特定 「屬性」檢測器。
toolObj.setToolName()	指定 「工具」面板的設定工具名稱。
toolObj.setToolTip()	設定當滑鼠放置在工具圖示上時，要出現的工具提示。
toolObj.showPIControl()	顯示或隱藏 「屬性」檢測器中的控制項。
toolObj.showTransformHandles()	在一個可擴充工具 JavaScript 檔的 configureTool() 方法中進行呼叫，以指出工具作用中時，應該要出現自由變形控制點。

屬性摘要
ToolObj 物件可使用的屬性如下：

屬性	說明
toolObj.depth	整數；指定 「工具」面板的彈出式選單中的工具深度。
toolObj.iconID	整數；指定工具的資源 ID。
toolObj.position	唯讀；整數；指定 「工具」面板中工具的位置。

toolObj.depth

適用版本
Flash MX 2004。

用法
```
toolObj.depth
```

說明
唯讀屬性：整數；指定 「工具」面板的彈出式選單中的工具深度。此屬性僅在建立可擴充工具時使用。

範例
下列範例會指定工具具有 1 的深度，代表 「工具」面板的工具下具有一個層級：

```
fl.tools.activeTool.depth = 1;
```

toolObj.enablePIControl()

適用版本
Flash MX 2004。

用法
```
toolObj.enablePIControl(control, bEnable)
```

參數
control 字串；指定要啟用或停用的控制項名稱。合法值是依照這個工具呼叫的 「屬性」檢測器而定 (請參閱 toolObj.setPI())。

形狀 「屬性」檢測器具有下列控制項：

stroke	fill

文字 「屬性」檢測器具有下列控制項：

type	font	pointsize
color	bold	italic
direction	alignLeft	alignCenter
alignRight	alignJustify	spacing
position	autoKern	small
rotation	format	lineType
selectable	html	border
deviceFonts	varEdit	options
link	maxChars	target

影片 「屬性」檢測器具有下列控制項：

size	publish	background
framerate	player	profile

bEnable Boolean 值；判斷啟用 (true) 或停用 (false) 控制項。

傳回值
無。

說明
方法：啟用或停用 「屬性」檢測器中的指定控制項。只能在建立可擴充工具時使用。

範例
可擴充工具 JavaScript 檔中的下列命令，會將 Flash 設定為不顯示這個工具的 「屬性」檢測器中的筆畫選項：

```
theTool.enablePIControl("stroke",false);
```

toolObj.iconID

適用版本
Flash MX 2004。

用法
```
toolObj.iconID
```

說明
唯讀屬性：具有 -1 值的整數。此屬性僅在建立可擴充工具時使用。iconID 值若為 -1，代表 Flash 不會為工具尋找圖示，而是由工具的指令碼指定顯示在 「工具」面板中的圖示；請參閱 toolObj.setIcon()。

範例
下列範例指定 -1 的值 (目前工具的圖示 ID) 給 toolIconID 變數：

```
var toolIconID = fl.tools.activeTool.iconID
```

toolObj.position

適用版本
Flash MX 2004。

用法
```
toolObj.position
```

說明
唯讀屬性：整數；指定 「工具」面板中工具的位置。此屬性僅在建立可擴充工具時使用。

範例
工具 JavaScript 檔的 mouseDown() 方法中的下列命令，會在 「輸出」面板中，以整數顯示 「工具」面板中的工具位置：

```
myToolPos = fl.tools.activeTool.position;
fl.trace(myToolPos);
```

toolObj.setIcon()

適用版本
Flash MX 2004。

用法
```
toolObj.setIcon(file)
```

參數
file 字串；指定要做為圖示的 PNG 檔名稱。PNG 檔必須放置在和 JSFL 檔相同的資料夾中。

傳回值
無。

說明
方法；識別要在 「工具」面板中做為工具圖示的 PNG 檔。此方法僅使用在建立可擴充的工具時。

範例
下列範例會指定 PolyStar.png 檔中的影像，做為工具 (名為 PolyStar) 的圖示。此程式碼取自樣本 PolyStar.jsfl 檔 (請參閱第 11 頁 「樣本 PolyStar 工具」)：
```
theTool = fl.tools.activeTool;
theTool.setIcon("PolyStar.png");
```

toolObj.setMenuString()

適用版本
Flash MX 2004。

用法
```
toolObj.setMenuString(menuStr)
```

參數
menuStr 字串；指定彈出式選單中出現的名稱，做為工具名稱。

傳回值
無。

說明
方法；設定彈出式選單中出現的字串，做為工具名稱。此方法僅使用在建立可擴充的工具時。

範例
下列範例會指定名為 theTool 的工具，在工具的彈出式選單中顯示名稱 "PolyStar Tool"。此程式碼取自樣本 PolyStar.jsfl 檔 (請參閱第 11 頁 「樣本 PolyStar 工具」)：
```
theTool = fl.tools.activeTool;
theTool.setMenuString("PolyStar Tool");
```

toolObj.setOptionsFile()

適用版本
Flash MX 2004。

用法
```
toolObj.setOptionsFile(xmlFile)
```

參數
xmlFile 字串；指定具有工具選項說明的 XML 檔名稱。XML 檔必須放置在和 JSFL 檔相同的資料夾中。

傳回值
無。

說明
方法；將 XML 檔與工具建立關聯。這個檔案指定要在強制回應面板出現的選項，此面板是由「屬性」檢測器中的「選項」按鈕叫用的。您通常會在 JSFL 檔案內的 configureTool() 函數中使用此方法。請參閱 configureTool()。

例如，PolyStar.xml 檔會指定與 Polygon 工具關聯的三個選項：

```
<properties>
    <property name="Style"
        variable="style"
        list="polygon,star"
        defaultValue="0"
        type="Strings"/>

    <property name="Number of Sides"
        variable="nsides"
        min="3"
        max="32"
        defaultValue="5"
        type="Number" />

    <property name="Star point size"
        variable="pointParam"
        min="0"
        max="1"
        defaultValue=".5"
        type="Double" />

</properties>
```

範例
下列範例會指定名為 PolyStar.xml 的檔案，與目前作用中的工具產生關聯。此程式碼取自樣本 PolyStar.jsfl 檔 (請參閱第 11 頁「樣本 PolyStar 工具」)：

```
theTool = fl.tools.activeTool;
theTool.setOptionsFile("PolyStar.xml");
```

toolObj.setPI()

適用版本
Flash MX 2004。

用法
```
toolObj.setPI(pi)
```

參數
pi 字串；指定要為這個工具呼叫的 「屬性」檢測器。

傳回值
無。

說明
方法；指定工具作用中時，要使用的 「屬性」檢測器。此方法僅使用在建立可擴充的工具時。可接受的值為 "shape" (預設值)、"text" 和 "movie"。

範例
下列範例會指定工具啟用時使用形狀 「屬性」檢測器。此程式碼取自樣本 PolyStar.jsfl 檔 (請參閱第 11 頁 「樣本 PolyStar 工具」)：

```
theTool = fl.tools.activeTool;
theTool.setPI("shape");
```

toolObj.setToolName()

適用版本
Flash MX 2004。

用法
```
toolObj.setToolName(name)
```

參數
name 字串；指定工具名稱。

傳回值
無。

說明
方法；指定 「工具」面板的設定工具名稱。此方法僅使用在建立可擴充的工具時。這個名稱僅供 XML 配置檔使用，Flash 會讀取 XML 配置檔建構 「工具」面板。這個名稱不會顯示在 Flash 使用者介面上。

範例
下列範例會將名稱 polystar 指定給名為 theTool 的工具。此程式碼取自樣本 PolyStar.jsfl 檔 (請參閱第 11 頁 「樣本 PolyStar 工具」)：

```
theTool = fl.tools.activeTool;
theTool.setToolName("polystar");
```

toolObj.setToolTip()

