

OLE 自動化指南

OLE 自動化指南

版權聲明

Copyright © 2008 Symantec Corporation. 版權 © 2008 賽門鐵克公司。All rights reserved. 版權所有。

Symantec 和 Symantec 標誌是 Symantec Corporation 或其子公司在美國及其他國家的商標或註冊商標。其他名稱可能為其個別所有者的商標。

本賽門鐵克產品，可能包含賽門鐵克必須支付授權費予第三方之第三方軟體(「第三方程式」)。部分第三方程式採取開放原始碼，或由免費軟體授權方式取得。本軟體隨附之授權許可協議，並未改變依開放原始碼或免費軟體授權所規定之任何權利或義務。請參閱本說明文件之「第三方版權聲明附錄」，或本賽門鐵克產品隨附之 ReadMe 檔，以取得第三方程式相關資訊。

本文件中所述產品的散佈受到授權許可協議的規範，限制其使用、複製、散佈及解譯/逆向工程。未事先獲得賽門鐵克公司及其授權者(如果有)的書面授權，本產品的任何部分均不得以任何方式、任何形式複製。

本文件完全依「現狀」提供，不做任何所有明示或暗示的條件、聲明及保證，其中包含在任何特定用途之適售性與適用性的暗示保證、任何特定用途或不侵害他人權益，除了此棄權聲明認定的不合法部分以外。對於所提供的效能或這份說明文件的使用等相關意外或必然損害，賽門鐵克公司不負任何責任。本說明文件所包含的資訊若有變更，恕不另行通知。

根據 FAR 第 12.212 節的定義，授權許可的軟體及文件為商業電腦產品，並僅限於 FAR 第 52.227-19 節「商業電腦軟體 - 有限權利」、DFARS 第 227.7202 節「商業電腦軟體或商業電腦軟體文件權利」及任何相關增修法規所規定的有限權利範圍。美國政府僅得根據本協議條款，進行授權軟體和文件的任何使用、修改、重製發行、運用、顯示或揭露。

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

<http://www.symantec.com>

技術支援

「賽門鐵克技術支援」維護全球的支援中心。「技術支援」的主要角色是回應有關產品特性與功能的特定查詢。「技術支援」小組亦負責編寫線上「知識庫」。「技術支援」小組會與賽門鐵克的其他部門協力合作，以期在極短的時間內回答您的問題。例如，「技術支援」小組會與「產品工程」以及「賽門鐵克安全機制應變中心」合作，提供警示服務及病毒定義檔更新。

賽門鐵克的維護工作，包含下列項目：

- 廣泛的支援選項，讓您在面對任何規模的組織時都能彈性選取正確的服務量
- 電話與網頁支援服務，提供快速回應與最新資訊
- 升級保證，可傳遞自動軟體升級防護
- 全年無休的全球支援
- 進階功能，包括了「專案管理服務」

如需關於賽門鐵克維護服務方案的詳細資訊，您可以造訪我們的網站，網址為：

www.symantec.com/techsupp/

聯絡技術支援

具有現行維護服務合約的客戶，可以透過下列網址存取「技術支援」資訊：

www.symantec.com/techsupp/

在聯絡「技術支援」前，請確認是否符合列示於產品文件中的系統要求。另外，您應位於發生問題的電腦前，以防需要複寫問題。

在聯絡「技術支援」時，請準備好以下資訊：

- 產品版次
- 硬體資訊
- 可用的記憶體、磁碟空間和 NIC 資訊
- 作業系統
- 版本和修正程式等級
- 網路拓樸
- 路由器、閘道和 IP 位址資訊
- 問題說明：
 - 錯誤訊息和日誌檔
 - 在聯絡賽門鐵克前所執行的疑難排解

- 最新的軟體架構變更及網路變更

授權與註冊

如果您的賽門鐵克產品需要註冊或授權碼，請存取我們的技術支援網頁，網址為：

www.symantec.com/techsupp/

客戶服務

您可以在下列網址取得客戶服務資訊：

www.symantec.com/techsupp/

您可以使用「客戶服務」來協助解決下列各種問題：

- 產品授權或序列化等相關問題
- 產品註冊更新，如地址或姓名變更
- 一般產品資訊 (功能、可用的語言、當地代理商)
- 產品更新及升級的最新資訊
- 升級保證與維護合約的相關資訊
- 與賽門鐵克採購計畫相關的資訊
- 對賽門鐵克技術支援選項的相關建議
- 非技術性的預售問題
- 與光碟或手冊相關的問題