適用版本
Flash MX 2004。

用法
```
toolObj.setToolTip(toolTip)
```

參數
toolTip 字串；指定工具要使用的工具提示。

傳回值
無。

說明
方法；設定當滑鼠放置在工具圖示上時，要出現的工具提示。此方法僅使用在建立可擴充的工具時。

範例
下列範例會將工具的工具提示指定為 PolyStar Tool。此程式碼取自樣本 PolyStar.jsfl 檔 (請參閱第 11 頁 「樣本 PolyStar 工具」)：
```
theTool = fl.tools.activeTool;
theTool.setToolTip("PolyStar Tool");
```

toolObj.showPIControl()

適用版本
Flash MX 2004。

用法
```
toolObj.showPIControl(control, bShow)
```

參數
control 字串；指定要顯示或隱藏的控制項名稱。此方法僅使用在建立可擴充的工具時。有效值是依照這個工具呼叫的 「屬性」檢測器而定 (請參閱 toolObj.setPI())。

形狀 「屬性」檢測器具有下列控制項：

stroke	fill

文字 「屬性」檢測器具有下列控制項：

type	font	pointsize
color	bold	italic
direction	alignLeft	alignCenter
alignRight	alignJustify	spacing
position	autoKern	small
rotation	format	lineType
selectable	html	border
deviceFonts	varEdit	options
link	maxChars	target

影片 「屬性」檢測器具有下列控制項：

size	publish	background
framerate	player	profile

bShow Boolean 值；會判斷顯示或隱藏指定的控制項 (true 會顯示控制項；false 會隱藏控制項)。

傳回值
無。

說明
方法；顯示或隱藏 「屬性」檢測器中的控制項。此方法僅使用在建立可擴充的工具時。

範例
可擴充工具 JavaScript 檔中的下列命令，會將 Flash 設定為不顯示這個工具的 「屬性」檢測器中的填色選項：

```
fl.tools.activeTool.showPIControl("fill", false);
```

toolObj.showTransformHandles()

適用版本
Flash MX 2004。

用法

```
toolObj.showTransformHandles(bShow)
```

參數
bShow Boolean 值；會判斷顯示或隱藏目前工具的自由變形控制點 (true 會顯示控制點；false 會隱藏控制點)。

傳回值
無。

說明
方法；在一個可擴充工具 JavaScript 檔的 configureTool() 方法中進行呼叫，以指出工具作用中時，應該要出現的自由變形控制點。此方法僅使用在建立可擴充的工具時。

範例
請參閱 configureTool()。

第 47 章 Tools 物件

適用版本
Flash MX 2004。

說明
Tools 物件可由 Flash 物件存取 (fl.tools)。tools.toolObjs 屬性包含 ToolObj 物件的陣列，而且 tools.activeTool 屬性會傳回目前作用中工具的 ToolObj 物件 (請參閱 ToolObj 物件以及第 13 頁 「最上層函數和方法」中的可擴充工具清單)。

備註：下列方法和屬性只能在建立可擴充工具時使用。

方法摘要
Tools 物件可使用的方法如下：

方法	說明
tools.constrainPoint()	需要兩個點，會傳回一個新的調整點或 「限制 (constrained)」點。
tools.getKeyDown()	傳回最後按下的按鍵。
第 472 頁 「tools.setCreatingBbox()」	PLACEHOLDER
tools.setCursor()	將指標設定成指定的外觀。
tools.snapPoint()	需要一個點做為輸入，會傳回一個可能調整過或 「貼齊」最近幾何物件的新點。

屬性摘要
Tools 物件可使用的屬性如下：

屬性	說明
tools.activeTool	唯讀；傳回目前作用中工具的 ToolObj 物件。
tools.altIsDown	唯讀；Boolean 值；會識別是否按下 Alt 鍵。
tools.ctlIsDown	唯讀；Boolean 值；會識別是否按下 Control 鍵。
tools.mouseIsDown	唯讀；Boolean 值；會識別目前是否按下滑鼠左鍵。
tools.penDownLoc	唯讀；點；代表 「舞台」上最近一次按下滑鼠事件的位置。
tools.penLoc	唯讀；點；代表滑鼠的目前位置。
tools.shiftIsDown	唯讀；Boolean 值；會識別是否按下 Shift 鍵。
tools.toolObjs	唯讀；ToolObj 物件的陣列。

tools.activeTool

適用版本
Flash MX 2004。

用法
```
tools.activeTool
```

說明
唯讀屬性：傳回目前作用中工具的 ToolObj 物件。

範例
下列範例會將代表目前作用中工具的物件儲存於 theTool 變數中：
```
var theTool = fl.tools.activeTool;
```

tools.altIsDown

適用版本
Flash MX 2004。

用法
```
tools.altIsDown
```

說明
唯讀屬性：Boolean 值；會識別是否按下 Alt 鍵。如果已按下 Alt 鍵，這個值會是 true，否則會是 false。

範例
下列範例會判斷是否按下 Alt 鍵：
```
var isAltDown = fl.tools.altIsDown;
```

tools.constrainPoint()

適用版本
Flash MX 2004。

用法
```
tools.constrainPoint(pt1, pt2)
```

參數
pt1, pt2 點，指定按下起始點和拖曳結束點。

傳回值
新的調整點或限制點。

說明
方法；需要兩個點，會傳回一個新的調整點或限制點。如果在執行命令時按下 Shift 鍵，則限制傳回點遵從 45 度限制 (對帶有箭頭符號的線條特別有用)，或限制物件維持原來的比例 (例如使用 「矩形」工具拉出一個完美的正方形)。

範例
下列範例會傳回 「限制」點：

```
pt2 = fl.tools.constrainPoint(pt1, tempPt);
```

tools.ctlIsDown

適用版本
Flash MX 2004。

用法
```
tools.ctlIsDown
```

說明
唯讀屬性；Boolean 值；如果已按下 Control 鍵，會是 true；否則為 false。

範例
下列範例會判斷是否按下 Control 鍵：

```
var isCtrldown = fl.tools.ctrlIsDown;
```

tools.getKeyDown()

適用版本
Flash MX 2004。

用法
```
tools.getKeyDown()
```

參數
無。

傳回值
按鍵的整數值。

說明
方法；傳回最後按下的按鍵。

範例
下列範例會顯示最近按下之按鍵的整數值：

```
var theKey = fl.tools.getKeyDown();
fl.trace(theKey);
```

tools.mouseIsDown

適用版本
Flash MX 2004。

用法
```
tools.mouseIsDown
```

說明
唯讀屬性；Boolean 值；如果目前已按下滑鼠左鍵，會是 true；否則為 false。

範例
下列範例會判斷滑鼠是否已按下滑鼠左鍵。

```
var isMouseDown = fl.tools.mouseIsDown;
```

tools.penDownLoc

適用版本
Flash MX 2004。

用法
```
tools.penDownLoc
```

說明
唯讀屬性；點；代表「舞台」上最近一次按下滑鼠事件的位置。tools.penDownLoc 屬性包括 x 和 y 這兩個屬性，對應於滑鼠指標的 x,y 位置。

範例
下列範例會判斷 「舞台」上最近一次按下滑鼠事件的位置，並且在 「輸出」面板中顯示 x 和 y 值：

```
var pt1 = fl.tools.penDownLoc;
fl.trace("x,y location of last mouseDown event was " + pt1.x + ", " + pt1.y)
```

請參閱
tools.penLoc

tools.penLoc

適用版本
Flash MX 2004。

用法
```
tools.penLoc
```

說明
唯讀屬性；點；代表滑鼠指標的目前位置。tools.penLoc 屬性包括 x 和 y 這兩個屬性，對應於滑鼠指標的 x,y 位置。

範例
下列範例會判斷目前滑鼠的位置：

```
var tempPt = fl.tools.penLoc;
```

請參閱
tools.penDownLoc

tools.setCreatingBbox()