維護合約資源

如果您要與賽門鐵克聯絡和現有維護合約相關的事項，請聯絡您所在地區的維護合約管理小組，聯絡資訊如下：

亞太地區及日本

contractsadmin@symantec.com

歐洲、中東地區及非洲

semea@symantec.com

北美洲及拉丁美洲

[supportolutions@symantec.com](mailto:supportsolutions@symantec.com)

其他企業服務

賽門鐵克提供了廣泛的服務，讓您能最佳利用對賽門鐵克產品的投資，以開發您的知識、專材，以及總體的洞察力，讓您能預先管理企業風險。

其他可用的企業服務為：

賽門鐵克早期預警解決方案	這些解決方案能提供網路攻擊的早期預警，廣泛的威脅分析，以及在發生攻擊前事先防範的對策。
安全管理服務	這些服務可以移除管理和監控安全裝置和事件的負擔，確保對真實威脅的快速回應。
諮詢顧問服務	「賽門鐵克諮詢顧問服務」提供由賽門鐵克以及信任夥伴派遣的現場技術專員。 「賽門鐵克諮詢顧問服務」提供更多元的套裝式以及客製式的選項，涵蓋評估、設計、實作、監控和管理等各方面，每項服務都專注於建立並維護您 IT 資源之整合性與可用性。
教育訓練服務	教育訓練服務提供了完整的技術訓練、安全教育、安全憑證，以及認知溝通方案。

若要存取更多關於「企業服務」的資訊，請造訪我們的網站，網址為：

www.symantec.com

從網站索引中選取您的國家/地區或語言。

目錄

技術支援	3
第 1 章	與 Symantec pcAnywhere 搭配使用 OLE 自動化 9
關於 OLE 自動化	9
關於 pcAnywhere 自動化伺服器	10
您可以利用 pcAnywhere 自動化伺服器做什麼	10
在開始之前	10
自動註冊主控端引擎	11
手動註冊主控端引擎	11
存取 pcAnywhere 自動化伺服器	11
使用 Visual Basic 存取 pcAnywhere 自動化伺服器	11
使用 Visual C++ 存取 pcAnywhere 自動化伺服器	12
執行被控端與主控端 OLE 物件	14
何處有更多的資訊	14
第 2 章	Visual Basic 物件定義 15
關於 Visual Basic 物件	15
CRemoteDataManager 方法	16
CRemoteData 屬性	18
連線類型屬性	20
撥號內容	21
COM 裝置屬性	22
NetBIOS 裝置屬性	25
透過 CAPI 2.0 裝置的 ISDN 屬性	25
CRemoteDataEx 物件	26
主控端功能的 Visual Basic 範例程式碼	27
CHostDataManager 方法	29
CHostData 屬性	31
連線類型屬性	32
AssignConnection(ConnectionType) 方法	34
UnassignConnection(ConnectionType) 方法	35
撥號內容	35
COM 裝置屬性	36
NetBIOS 裝置屬性	39
透過 CAPI 2.0 裝置的 ISDN 屬性	39

CHostDataEx 物件 40

 被控端功能的 Visual Basic 程式碼範例 46

Awrem32 函數 47

第 3 章

Visual C++ 物件定義 51

關於 Visual C++ 物件 51

CRemoteDataManager 方法 52

CRemoteData 物件 54

 Get 及 Set 方法 54

 主控端物件細節方法 57

 主控端物件方法 59

CRemoteDataEx 物件 60

 主控端功能的 Visual C++ 程式碼範例 60

CHostDataManager 方法 62

CHostData 物件 64

 Get 及 Set 方法 65

 被控端物件細節方法 66

 被控端物件方法 70

CHostDataEx 物件 71

 被控端功能 Visual C++ 的程式碼範例 73

Awrem32 函數 74

索引 77

與 Symantec pcAnywhere 搭配使用 OLE 自動化

本章包含以下主題：

- [關於 OLE 自動化](#)
- [關於 pcAnywhere 自動化伺服器](#)
- [您可以利用 pcAnywhere 自動化伺服器做什麼](#)
- [在開始之前](#)
- [何處有更多的資訊](#)

關於 OLE 自動化

OLE 自動化是一種技術，可讓您建立外部應用程式或其他開發工具 (例如程序檔或巨集)，以便在某個應用程式中控制與自動執行任何可使用之功能。

OLE 自動化由下列元件組成：

OLE 自動化伺服器	一種應用程式或軟體元件，提供可以由其他應用程式或開發工具存取或控制的功能。 例如，pcAnywhere 自動化伺服器就是一種 OLE 自動化伺服器。
OLE 自動化控制器	應用程式或開發工具，可存取與控制由 OLE 自動化伺服器所提供的元件 您可以使用任何支援 OLE 自動化的程式語言。最常見的兩種程式語言就是 Microsoft Visual Basic 及 Microsoft Visual C++。

外部應用程式會先連接至伺服器，然後要求存取一或數個它所發佈的介面，藉而存取「OLE 自動化伺服器」。介面是一個可以允許存取一或數個相關方法或屬性的進入點。在應用程式取得伺服器上的介面後，它就會繼續呼叫任何介面方法，就好像它是外部應用程式的一部份一樣。

關於 pcAnywhere 自動化伺服器

「pcAnywhere 自動化伺服器」可讓外部應用程式管理 pcAnywhere 主控端和被控端檔案，以達成自動化遠端遙控與檔案傳輸的工作。「pcAnywhere 自動化伺服器」的功能類似於 Symantec pcAnywhere 使用者介面的可程式化替用品，而且可以映射其多種預設行為。

例如，在 pcAnywhere 中建立被控端物件時，預設會指派第一個可用的數據機 TAPI 裝置。同樣地，使用「pcAnywhere 自動化伺服器」建立被控端物件，然後透過已指派的連線清單列舉時，就已指派了第一個可用的數據機 TAPI 裝置。

您可以利用 pcAnywhere 自動化伺服器做什麼

「pcAnywhere 自動化伺服器」可讓您對各種管理與生產力工作進行自動化處理。

您可使用「pcAnywhere 自動化伺服器」執行下列動作：

- 透過網路，在多台電腦上自動散佈與安裝軟體的更新檔案
- 排定時間在電腦間自動進行檔案傳輸，以供稽核或保存之用
- 在網路上每一部 pcAnywhere 被控端上，將名稱自動新增到允許的呼叫者清單中，或從中移除

本文件包含以 Visual Basic 及 C++ 兩種格式撰寫的數個範例，說明如何連接及使用「pcAnywhere 自動化伺服器」。

請參閱第 15 頁的「關於 [Visual Basic 物件](#)」。

請參閱第 51 頁的「關於 [Visual C++ 物件](#)」。

在開始之前

連接到「pcAnywhere 自動化伺服器」及其介面時，識別碼參數（亦即「全域唯一識別碼 (GUID)」）會被傳遞到自動化程式庫 API 函數。不同的 GUID 會被指派給「pcAnywhere 自動化伺服器」與每一個提供的介面。GUID 必須出現在系統登錄中，以便將外部應用程式連線到「pcAnywhere 自動化伺服器」及其介面。

如果在安裝 Symantec pcAnywhere 的電腦上執行外部應用程式，您可以自動註冊 GUID 項目。否則，您必須手動註冊 GUID 項目。

請參閱第 11 頁的「[自動註冊主控端引擎](#)」。

請參閱第 11 頁的「[手動註冊主控端引擎](#)」。

自動註冊主控端引擎

在您使用 OLE 用戶端第一次連線到另一台電腦之前，必須先自行註冊主控端引擎。當您在 Symantec pcAnywhere 中啟動主控端物件時，即可自動完成此作業。

自動註冊主控端引擎

- 1 若要開啟 Symantec pcAnywhere，請按照下列任一方法進行：
 - 在桌面上，連接兩下 Symantec pcAnywhere 程式圖示。
 - 在 Windows 工作列上，按下「開始」>「程式集」>Symantec pcAnywhere。
- 2 在「pcAnywhere 管理員」視窗中，按下「主控端」。
- 3 連接兩下主控端連線項目。

這個程序會自動註冊主控端引擎。您不需要完成此連線。

手動註冊主控端引擎

如果沒有在執行外部應用程式的電腦上安裝 pcAnywhere，您必須藉由執行「pcAnywhere 自動化伺服器」執行檔 (Winawsvr.exe) 來手動註冊 GUID。您只需要執行此執行檔一次，就可將 GUID 加入到登錄中。Winawsvr.exe 檔案位於安裝目錄中。

存取 pcAnywhere 自動化伺服器

您可以使用支援「OLE 自動化」的任何語言平台來存取「pcAnywhere 自動化伺服器」。支援「OLE 自動化」最常見的兩種語言平台就是 Visual Basic 及 Visual C++。