適用版本
Flash 11。

用法
```
tools.setCreatingBbox()
```

參數
Placeholder　整數；定義指標的外觀，如下列清單的說明：

傳回值
無。

說明
方法；將指標設定成指定的外觀。

範例
下列範例會將指標設定為黑箭頭。

```
fl.tools.setCursor(1);
```

tools.setCursor()

適用版本
Flash MX 2004。

用法
```
tools.setCursor(cursor)
```

參數
cursor　整數；定義指標的外觀，如下列清單的說明：

- 0 = 加號游標 (+)
- 1 = 黑箭頭
- 2 = 白箭頭
- 3 = 四向箭頭
- 4 = 雙向水平箭頭
- 5 = 雙向垂直箭頭
- 6 = X
- 7 = 手掌游標

傳回值
無。

說明
方法；將指標設定成指定的外觀。

範例
下列範例會將指標設定為黑箭頭。

```
fl.tools.setCursor(1);
```

tools.shiftIsDown

適用版本
Flash MX 2004。

用法
```
tools.shiftIsDown
```

說明
唯讀屬性；Boolean 值；如果已按下 Shift 鍵，會是 true；否則為 false。

範例
下列範例會判斷是否按下 Shift 鍵。

```
var isShiftDown = fl.tools.shiftIsDown;
```

tools.snapPoint()

適用版本
Flash MX 2004。

用法
```
tools.snapPoint(pt)
```

參數
pt 指定要傳回貼齊點的點位置。

傳回值
可能調整過或「貼齊」最近幾何物件的新點。

說明
方法；需要一個點做為輸入，會傳回一個可能調整過或「貼齊」最近幾何物件的新點。如果已在 Flash 使用者介面的「檢視」選單中停用貼齊，則傳回點會是原點。

範例
下列範例會傳回貼齊最近幾何物件的新點。

```
var theSnapPoint = fl.tools.snapPoint(pt1);
```

tools.toolObjs

適用版本
Flash MX 2004。

用法
```
tools.toolObjs
```

說明
唯讀屬性：ToolObj 物件的陣列 (請參閱 ToolObj 物件)。

第 48 章 Vertex 物件

適用版本
Flash MX 2004。

說明
Vertex 物件為保留座標資料的部分形狀資料結構。

方法摘要
Vertex 物件可以搭配下列方法使用：

方法	說明
vertex.getHalfEdge()	取得共享這個頂點的 HalfEdge 物件。
vertex.setLocation()	設定這個頂點的位置。

屬性摘要
Vertex 物件可使用的屬性如下：

屬性	說明
vertex.x	唯讀；頂點的 x 位置，以像素為單位。
vertex.y	唯讀；頂點的 y 位置，以像素為單位。

vertex.getHalfEdge()

適用版本
Flash MX 2004。

用法
```
vertex.getHalfEdge()
```

參數
無。

傳回值
HalfEdge 物件。

說明
方法；取得共享這個頂點的 HalfEdge 物件。

範例
下列範例會顯示如何取得共享相同頂點的其它不完整邊緣：

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var theVertex = hEdge.getVertex();
var someHEdge = theVertex.getHalfEdge(); // Not necessarily the same half edge
var theSameVertex = someHEdge.getVertex();
fl.trace('the same vertex: ' + theSameVertex);
```

vertex.setLocation()

適用版本
Flash MX 2004。

用法
```
vertex.setLocation(x, y)
```

參數
x 浮點值：指定要放置頂點的 x 座標，以像素為單位。

y 浮點值：指定要放置頂點的 y 座標，以像素為單位。

傳回值
無。

說明
方法；設定頂點的位置。使用這個方法之前，您必須呼叫 shape.beginEdit()。

範例
下列範例會將頂點設定至原點：

```
var shape = fl.getDocumentDOM().selection[0];
shape.beginEdit();
var hEdge = shape.edges[0].getHalfEdge(0);
var vertex = hEdge.getVertex();
var someHEdge = vertex.getHalfEdge();
var vertex = someHEdge.getVertex();
// Move the vertex to the origin.
vertex.setLocation(0.0, 0.0);
shape.endEdit();
```

vertex.x

適用版本
Flash MX 2004。

用法
```
vertex.x
```

說明
唯讀屬性；頂點的 x 位置，以像素為單位。

範例

下列範例會在「輸出」面板中顯示頂點 x 和 y 值的位置：

```
var shape = fl.getDocumentDOM().selection[0];
var hEdge = shape.edges[0].getHalfEdge(0);
var vertex = hEdge.getVertex();

fl.trace('x location of vertex is: ' + vertex.x);
fl.trace('y location of vertex is: ' + vertex.y);
```

vertex.y

適用版本

Flash MX 2004。

用法

```
vertex.y
```

說明

唯讀屬性；頂點的 y 位置，以像素為單位。

範例

請參閱：vertex.x。

第 49 章 VideoItem 物件

繼承　Item 物件 > VideoItem 物件

適用版本
Flash MX 2004。

說明
VideoItem 物件為 Item 物件的子類別。

方法摘要
除了 Item 物件方法，VideoItem 物件還有下列方法：

屬性	說明
videoItem.exportToFLV()	將指定的項目匯出至 FLV 檔。

屬性摘要
除了 Item 物件的屬性之外，VideoItem 物件還可以搭配下列屬性使用：

屬性	說明
videoItem.fileLastModifiedDate	唯讀；包含十六進位數字的字串，代表在 1970 年 1 月 1 日和原始檔案 (在磁碟上) 匯入至元件庫時的修改日期之間經過的秒數。
videoItem.sourceFileExists	唯讀；Boolean 值，指定匯入至元件庫中的檔案是否仍然存在於匯入來源位置。
videoItem.sourceFileIsCurrent	唯讀；Boolean 值，指定元件庫項目的檔案修改日期是否與已匯入的原始檔案 (在磁碟上) 的修改日期相同。
videoItem.sourceFilePath	唯讀；指定視訊項目路徑的字串。
videoItem.videoType	唯讀；字串，指定項目代表的視訊類型。

videoItem.exportToFLV()

適用版本
Flash CS4 Professional。

用法
```
videoItem.exportToFLV(fileURI)
```

參數
fileURI　字串；指定匯出檔案的路徑和名稱，表示為 file:/// URI。

傳回值
如果檔案成功匯出，就會傳回 Boolean 值 true，否則便傳回 false。

說明
方法：將指定的項目匯出至 FLV 檔。

範例
假設元件庫中的第一個項目是視訊項目，下列程式碼會將它匯出為 FLV 檔：

```
var videoFileURL = "file:///C|/out.flv";
var libItem = fl.getDocumentDOM().library.items[0];
libItem.exportToFLV(videoFileURL);
```

videoItem.fileLastModifiedDate

適用版本
Flash CS4 Professional。

用法
```
videoItem.fileLastModifiedDate
```

說明
唯讀屬性：包含十六進位數字的字串，代表在 1970 年 1 月 1 日和匯入至元件庫之原始檔案修改日期之間經過的秒數。如果檔案已不存在，此值為「00000000」。

範例
假設元件庫中的第一個項目是視訊項目，下列程式碼會將它顯示為十六進位數字 (如上述)。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("Mod date when imported = " + libItem.fileLastModifiedDate);
```

請參閱
videoItem.sourceFileExists、videoItem.sourceFileIsCurrent、videoItem.sourceFilePath、FLfile.getModificationDate()

videoItem.sourceFileExists

適用版本
Flash CS4 Professional。

用法
```
videoItem.sourceFileExists
```

說明
唯讀屬性：如果匯入至元件庫中的檔案仍然位於匯入來源位置，會傳回 Boolean 值 true，否則便傳回 false。

範例
假設元件庫中的第一個項目是視訊項目，下列程式碼會在匯入至元件庫中的檔案仍然存在時顯示 "true"。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("sourceFileExists = "+ libItem.sourceFileExists);
```

請參閱
videoItem.sourceFileIsCurrent、videoItem.sourceFilePath

videoItem.sourceFileIsCurrent

適用版本
Flash CS4 Professional。

用法
```
videoItem.sourceFileIsCurrent
```

說明
唯讀屬性：如果元件庫項目的檔案修改日期和 (磁碟上) 已匯入的原始檔案的修改日期相同，會傳回 Boolean 值 true，否則便傳回 false。

範例
假設元件庫中的第一個項目是視訊項目，下列程式碼會在匯入的原始檔案自匯入後未曾在磁碟上修改時顯示 "true"。

```
var libItem = fl.getDocumentDOM().library.items[0];
fl.trace("fileIsCurrent = "+ libItem.sourceFileIsCurrent);
```

請參閱
videoItem.fileLastModifiedDate、videoItem.sourceFilePath

videoItem.sourceFilePath

適用版本
Flash 8。

用法
```
videoItem.sourceFilePath
```

說明
唯讀屬性：字串 (表示為 file:/// URI)，指定視訊項目的路徑。

範例
下列範例會顯示元件庫中，其類型為 video 之任何項目的名稱和來源檔案路徑：

```
for (idx in fl.getDocumentDOM().library.items) {
if (fl.getDocumentDOM().library.items[idx].itemType == "video") {
    var myItem = fl.getDocumentDOM().library.items[idx];
    fl.trace(myItem.name + " source is " + myItem.sourceFilePath);
    }
}
```

請參閱
videoItem.sourceFileExists

videoItem.videoType

適用版本

Flash 8。

用法

```
videoItem.videoType
```

說明

唯讀屬性；字串；指定項目代表的視訊類型。可能的值為 "embedded video"、"linked video" 和 "video"。

範例

下列範例會顯示元件庫中，其類型為 video 之任何項目的名稱和類型：

```
for (idx in fl.getDocumentDOM().library.items) {
if (fl.getDocumentDOM().library.items[idx].itemType == "video") {
var myItem = fl.getDocumentDOM().library.items[idx];
fl.trace(myItem.name + " is " + myItem.videoType);
}
}
```

第 50 章 XMLUI 物件

適用版本
Flash MX 2004。

說明
Flash 8 支援以「XML 使用者介面語言」(XUL) 子集所撰寫的自訂對話方塊。「XML 使用者介面」(XMLUI) 對話方塊可供數個 Flash 功能 (例如命令和行為指令) 使用，針對您利用擴充功能建立的功能提供使用者介面。XMLUI 物件可用來取得和設定 XMLUI 對話方塊的屬性，以及接受或取消特定屬性。XMLUI 方法可用於回呼，例如按鈕中的 oncommand 處理常式。

您可以使用 document.xmlPanel() 方法撰寫 dialog.xml 檔，並從 JavaScript API 呼叫這個檔案。若要擷取代表目前 XMLUI 對話方塊的物件，請使用 fl.xmlui。

方法摘要
XMLUI 物件可使用的方法如下：

方法	說明
xmlui.accept()	使用接受狀態來關閉目前的 XMLUI 對話方塊。
xmlui.cancel()	使用取消狀態來關閉目前的 XMLUI 對話方塊。
xmlui.get()	擷取目前 XMLUI 對話方塊的指定屬性值。
xmlui.getControlItemElement()	傳回指定控制項目前的控制項目。
xmlui.getEnabled()	傳回 Boolean 值，指定啟用或停用 (灰色) 控制項。
xmlui.getVisible()	傳回 Boolean 值，指定控制項是可見或隱藏。
xmlui.set()	修改目前 XMLUI 對話方塊的指定屬性值。
xmlui.setControlItemElement()	設定目前項目的標籤和值。
xmlui.setControlItemElements()	設定目前項目的標籤、值配對。
xmlui.setEnabled()	啟用或停用 (灰白) 控制項。
xmlui.setVisible()	顯示或隱藏控制項。

xmlui.accept()

適用版本
Flash MX 2004。

用法
```
xmlui.accept()
```

參數
無。

傳回值
無。

說明
方法；使用接受狀態來關閉目前的 XMLUI 對話方塊，等於使用者按下「確定」按鈕。

請參閱
fl.xmlui、document.xmlPanel()、xmlui.cancel()

xmlui.cancel()

適用版本
Flash MX 2004。

用法
```
xmlui.cancel()
```

參數
無。

傳回值
無。

說明
方法；使用取消狀態來關閉目前的 XMLUI 對話方塊，等於使用者按下「取消」按鈕。

請參閱
fl.xmlui、document.xmlPanel()、xmlui.accept()

xmlui.get()

適用版本
Flash MX 2004。

用法
```
xmlui.get(controlPropertyName)
```

參數
controlPropertyName 字串；指定您要擷取其值的 XMLUI 屬性名稱。

傳回值
代表指定屬性的值的字串。對於可能獲得 true 或 false Boolean 值的狀況，會傳回 "true" 或 "false" 字串。

說明
方法；擷取目前 XMLUI 對話方塊的指定屬性值。

範例
下列範例會傳回名為 URL 屬性的值：

```
fl.xmlui.get("URL");
```

請參閱
fl.xmlui、document.xmlPanel()、xmlui.getControlItemElement()、xmlui.set()

xmlui.getControlItemElement()

適用版本
Flash 8。

用法

```
xmlui.getControlItemElement(controlPropertyName)
```

參數
controlPropertyName 字串；指定您要擷取其控制項目元素的屬性。

傳回值
物件；代表控制項 (使用 controlPropertyName 指定) 目前的控制項目。

說明
方法；傳回 ListBox 或 ComboBox 控制項 (使用 controlPropertyName 指定的) 中選取行的標籤和值。

範例
下列範例會傳回 myListBox 控制項目前選取行的標籤和值：

```
var elem = new Object();
elem = fl.xmlui.getControlItemElement("myListBox");
fl.trace("label = " + elem.label + " value = " + elem.value);
```

請參閱
fl.xmlui、document.xmlPanel()、xmlui.get()、xmlui.setControlItemElement()、xmlui.setControlItemElements()

xmlui.getEnabled()

適用版本
Flash 8。

用法

```
xmlui.getEnabled(controlID)
```

參數
controlID 字串；指定您要擷取其狀態的控制項之 ID 特質。

傳回值
如果控制項是啟用的話，會傳回 Boolean 值 true；否則為 false。

說明
方法；傳回 Boolean 值，指定啟用或停用 (灰色) 控制項。

範例
下列範例會傳回值，指出具備 ID 特質 myListBox 的控制項是否啟用：

```
var isEnabled = fl.xmlui.getEnabled("myListBox");
fl.trace(isEnabled);
```

請參閱
fl.xmlui、document.xmlPanel()、xmlui.setEnabled()

xmlui.getVisible()

適用版本
Flash 8。

用法
```
xmlui.getVisible(controlID)
```

參數
controlID 字串；指定您要擷取其可見性狀態的控制項之 ID 特質。

傳回值
如果控制項為可見，則 Boolean 值為 true；如果為不可見 (隱藏)，則為 false。

說明
方法；傳回 Boolean 值，指定控制項是可見或隱藏。

範例
下列範例會傳回值，指出具備 ID 特質 myListBox 的控制項是否可見：

```
var isVisible = fl.xmlui.getVisible("myListBox");
fl.trace(isVisible);
```

請參閱
xmlui.setVisible()

xmlui.set()

適用版本
Flash MX 2004。

用法

```
xmlui.set(controlPropertyName, value)
```

參數

controlPropertyName 字串；指定要修改的 XMLUI 屬性名稱。

value 字串；指定您要用來設定 XMLUI 屬性的值。

傳回值

無。

說明

方法；修改目前 XMLUI 對話方塊的指定屬性值。

範例

下列範例會將名為 URL 屬性的值設定為 www.adobe.com：