撰寫這兩種平台之程式碼的原則很類似，但是在 Visual Basic 環境中，有很多低層次的工作都是由 Visual Basic 執行時期系統在幕後執行的。

使用 Visual Basic 存取 pcAnywhere 自動化伺服器

Visual Basic 程式語言內建的支援，可與 OLE 自動化伺服器 (例如「pcAnywhere 自動化伺服器」) 進行互動。您可以建立一個標準的 Exe 專案，然後利用每一種方法輸入程式碼，來存取「pcAnywhere 自動化伺服器」。Visual Basic 在來源檔案中採用高層次方法呼叫，並且在內部將這些呼叫延伸至相對應的低層次 OLE 自動化方法呼叫。

使用 Visual Basic 存取 pcAnywhere 自動化伺服器

- 1 為每個您想存取的 pcAnywhere 物件新增一對「物件」變數。

例如，在使用主控端物件時，假定 (DIM) **RemoteDataManager** 和 **RemoteDataObject** 為「物件」。

- 2 使用 RemoteDataManager 來連接主控端物件的資料管理程式。

例如，使用 **WINAWSVR.REMOTEDATAMANAGER** 當作參數，呼叫 **CreateObject** 方法。

Visual Basic 會使用文字參數，在登錄檔中找出管理員的識別碼，並將介面傳回該管理員。

- 3 有了有效的資料管理員物件之後，請用它來執行以下任何一項操作：

- 決定目前的目錄。
- 切換到另一個目錄。
- 列舉目前目錄中相關的資料物件檔案。
- 建立、擷取或刪除資料物件檔案。

- 4 建立或擷取資料物件之後，您可以取得或設定物件的屬性。

Visual Basic 語法並不會使用內容的名稱來分別取得及設定其數值。相反地，有關指定運算元之內容的位置會決定加底線的方法呼叫是 **Get** 或 **Set**。

下列範例介紹 **Get** 及 **Set**：

- 若要取得物件的電話號碼數值，請將內容名稱放在指定運算元的右邊。
例如，`s = RemoteData.PhoneNumber()`，其中 `s` 是字串變數。
- 若要設定電話號碼，請將屬性名稱放在指派運算子的左邊。
例如，`RemoteData.PhoneNumber = "555-1212"`

使用 Visual C++ 存取 pcAnywhere 自動化伺服器

「pcAnywhere 自動化伺服器」使用類型程式庫，將其介面和方法的相關資訊，提供給使用 Visual C++ 撰寫的自動化用戶端。這些類型程式庫使用 Microsoft Foundation Classes (MFC)，這可以使用 Visual C++ ClassWizard 匯入到您的應用程式。

類型程式庫中的資料管理員類別提供了獲取 pcAnywhere 自動化伺服器介面所需的功能，並在此介面的關聯物件類別上，執行高階作業。

使用資料管理員物件執行以下操作：

- 決定或變更目前的目錄。
- 透過目前目錄中資料物件檔案的清單列舉。

■ 建立、擷取或刪除已命名的物件。

一旦建立或擷取了之後，物件便會使用相關聯的資料物件類別來檢查或修改其任何的顯示屬性。這些屬性中，大部份都是藉由一對以 **Get** 或 **Set** 這二個字開頭的方法來啟動的。例如，使用者會呼叫 **GetPhoneNumber** 方法來檢查物件的現有電話號碼內容，並呼叫 **SetPhoneNumber** 來設定它。

匯入與檢視類別

「pcAnywhere 自動化伺服器」使用下列的類型程式庫：

Winawsvr.tlb	提供連線到「pcAnywhere 自動化伺服器」與存取其介面所需的資訊
Awrem32.tlb	提供控制 pcAnywhere 連線所需的資訊

下列程序說明如何從「pcAnywhere 自動化伺服器」類型程式庫，匯入類別定義到 MFC 應用程式，然後檢視新增到您應用程式的類別。

匯入類別

- 1 在 Visual C++ 中建立 MFC 應用程式。
- 2 在「檢視」功能表上，按下「**類別精靈**」。
- 3 在「類別精靈」對話方塊中，按下「**新增類別**」，然後按下「**從類型程式庫**」。
- 4 連按兩下 **winawsvr.tlb**。
- 5 在「確認類別」對話方塊中，按下「**確定**」以匯入所有的類別定義。
- 6 在「類別精靈」對話方塊中，按下「**新增類別**」，然後按下「**從類型程式庫**」。
- 7 連按兩下 **awrem32.tlb**。
- 8 在「確認類別」對話方塊中，按下「**確定**」以匯入所有的類別定義。
- 9 在「類別精靈」對話方塊中，按下「**確定**」以完成匯入程序。