```
fl.xmlui.set("URL", "www.adobe.com");
```

請參閱

fl.xmlui、document.xmlPanel()、xmlui.get()、xmlui.setControlItemElement()、xmlui.setControlItemElements()

xmlui.setControlItemElement()

適用版本

Flash 8。

用法

```
xmlui.setControlItemElement(controlPropertyName, elementItem)
```

參數

controlPropertyName 字串；指定要設定的控制項目元素。

elementItem JavaScript 物件；帶有名為 label 的字串屬性，以及名為 value 的選擇性字串屬性。如果 value 屬性不存在，則會建立這個屬性，並指定與 label 相同的值。

傳回值

無。

說明

方法；設定 ListBox 或 ComboBox 控制項 (由 controlPropertyName 指定) 中目前選取的行的標籤和值。

範例

下列範例會設定控制屬性 (名稱為 PhoneNumber) 目前項目的標籤和值：

```
var elem = new Object();
elem.label = "Fax";
elem.value = "707-555-5555";
fl.xmlui.setControlItemElement("PhoneNumber",elem);
```

請參閱
fl.xmlui、document.xmlPanel()、xmlui.getControlItemElement()、xmlui.set()、xmlui.setControlItemElements()

xmlui.setControlItemElements()

適用版本
Flash 8。

用法
```
xmlui.setControlItemElements(controlID, elementItemArray)
```

參數
controlID 字串；指定您要設定的控制項之 ID 特質。

elementItemArray JavaScript 物件的陣列；其中每一個物件都帶有名為 label 的字串屬性，以及名為 value 的選擇性字串屬性。如果 value 屬性不存在，則會建立這個屬性，並指定與 label 相同的值。

傳回值
無。

說明
方法；清除 controlID 指定的 ListBox 或 ComboBox 控制項的值，並且以 elementItemArray 指定的 label、value 配對取代清單或選單項目。

範例
下列範例會在具有 ID 特質 myControlID 的控制項中，將項目的標籤和值設定成指定的 label、value 配對。

```
var nameArray = new Array("January", "February", "March");
var monthArray = new Array();
for (i=0;i<nameArray.length;i++){
    elem = new Object();
    elem.label = nameArray[i];
    elem.value = i;
    monthArray[i] = elem;
}
fl.xmlui.setControlItemElements("myControlID", monthArray);
```

請參閱
xmlui.getControlItemElement()、xmlui.set()、xmlui.setControlItemElement()

xmlui.setEnabled()

適用版本
Flash 8。

用法
```
xmlui.setEnabled(controlID, enable)
```

參數
controlID 字串；指定您要啟用或停用的控制項之 ID 特質。

enable 如果要啟用控制項，則 Boolean 值為 true；如果要停用 (灰白)，則為 false。

傳回值
無。

說明
方法；啟用或停用 (灰白) 控制項。

範例
下列範例會將具有 myControl ID 特質的控制項變灰色：

```
fl.xmlui.setEnabled("myControl", false);
```

請參閱
xmlui.getEnabled()

xmlui.setVisible()

適用版本
Flash 8。

用法

```
xmlui.setVisible(controlID, visible)
```

參數
controlID 字串；指定您要顯示或隱藏的控制項之 ID 特質。

visible 如果要顯示控制項，則 Boolean 值為 true；如果要隱藏，則為 false。

傳回值
無。

說明
方法；顯示或隱藏控制項。

範例
下列範例會隱藏具有 myControl ID 特質的控制項：

```
fl.xmlui.setVisible("myControl", false);
```

請參閱
xmlui.getVisible()

第 51 章 C 語言層次擴充

本章說明 C 語言層次擴充機制，它可以讓您結合 JavaScript 和自訂 C 語言程式碼來實作 Adobe Flash CS4 Professional 擴充檔案。這個版本的 Flash 並未導入此機制的任何變更。

關於擴充功能

若要實作擴充功能，您必須使用 C 定義函數、將函數合併至「動態連結程式庫」(DLL) 或共享元件庫、將元件庫儲存在適當的目錄中，然後使用 Adobe Flash JavaScript API 從 JavaScript 呼叫函數。

例如，您可能要定義密集運算效率較 JavaScript 更佳的函數，以改善效能或者建立進階工具或特效。

這個擴充機制是 Adobe Dreamweaver CS3 API 的子集。如果您對該 API 很熟悉，可能會在 C 語言層次擴充機制 API 中辨識這些函數。不過，此 API 與 Dreamweaver API 有以下差異：

- 此 API 不包含 Dreamweaver API 中的所有命令。
- Dreamweaver API 中 wchar_t 和 char 類型的所有宣告，在此 API 中都實作為 unsigned short 宣告，以便在傳遞字串時支援 Unicode。
- 此 API 中的 JSVal JS_BytesToValue() 函數不屬於 Dreamweaver API 的一部分。
- 必須將 DLL 或共享元件庫檔案儲存於不同位置 (請參閱第 489 頁「整合 C 函數」)。

整合 C 函數

您可以使用 C 語言層次擴充機制結合 JavaScript 和 C 程式碼來實作 Flash 擴充檔案。下列步驟摘要說明實作此功能的程序：

1. 使用 C 或 C++ 語言定義函數。
2. 將函數集結至 DLL 檔案 (Windows) 或共享元件庫 (Macintosh)。
3. 將 DLL 檔案或元件庫儲存在適當的位置：
   - Windows 7：

     < 開機磁碟 >\Users\< 使用者名稱 >\AppData\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\External Libraries
   - Windows Vista：

     < 開機磁碟 >\Users\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\External Libraries
   - Windows XP：

     < 開機磁碟 >\Documents and Settings\< 使用者名稱 >\Local Settings\Application Data\Adobe\Flash CS5 或 CS5.5\< 語言 >\Configuration\External Libraries
   - Mac OS X：

     Macintosh HD/Users/< 使用者名稱 >/Library/Application Support/Adobe/Flash CS5 或 CS5.5/< 語言 >/Configuration/External Libraries
4. 建立用來呼叫函數的 JSFL 檔。

5 在 Flash 編寫環境中從「命令」選單執行 JSFL 檔。

如需詳細資訊，請參閱：第 493 頁「DLL 樣本實作」。

C 語言層次擴充機制和 JavaScript 解譯器

DLL 或共享元件庫中的 C 程式碼，分別於三個不同時段與 Flash JavaScript API 互動：

- 啟動時，註冊元件庫的函數
- 呼叫 C 函數時，解壓縮從 JavaScript 傳遞至 C 的引數
- 傳回 C 函數之前，封裝傳回值

為了完成這些工作，解譯器會定義幾種資料類型並公開 (Expose) API。本節列出的資料類型和函數定義，將顯示於 mm_jsapi.h 檔案中。為了讓元件庫正確地運作，您必須使用下列文字，在元件庫的每個檔案上加入 mm_jsapi.h 檔案：

```
#include "mm_jsapi.h"
```

加入 mm_jsapi.h 檔案，將一併加入定義 MM_Environment 結構的 mm_jsapi_environment.h 檔案。

若要取得 mm_jsapi.h 檔案的副本，請從樣本 ZIP 或 SIT 檔中擷取 (請參閱：第 493 頁「DLL 樣本實作」)，或將下列程式碼複製到您命名為 mm_jsapi.h 的檔案中：

```
#ifndef _MM_JSAPI_H_
#define _MM_JSAPI_H_

/************************************************************************
    * Public data types
    ************************************************************************/

typedef struct JSContext JSContext;
typedef struct JSObject JSObject;
typedef long jsval;
#ifndef JSBool
typedef long JSBool;
#endif

typedef JSBool (*JSNative)(JSContext *cx, JSObject *obj, unsigned int argc,
jsval *argv, jsval *rval);

/* Possible values for JSBool */
#define JS_TRUE 1
#define JS_FALSE 0



/************************************************************************
    * Public functions
    ************************************************************************/

/* JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs) */
#define JS_DefineFunction(n, c, a) \
(mmEnv.defineFunction ? (*(mmEnv.defineFunction))(mmEnv.libObj, n, c, a) \
: JS_FALSE)

/* unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength) */
#define JS_ValueToString(c, v, l) \
(mmEnv.valueToString? (*(mmEnv.valueToString))(c, v, l) : (char *)0)

/* unsigned char *JS_ValueToBytes(JSContext *cx, jsval v, unsigned int *pLength) */
#define JS_ValueToBytes(c, v, l) \
(mmEnv.valueToBytes? (*(mmEnv.valueToBytes))(c, v, l) : (unsigned char *)0)
```

```
/* JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp); */
#define JS_ValueToInteger(c, v, l) \
(mmEnv.valueToInteger ? (*(mmEnv.valueToInteger))(c, v, l) : JS_FALSE)

/* JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp); */
#define JS_ValueToDouble(c, v, d) \
(mmEnv.valueToDouble? (*(mmEnv.valueToDouble))(c, v, d) : JS_FALSE)

/* JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp); */
#define JS_ValueToBoolean(c, v, b) \
(mmEnv.valueToBoolean ? (*(mmEnv.valueToBoolean))(c, v, b) : JS_FALSE)

/* JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op); */
#define JS_ValueToObject(c, v, o) \
(mmEnv.valueToObject? (*(mmEnv.valueToObject))(c, v, o) : JS_FALSE)

/* JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp); */
#define JS_StringToValue(c, b, s, v) \
(mmEnv.stringToValue? (*(mmEnv.stringToValue))(c, b, s, v) : JS_FALSE)

/* JSBool JS_BytesToValue(JSContext *cx, unsigned char *bytes, uint sz, jsval *vp); */
#define JS_BytesToValue(c, b, s, v) \
(mmEnv.bytesToValue? (*(mmEnv.bytesToValue))(c, b, s, v) : JS_FALSE)

/* JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp); */
#define JS_DoubleToValue(c, d, v) \
(mmEnv.doubleToValue? (*(mmEnv.doubleToValue))(c, d, v) : JS_FALSE)

/* jsval JS_IntegerToValue(long lv); */
#define JS_IntegerToValue(lv) (((jsval)(lv) << 1) | 0x1)

/* jsval JS_BooleanToValue(JSBool bv); */
#define JS_BooleanToValue(bv) (((jsval)(bv) << 3) | 0x6)

/* jsval JS_ObjectToValue(JSObject *obj); */
#define JS_ObjectToValue(ov)((jsval)(ov))

/* unsigned short *JS_ObjectType(JSObject *obj); */
#define JS_ObjectType(o) \
(mmEnv.objectType ? (*(mmEnv.objectType))(o) : (char *)0)

/* JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length, jsval *v) */
#define JS_NewArrayObject(c, l, v) \
(mmEnv.newArrayObject ? (*(mmEnv.newArrayObject))(c, l, v) : (JSObject *)0)

/* long JS_GetArrayLength(JSContext *cx, JSObject *obj) */
#define JS_GetArrayLength(c, o) \
(mmEnv.getArrayLength ? (*(mmEnv.getArrayLength))(c, o) : -1)

/* JSBool JS_GetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp) */
#define JS_GetElement(c, o, i, v) \
(mmEnv.getElement ? (*(mmEnv.getElement))(c, o, i, v) : JS_FALSE)

/* JSBool JS_SetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp) */
#define JS_SetElement(c, o, i, v) \
(mmEnv.setElement ? (*(mmEnv.setElement))(c, o, i, v) : JS_FALSE)

/* JSBool JS_ExecuteScript(JSContext *cx, JSObject *obj, unsigned short *script,
    * unsigned int sz, jsval *rval) */
#define JS_ExecuteScript(c, o, s, z, r) \
(mmEnv.executeScript? (*(mmEnv.executeScript))(c, o, s, z, (LPCTSTR)__FILE__, \
```

```
__LINE__, r) : JS_FALSE)

/* JSBool JS_ReportError(JSContext *cx, unsigned short *error, unsigned int sz) */
#define JS_ReportError(c, e, s) \
(mmEnv.reportError? (*(mmEnv.reportError))(c, e, s) : JS_FALSE)



/************************************************************************
    * Private data types, macros, and globals
    ************************************************************************/

typedef struct {
JSObject *libObj;
JSBool (*defineFunction)(JSObject *libObj, unsigned short *name, JSNative call,
unsigned int nargs);
unsigned short *(*valueToString)(JSContext *cx, jsval v, unsigned int *pLength);
unsigned char *(*valueToBytes)(JSContext *cx, jsval v, unsigned int *pLength);
JSBool (*valueToInteger)(JSContext *cx, jsval v, long *lp);
JSBool (*valueToDouble)(JSContext *cx, jsval v, double *dp);
JSBool (*valueToBoolean)(JSContext *cx, jsval v, JSBool *bp);
JSBool (*valueToObject)(JSContext *cx, jsval v, JSObject **op);
JSBool (*stringToValue)(JSContext *cx, unsigned short *b, unsigned int sz, jsval *vp);
JSBool (*bytesToValue)(JSContext *cx, unsigned char *b, unsigned int sz, jsval *vp);
JSBool (*doubleToValue)(JSContext *cx, double dv, jsval *vp);
unsigned short *(*objectType)(JSObject *obj);
JSObject *(*newArrayObject)(JSContext *cx, unsigned int length, jsval *vp);
long (*getArrayLength)(JSContext *cx, JSObject *obj);
JSBool (*getElement)(JSContext *cx, JSObject *obj, unsigned int idx,
jsval *vp);
JSBool (*setElement)(JSContext *cx, JSObject *obj, unsigned int idx,
jsval *vp);
JSBool (*executeScript)(JSContext *cx, JSObject *obj, unsigned short *script,
unsigned int sz, unsigned short *file, unsigned int lineNum, jsval *rval);
JSBool (*reportError)(JSContext *cx, unsigned short *error, unsigned int sz);
} MM_Environment;

extern MM_Environment mmEnv;

// Declare the external entry point and linkage
#ifdef _WIN32
# ifndef _MAC
// Windows
__declspec( dllexport ) void MM_InitWrapper( MM_Environment *env, unsigned int envSize );
# endif
#else
extern void MM_InitWrapper( MM_Environment *env, unsigned int envSize );
#endif


#define MM_STATE\
/* Definitions of global variables */ \
MM_Environment mmEnv; \
\
void\
MM_InitWrapper(MM_Environment *env, unsigned int envSize) \
{ \
```

```
extern void MM_Init();\
\
char **envPtr = (char **)env; \
char **mmPtr =(char **)(&mmEnv);\
char **envEnd = (char **)((char *)envPtr + envSize);\
char **mmEnd =(char **)((char *)mmPtr+ sizeof(MM_Environment)); \
\
/* Copy fields from env to mmEnv, one pointer at a time */\
while (mmPtr < mmEnd && envPtr < envEnd)\
*mmPtr++ = *envPtr++; \
\
/* If env doesn't define all of mmEnv's fields, set extras to NULL */ \
while (mmPtr < mmEnd) \
*mmPtr++ = (char *)0; \
\
/* Call user's MM_Init function */\
MM_Init();\
} \

#endif /* _MM_JSAPI_H_ */
```

DLL 樣本實作

本節說明如何建立簡單的 DLL 實作。如果您想要瞭解這個程序的運作方式，但不要自己實際建立 DLL，可以安裝 Samples.zip 檔案中所提供的 DLL 樣本檔；這些檔案位於 ExtendingFlash/dllSampleComputeSum 資料夾中 (如需下載 Samples.zip 檔案的詳細資訊，請參閱第 11 頁 「樣本實作」)。從 dllSampleComputeSum.dmg 或 dllSampleComputeSum.zip 檔案中擷取樣本檔，然後執行下列步驟：

- 將 Sample.jsfl 檔儲存在 Configuration/Commands 目錄中 (請參閱第 2 頁 「儲存 JSFL 檔」)。
- 將 Sample.dll 檔儲存在 Configuration/External Libraries 目錄中 (請參閱第 489 頁 「整合 C 函數」)。
- 在 Flash 編寫環境中，選取「命令 > 樣本」。JSFL 檔案中的 trace 陳述式會將 Sample.dll 中定義的函數結果傳回「輸出」面板。

本節其餘部分將討論樣本的開發。在此情況下，DLL 僅包含一個會增加兩個數字的函數。C 程式碼如下所示：