類別會新增到應用程式。這些類別可讓您操作物件及管理連線。

從類型程式庫匯入類別定義，也會增加應用程式所支援的檔案。這些檔案包含類別定義以及供「pcAnywhere 自動化伺服器」使用的原始程式碼。

請參閱第 14 頁的「[檢視類別定義與實行檔案](#)」。

檢視已新增的類別

- 1 在 Visual C++ 中，開啟 MFC 應用程式。
- 2 在「工作區」視窗中，按下「**類別檢視**」標籤。

檢視類別定義與實行檔案

將「pcAnywhere 自動化伺服器」類型程式庫匯入到應用程式時，也會新增下列檔案：

- Winawsvr.h
- Winawsvr.cpp
- Awrem32.h
- Awrem32.cpp

這些檔案包含類別定義以及供「pcAnywhere 自動化伺服器」使用的原始程式碼。您不必編輯這些檔案，但是每一個包含有呼叫「pcAnywhere 自動化伺服器」方法的應用程式來源檔案，都必須包括 Winawsvr.h。

檢視類別定義與實行檔案

- 1 在 Visual C++ 中，開啟 MFC 應用程式。
- 2 在「工作區」視窗中，按下「檔案檢視」標籤。

執行被控端與主控端 OLE 物件

Symantec pcAnywhere 會要求您將所有的被控端與主控端物件存放在預設的資料目錄中。在您啟動使用 OLE 建立的被控端或主控端物件之前，請確認該物件位於 pcAnywhere 預設的資料目錄中。預設目錄在 Windows XP 中為 All Users 資料夾，在 Vista 中為 ProgramData 資料夾。

何處有更多的資訊

如需關於 OLE 自動化的詳細資訊，請參閱下列的參考文件：

- Box, Don.1998. *Essential COM*.Reading, Mass.:Addison-Wesley.
- Horton, Ivor.1997. *Beginning MFC Programming*.Birmingham, UK.:Wrox Press.
- Rogerson, Dale.1997. *Inside COM*.Redmond, Wash.:Microsoft Press.
- Templeman, Julian.1997. *Beginning MFC COM Programming*.Birmingham, UK.:Wrox Press.

Visual Basic 物件定義

本章包含以下主題：

- [關於 Visual Basic 物件](#)
- [CRemoteDataManager 方法](#)
- [CRemoteData 屬性](#)
- [CRemoteDataEx 物件](#)
- [CHostDataManager 方法](#)
- [CHostData 屬性](#)
- [CHostDataEx 物件](#)
- [Awrem32 函數](#)

關於 Visual Basic 物件

「pcAnywhere 自動化伺服器」提供下列支援 OLE 自動化的元件：

Winawsvr	提供連線到「pcAnywhere 自動化伺服器」與存取其介面所需的資訊
Awrem32	提供控制 pcAnywhere 連線所需的資訊

[表 2-1](#) 說明組成 Winawsvr 的各個物件。

表 2-1 Winawsvr 物件

物件	描述	參考文件
CRemoteDataManager	提供用來建立、開啟、修改、儲存及刪除 CRemoteData 物件的方法	請參閱第 16 頁的「 CRemoteDataManager 方法 」。

物件	描述	參考文件
CRemoteData	定義存取與控制 pcAnywhere 主控端功能的參數	請參閱第 18 頁的「CRemoteData 屬性」。 請參閱第 26 頁的「CRemoteDataEx 物件」。
CHostDataManager	提供用來建立、開啟、修改、儲存及刪除 CHostData 物件的方法	請參閱第 29 頁的「CHostDataManager 方法」。
CHostData	定義存取與控制 pcAnywhere 被控端功能的參數	請參閱第 31 頁的「CHostData 屬性」。 請參閱第 40 頁的「CHostDataEx 物件」。

Awrem32 有一個物件是由 8 個介面組成，支援遠端遙控與檔案傳輸階段作業。
請參閱第 47 頁的「Awrem32 函數」。

某些功能已不再支援。但是，物件定義仍可供給先前版本的 pcAnywhere 使用。
對於要求密碼的功能，可以設定密碼，但無法擷取。這是為了安全起見。

CRemoteDataManager 方法

CRemoteDataManager 方法提供存取與控制 CRemoteData 物件的參數與回傳值。

表 2-2 CRemoteDataManager 方法

CRemoteDataManager 方法	參數或傳回值	描述
CurrentDirectory()	傳回值 = 字串	傳回儲存 pcAnywhere 主控端物件之目前目錄的完整路徑名稱
ChangeDirectory(NewDirectory)	參數 = NewDirectory	變更儲存 pcAnywhere 主控端物件的目前目錄
FindFirst(Pattern)	參數 = 樣式字串	尋找目前目錄中的第一個 pcAnywhere 主控端物件檔案 (*.chf)；檔案名稱樣式會根據指定的檔案名稱樣式(星號[*]會找出目前目錄中所有的檔案)進行篩選。
FindFirst(Name string)	參數 = 名稱字串	傳回符合指定樣式之主控端物件檔案 (*.chf) 完整路徑名稱的緩衝區。
FindNext(Name)	參數 = 名稱字串	傳回符合 FindFirst() 原始呼叫中所指定樣式之主控端物件檔案完整路徑名稱的緩衝區。

CRemoteDataManager 方法	參數或傳回值	描述
FindNext(Name)	傳回值 = Boolean	如果找到另一個符合 FindFirst() 呼叫所指定樣式的主控端物件檔案，則是 TRUE。相符檔案的完整路徑名稱儲存在 Name
RetrieveObject(Name)	參數 = 名稱字串	要載入的完整主控端物件檔案名稱
RetrieveObject(AccessMode)	參數 = 整數形式的 AccessMode	指定要如何使用此物件。這與密碼保護有關 選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
RetrieveObject(Password)	參數 = 密碼字串	物件密碼。可以是空值
RetrieveObjectEx(Name)	參數 = 名稱字串	要載入的完整主控端物件檔案名稱
RetrieveObjectEx(AccessMode)	參數 = 整數形式的 AccessMode	指定要如何使用此物件。這與密碼保護有關 選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
RetrieveObjectEx(Password)	參數 = 密碼字串	物件密碼。可以是空值
RetrieveObjectEx	傳回值 = 物件	指定檔案中的 CRemoteDataEx 物件
CreateObject(Name)	參數 = 名稱字串	建立 CRemoteData 物件，並將 LPDISPATCH 指標傳回給它。參數是新物件的完整主控端物件檔案名稱
CreateObject	傳回值 = 物件	CRemoteData
CreateObjectEx(Name)	參數 = 名稱字串	建立 CRemoteDataEx 物件，並將 LPDISPATCH 指標傳回給它。參數是新物件的完整主控端物件檔案名稱
CreateObjectEx	傳回值 = 物件	CRemoteDataEx

CRemoteDataManager 方法	參數或傳回值	描述
DeleteObject(Name)	參數 = 名稱字串	要刪除之物件的完整主控端物件檔案名稱
DeleteObject(Password)	參數 = 密碼字串	物件密碼
DeleteObject	傳回值 = Boolean	如果刪除了物件，則為 TRUE。

CRemoteData 屬性

以下顯示 CRemoteData 物件可用的屬性與參數。

在表 2-3 中，以實際值取代角括號中的資訊。

表 2-3 CRemoteData 屬性與參數

屬性	參數	描述
<CRemoteData>.ComputerName(String)	String	設定被控端電腦的電腦名稱或 IP 位址。
<CRemoteData>.PhoneNumber(String)	String	設定被控端電腦的電話號碼。
<CRemoteData>.UseDialingProperties(Bool)	Bool	設定系統撥號內容。
<CRemoteData>.RedialCount(Integer)	Integer	設定連續重撥多少次之後便取消呼叫。
<CRemoteData>.RedialDelay(Integer)	Integer	設定隔幾秒重撥一次。
<CRemoteData>.AutoLoginName(String)	String	設定自動登入的使用者名稱。 關於使用網域登入的詳細資訊： 請參閱第 26 頁的「CRemoteDataEx 物件」。
<CRemoteData>.AutoLoginPassword(String)	String	設定在主控端物件中自動登入的密碼。 為了安全起見，「pcAnywhere 自動化伺服器」不提供讀取密碼值的功能，因此不會傳回任何密碼值。
<CRemoteData>.Password(String)	String	設定搭配 ExecuteProtection、ReadProtection 與 WriteProtection 設定值使用的主控端物件密碼。 為了安全起見，「pcAnywhere 自動化伺服器」不提供讀取密碼值的功能，因此不會傳回任何密碼值。
<CRemoteData>.ExecuteProtection(Bool)	Bool	設定執行物件的密碼需求。根據密碼設定。