```
// Source code in C
// Save the DLL or shared library with the name "Sample".
#include <windows.h>
#include <stdlib.h>

#include "mm_jsapi.h"

// A sample function
// Every implementation of a JavaScript function must have this signature.
JSBool computeSum(JSContext *cx, JSObject *obj, unsigned int argc, jsval *argv, jsval *rval)
{
    long a, b, sum;

    // Make sure the right number of arguments were passed in.
    if (argc != 2)
        return JS_FALSE;

    // Convert the two arguments from jsvals to longs.
    if (JS_ValueToInteger(cx, argv[0], &a) == JS_FALSE ||
        JS_ValueToInteger(cx, argv[1], &b) == JS_FALSE)
            return JS_FALSE;

    /* Perform the actual work. */
    sum = a + b;

    /* Package the return value as a jsval. */
    *rval = JS_IntegerToValue(sum);

    /* Indicate success. */
    return JS_TRUE;
}
```

撰寫完此程式碼後，請建立 DLL 檔或共享元件庫，並將其儲存在適當的 Configuration/External Libraries 目錄中 (請參閱第 489 頁「整合 C 函數」)。然後以下列程式碼建立 JSFL 檔，並將其儲存在 Configuration/Commands 目錄中 (請參閱第 2 頁「儲存 JSFL 檔」)。

```
// JSFL file to run C function defined above.
var a = 5;
var b = 10;
var sum = Sample.computeSum(a, b);
fl.trace("The sum of " + a + " and " + b + " is " + sum );
```

若要執行 DLL 中定義的函數，請選取 Flash 編寫環境中的「命令 > 樣本」。

資料類型

JavaScript 解譯器會定義本節所述的資料類型。

typedef struct JSContext JSContext

不透明資料類型的指標會傳遞至 C 語言層次函數。API 中的某些函數接受此指標做為引數。

typedef struct JSObject JSObject

不透明資料類型的指標會傳遞至 C 語言層次函數。此資料類型代表某個物件，可能為陣列物件或其它物件類型。

typedef struct jsval jsval

為不透明資料結構，可含整數或者是 float、字串或物件的指標。API 中的部分函數可藉由讀取 jsval 結構的內容來讀取函數引數值，有些函數則可藉由撰寫 jsval 結構來撰寫函數的傳回值。

typedef enum { JS_FALSE = 0, JS_TRUE = 1 } JSBool

用來儲存 Boolean 值的簡單資料類型。

C 語言層次 API

C 語言層次擴充 API 包含 JSBool (*JSNative) 函數簽名及下列函數：

- JSBool JS_DefineFunction()
- unsigned short *JS_ValueToString()
- JSBool JS_ValueToInteger()
- JSBool JS_ValueToDouble()
- JSBool JS_ValueToBoolean()
- JSBool JS_ValueToObject()
- JSBool JS_StringToValue()
- JSBool JS_DoubleToValue()
- JSVal JS_BooleanToValue()
- JSVal JS_BytesToValue()
- JSVal JS_IntegerToValue()
- JSVal JS_ObjectToValue()
- unsigned short *JS_ObjectType()
- JSObject *JS_NewArrayObject()
- long JS_GetArrayLength()
- JSBool JS_GetElement()
- JSBool JS_SetElement()
- JSBool JS_ExecuteScript()

typedef JSBool (*JSNative)(JSContext *cx, JSObject *obj, unsigned int argc, jsval *argv, jsval *rval)

說明
方法：說明下列情況中 JavaScript 函數的 C 語言層次實作：

- cx 指標為不透明 JSContext 結構的指標，必須傳遞至 JavaScript API 中的某些函數。此變數會保留解譯器的執行內容。
- obj 指標為指令碼執行內容物件的指標。當指令碼正在執行時，關鍵字 this 等於此物件。
- argc 整數為傳遞至函數的引數數目。

- argv 指標為 jsval 結構陣列的指標。陣列長度為 argc 個元素。
- rval 指標為單一 jsval 結構的指標。函數的傳回值應寫入 *rval。

若成功，函數傳回 JS_TRUE；否則傳回 JS_FALSE。如果函數傳回 JS_FALSE，則目前指令碼會停止執行並顯示錯誤訊息。

JSBool JS_DefineFunction()

用法

```
JSBool JS_DefineFunction(unsigned short *name, JSNative call, unsigned int nargs)
```

說明

方法：使用 Flash 中的 JavaScript 解譯器註冊 C 語言層次函數。當 JS_DefineFunction() 函數註冊您在 call 引數中指定的 C 語言層次函數之後，您就可以使用在 name 引數中指定的名稱參考該 C 語言層次函數，並於 JavaScript 指令碼中叫用。name 引數必須區分大小寫。

此函數通常是從 Flash 在啟動時呼叫的 MM_Init() 函數呼叫。

引數

unsigned short *name、JSNativecall、unsigned intnargs

- name 引數為函數在 JavaScript 公開時的名稱。
- call 引數為 C 語言層次函數的指標。函數必須傳回 JSBool，指示成功或失敗。
- nargs 引數為函數預定要接收的引數數目。

傳回值

Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

unsigned short *JS_ValueToString()

用法

```
unsigned short *JS_ValueToString(JSContext *cx, jsval v, unsigned int *pLength)
```

說明

方法：從 jsval 結構擷取函數引數，在可能的情況下，將其轉換成字串，並將轉換值傳回至呼叫程式。

備註：請不要修改傳回的緩衝區指標，否則可能會損壞 JavaScript 解譯器的資料結構。若要變更字串，必須複製字元至其它的緩衝區並建立新的 JavaScript 字串。

引數

JSContext*cx、jsvalv、unsigned int*pLength

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- v 引數為 jsval 結構，也就是擷取字串的結構。
- pLength 引數為指向無正負號整數的指標。此函數會將 *plength 設定為字串長度 (以位元組為單位)。

傳回值

如果成功，則傳回指向以 null 結尾的字串的指標；如果失敗，則傳回指向 null 值的指標。呼叫程序在完成時不能釋放此字串。

JSBool JS_ValueToInteger()

用法

```
JSBool JS_ValueToInteger(JSContext *cx, jsval v, long *lp);
```

說明

方法：從 jsval 結構擷取函數引數，在可能的情況下，將其轉換成整數，並將轉換值傳回至呼叫程式。

引數

JSContext*cx、jsvalv、long*lp

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- v 引數為 jsval 結構，也就是擷取整數的結構。
- lp 引數為指向 4 位元組整數的指標。此函數會將轉換值儲存於 *lp。

傳回值

Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSBool JS_ValueToDouble()

用法

```
JSBool JS_ValueToDouble(JSContext *cx, jsval v, double *dp);
```

說明

方法：從 jsval 結構擷取函數引數，在可能的情況下，將其轉換成雙精度浮點數 (double)，並將轉換值傳回至呼叫程式。

引數

JSContext*cx、jsvalv、double*dp

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- v 引數為 jsval 結構，也就是擷取雙精度浮點數 (double) 的結構。
- dp 引數為指向 8 位元組雙精度浮點數 (double) 的指標。此函數會將轉換值儲存於 *dp。

傳回值

Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSBool JS_ValueToBoolean()

用法