屬性	參數	描述
<CRemoteData>.ReadProtection(Bool)	Bool	設定檢視主控端物件屬性的密碼需求。根據密碼設定。
<CRemoteData>.WriteProtection(Bool)	Bool	設定將變更儲存到主控端物件的密碼需求。根據密碼設定。
<CRemoteData>.LogSession(Bool)	Bool	啟用與停用階段作業記錄。
<CRemoteData>.RecordFile(String)	String	設定用來記錄使用中階段作業之檔案的位置所在完整目錄與名稱。
<CRemoteData>.RecordSession(Bool)	Bool	啟用與停用自動記錄階段作業。
<CRemoteData>.ReadObject(String)	String	設定物件的密碼。使用此屬性重新整理主控端物件的本機資料複本。
<CRemoteData>.WriteObject(String)	String	設定物件的密碼。使用此屬性建立主控端物件，或將您對主控端物件所做的變更寫入。

下表顯示 CRemoteData 的屬性與傳回值。

在表 2-4 中，以實際值取代角括號中的資訊。

表 2-4 CRemoteData 屬性與傳回值

屬性	回傳值	描述
String = <CRemoteData>.ComputerName	String	傳回被控端電腦的電腦名稱或 IP 位址
String = <CRemoteData>.PhoneNumber	String	傳回被控端電腦的電話號碼。
Bool = <CRemoteData>.UseDialingProperties	Bool	傳回在主控端物件中所設定的系統撥號內容
Integer = <CRemoteData>.RedialCount	Integer	傳回在主控端物件中所設定的重撥次數
Integer = <CRemoteData>.RedialDelay	Integer	傳回每兩次重撥之間的時間隔秒數
String = <CRemoteData>.AutoLoginName	String	傳回自動登入使用的登入名稱
Bool = <CRemoteData>.ExecuteProtection	Bool	傳回 ExecuteProtection 設定的值
Bool = <CRemoteData>.ReadProtection	Bool	傳回 ReadProtection 設定的值
Bool = <CRemoteData>.WriteProtection	Bool	傳回 WriteProtection 設定的值
Bool = <CRemoteData>.LogSession	Bool	如果已啟動階段作業記錄，則傳回 TRUE
String = <CRemoteData>.RecordFile	String	傳回階段作業記錄檔案的完整路徑與名稱
Bool = <CRemoteData>.RecordSession	Bool	傳回階段作業記錄設定的值

連線類型屬性

下表定義連線類型的屬性與參數。

在表 2-5 中，以實際值取代角括號中的資訊。

表 2-5 連線類型屬性與參數

屬性	參數	描述
<CRemoteData>.ConnectionType(String)	String	設定主控端的連線類型。傳入的值必須是由 FirstConnectionType() 與 NextConnectionType() 函數所定義的有效連線類型。 以下是有效連線類型的範例： ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN 透過 CAPI 2.0 ■ 數據機名稱 (與出現在電腦上的名稱相同)

下表定義連線類型的屬性與傳回值。

在表 2-6 中，以實際值取代角括號中的資訊。

表 2-6 連線類型屬性與傳回值

屬性	回傳值	描述
String = <CRemoteData>.ConnectionTypes	String	傳回主控端物件的連線類型。

屬性	回傳值	描述
Integer = <CRemoteData>.ConnectionType	Integer	傳回可用連線類型的個數。 以下是有效連線類型的範例： ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN 透過 CAPI 2.0 ■ 數據機名稱 (與出現在電腦上的名稱相同)
String = <CRemoteData>.FirstConnectionType	String	傳回第一個可用的連線類型。
String = <CRemoteData>.NextConnectionType	String	傳回下一個可用的連線類型。列舉所有連線類型時，可針對 <CRemoteData>.ConnectionType 中所設定的連線類型個數依序呼叫此方法。
Bool = <CRemoteData>.FindConnectionType(ConnectionType)	Bool	若在可用連線類型清單中找到此處指定的連線類型，則傳回 TRUE。

撥號內容

下表定義設定數據機連線撥號內容的屬性與參數。

在表 2-7 中，以實際值取代角括號中的資訊。

表 2-7 撥號內容的屬性與參數

屬性	參數	描述
<CRemoteData>.AreaCode(String)	String	設定數據機連線撥號內容的區碼
<CRemoteData>.CountryCode(String)	String	設定數據機連線撥號內容的國碼

下表定義數據機撥號內容的屬性與傳回值。

在表 2-8 中，以實際值取代角括號中的資訊。

表 2-8 撥號內容的屬性與傳回值

屬性	回傳值	描述
String = <CRemoteData>.AreaCode	String	傳回撥號內容的區碼。

屬性	回傳值	描述
String = <CRemoteData>.CountryCode	String	傳回撥號內容的國碼。
Integer = <CRemoteData>.CountryCodes	Integer	傳回可用國碼的數量。
String = <CRemoteData>.FirstCountryCode	String	傳回作業系統中所列的第一個可用國碼。
String = <CRemoteData>.NextCountryCode	String	傳回下一個可用的國碼。列舉所有國碼時，可針對在 <CRemoteData.CountryCodes> 中所設定的國碼個數依序呼叫此方法。