```
JSBool JS_ValueToBoolean(JSContext *cx, jsval v, JSBool *bp);
```

說明

方法：從 jsval 結構擷取函數引數，在可能的情況下，將其轉換成 Boolean 值，並將轉換值傳回至呼叫程式。

引數
JSContext*cx、jsvalv、JSBool*bp

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- v 引數為 jsval 結構，也就是擷取 Boolean 值的結構。
- bp 引數為指向 JSBool Boolean 值的指標。此函數會將轉換值儲存於 *bp。

傳回值
Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSBool JS_ValueToObject()

用法
```
JSBool JS_ValueToObject(JSContext *cx, jsval v, JSObject **op);
```

說明
方法：從 jsval 結構擷取函數引數，在可能的情況下，將其轉換成物件，並將轉換值傳回至呼叫程式。如果物件為陣列，請使用 JS_GetArrayLength() 和 JS_GetElement() 來讀取其內容。

引數
JSContext*cx、jsvalv、JSObject**op

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- v 引數為 jsval 結構，也就是擷取物件的結構。
- op 引數為指向 JSObject 指標的指標。此函數會將轉換值儲存於 *op。

傳回值
Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSBool JS_StringToValue()

用法
```
JSBool JS_StringToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

說明
方法：將字串傳回值儲存於 jsval 結構中。它會配置新的 JavaScript 字串物件。

引數
JSContext*cx、unsigned short *bytes、size_tsz、jsval*vp

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- bytes 引數為將儲存於 jsval 結構的字串。字串資料已複製，以便呼叫程式在不使用字串時釋放字串。如果沒有指定字串大小 (請參閱 sz 引數)，則字串必須以 null 結尾。
- sz 引數為字串的大小，以位元組為單位。如果 sz 為 0，則會自動計算以 null 結尾的字串的長度。
- vp 引數為指向 jsval 結構的指標，也就是要複製字串內容的結構。

傳回值
Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSBool JS_DoubleToValue()

用法
```
JSBool JS_DoubleToValue(JSContext *cx, double dv, jsval *vp);
```

說明
方法；將浮點數傳回值儲存於 jsval 結構中。

引數
JSContext*cx、doubledv、jsval*vp

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- dv 引數為 8 位元組浮點數。
- vp 引數為指向 jsval 結構的指標，也就是要複製雙精度浮點數 (double) 內容的結構。

傳回值
Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSVal JS_BooleanToValue()

用法
```
jsval JS_BooleanToValue(JSBool bv);
```

說明
方法；將 Boolean 傳回值儲存在 jsval 結構中。

引數
JSBool bv

- bv 引數為 Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

傳回值
JSVal 結構，包含會當做引數傳遞至函數的 Boolean 值。

JSVal JS_BytesToValue()

用法
```
JSBool JS_BytesToValue(JSContext *cx, unsigned short *bytes, uint sz, jsval *vp);
```

說明
方法；轉換位元組為 JavaScript 值。

引數
JSContext *cx、unsignedshort*bytes、uintsz、jsval *vp

- cx 引數為 JavaScript 內容。
- bytes 引數為轉換成 JavaScript 物件的字串 (以位元組為單位)。
- sz 引數為轉換的位元組數目。
- vp 引數為 JavaScript 值。

傳回值
Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSVal JS_IntegerToValue()

用法
```
jsval JS_IntegerToValue(long lv);
```

說明
方法：將長整數值轉換成 JSVal 結構。

引數
lv

lv 引數為要轉換成 jsval 結構的長整數值。

傳回值
JSVal 結構，包含會當做引數傳遞至函數的整數。

JSVal JS_ObjectToValue()

用法
```
jsval JS_ObjectToValue(JSObject *obj);
```

說明
方法：將物件傳回值儲存於 JSVal。使用 JS_NewArrayObject() 來建立陣列物件；使用 JS_SetElement() 來定義其內容。

引數
JSObject *obj

obj 引數為指向 JSObject 物件的指標，也就是要轉換成 JSVal 結構的物件。

傳回值
JSVal 結構，包含會當做引數傳遞至函數的物件。

unsigned short *JS_ObjectType()

用法
```
unsigned short *JS_ObjectType(JSObject *obj);
```

說明
方法：指定物件參考，傳回物件的類別名稱。例如，如果物件為 DOM 物件，則函數傳回 Document。如果物件是文件中的節點，則函數傳回 Element。如果是陣列物件，則函數傳回 Array。

備註：請不要修改傳回的緩衝區指標，否則可能會損壞 JavaScript 解譯器的資料結構。

引數
JSObject *obj

此引數通常是使用 JS_ValueToObject() 函數傳回及轉換。

傳回值
指向以 null 結尾的字串的指標。呼叫程式在完成時不能釋放此字串。

JSObject *JS_NewArrayObject()

用法
```
JSObject *JS_NewArrayObject(JSContext *cx, unsigned int length [, jsval *v])
```

說明
方法：建立包含 JSVals 陣列的新物件。

引數
JSContext*cx、unsigned intlength、jsval*v

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- length 引數為陣列可以保留的元素數目。
- v 引數為選擇性指標，指向儲存在陣列中的 jsvals。如果傳回值不為 null，則 v 為包含 length 元素的陣列。如果傳回值為 null，則陣列物件的初始內容未定義，可以使用 JS_SetElement() 函數來定義。

傳回值
指向新陣列物件的指標；如果失敗，則為指向 null 值的指標。

long JS_GetArrayLength()

用法
```
long JS_GetArrayLength(JSContext *cx, JSObject *obj)
```

說明
方法：指定陣列物件指標，以取得陣列中的元素數目。

引數
JSContext*cx、JSObject*obj

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- obj 引數為指向陣列物件的指標。

傳回值
陣列中的元素數目；失敗時則傳回 -1。

JSBool JS_GetElement()

用法
```
JSBool JS_GetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

說明
方法：讀取陣列物件的單一元素。

引數
JSContext*cx、JSObject*obj、jsintidx、jsval*vp

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- obj 引數為指向陣列物件的指標。
- idx 引數為陣列的整數索引。第一個元素的索引為 0，最後一個元素的索引為 (length 1-)。
- vp 引數為指向 jsval 的指標，也就是要複製陣列中 jsval 結構內容的地方。

傳回值
Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSBool JS_SetElement()

用法
```
JSBool JS_SetElement(JSContext *cx, JSObject *obj, jsint idx, jsval *vp)
```

說明
方法：撰寫陣列物件的單一元素。

引數
JSContext*cx、JSObject*obj、jsintidx、jsval*vp

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- obj 引數為指向陣列物件的指標。
- idx 引數為陣列的整數索引。第一個元素的索引為 0，最後一個元素的索引為 (length 1-)。
- vp 引數為指向 jsval 結構的指標，其內容要複製到陣列中的 jsval。

傳回值
Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。

JSBool JS_ExecuteScript()

用法
```
JS_ExecuteScript (JSContext *cx, JSObject *obj, unsigned short *script, unsigned int sz, jsval *rval)
```

說明

方法：編譯和執行 JavaScript 字串。如果指令碼產生傳回值，則傳回於 *rval。

引數

JSContext *cx、JSObject *obj、unsigned short *script、unsigned intsz、jsval *rval

- cx 引數為傳遞至 JavaScript 函數的不透明 JSContext 指標。
- obj 引數為指令碼執行內容物件的指標。當指令碼正在執行時，關鍵字 this 等於此物件。通常為傳遞至 JavaScript 函數的 JSObject 指標。
- script 引數為包含 JavaScript 程式碼的字串。如果沒有指定字串大小 (請參閱 sz 引數)，則字串必須以 null 結尾。
- sz 引數為字串的大小，以位元組為單位。如果 sz 為 0，則會自動計算以 null 結尾的字串的長度。
- rval 引數為單一 jsval 結構的指標。函數的傳回值會儲存於 *rval。

傳回值

Boolean 值：JS_TRUE 表示成功；JS_FALSE 表示失敗。