COM 裝置屬性

下表說明讓您自訂數據機與其他 COM 連線通訊埠設定值的屬性與參數。
在表 2-9 中，以實際值取代角括號中的資訊。

表 2-9COM 裝置屬性與參數

屬性	參數	描述
<CRemoteData>.ComParity(String)	String	設定通訊同位檢查 以下為有效的值： ■ <無> ■ 奇 ■ 偶 ■ 標記 ■ 空格
<CRemoteData>.ComFlowControl(String)	String	設定 COM 連線類型的流量控制 以下為有效的值： ■ <無> ■ XONXOFF ■ RTS/CTS ■ 二者

屬性	參數	描述
<CRemoteData>.ComStartedBy(String)	String	設定 COM 連線類型的起始設定 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI) ■ 接收兩次 <CR> 訊號 ■ 數據機回應
<CRemoteData>.ComEndedBy(String)	String	設定 COM 連線類型的結束設定 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI)
<CRemoteData>.ComSpeed(Long)	Long	含有最大 COM 速度設定值 以下為有效的值： ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

下表說明 COM 裝置的屬性與傳回值。

在表 2-10 中，以實際值取代角括號中的資訊。

表 2-10 COM 裝置屬性與傳回值

屬性	回傳值	描述
String = <CRemoteData>.ComParity	String	傳回以下任一個通訊同位檢查的值： ■ <無> ■ 奇 ■ 偶 ■ 標記 ■ 空格
String = <CRemoteData>.ComFlowControl	String	傳回主控端物件的 Com Flow 設定值 以下為有效的值： ■ <無> ■ XONXOFF ■ RTS/CTS ■ 二者
String = <CRemoteData>.ComStartedBy	String	傳回 COM 連線類型的 Com Start 控制 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI) ■ 接收兩次 <CR> 訊號 ■ 數據機回應
String = <CRemoteData>.ComEndedBy	String	傳回 COM 連線類型的 Com End 控制 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI)

屬性	回傳值	描述
Long = <CRemoteData>.ComSpeed	Long	傳回主控端物件 Com Speed 目前的設定值 以下為有效的值： ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

NetBIOS 裝置屬性

下表定義 NetBIOS 網路裝置的屬性與參數。
在表 2-11 中，以實際值取代角括號中的資訊。

表 2-11 NetBIOS 屬性與參數

屬性	參數	描述
<CRemoteData>.LanaNumber(Integer)	Integer	設定 NetBIOS 連線的 LAN Adapter (LANA) 編號

下表定義 NetBIOS 的屬性與傳回值。
在表 2-12 中，以實際值取代角括號中的資訊。

表 2-12 NetBIOS 屬性與傳回值

屬性	回傳值	描述
Integer = <CRemoteData>.LanaNumber	Integer	傳回 NetBIOS 連線 LAN Adapter (LANA) 編號目前的設定值

透過 CAPI 2.0 裝置的 ISDN 屬性

下表定義歐洲 ISDN 連線的屬性與參數。
在表 2-13 中，以實際值取代角括號中的資訊。

表 2-13ISDN 透過 CAPI 2.0 屬性與參數

屬性	參數	描述
<CRemoteData>.CapiChannelBonding(Bool)	Bool	啟用或停用 ISDN CAPI 裝置的通道連接
<CRemoteData>.CapiExtensions(String)	String	設定通訊所需的其他 CAPI 延伸功能

下表定義歐洲 ISDN 連線的屬性與傳回值。
在表 2-14 中，以實際值取代角括號中的資訊。

表 2-14ISDN 透過 CAPI 2.0 屬性與傳回值

屬性	回傳值	描述
Bool= <CRemoteData>.CapiChannelBonding	Bool	傳回主控端物件中目前 ISDN CAPI 通道連接的設定值
String = <CRemoteData>.CapiExtensions	String	傳回主控端物件目前的 CAPI 延伸功能清單

CRemoteDataEx 物件

CRemoteDataEx 物件所含功能與 CRemoteData 物件相同，但多了一些額外的功能。
下表說明屬性與參數。
在表 2-15 中，以實際值取代角括號中的資訊。

表 2-15CRemoteDataEx 參數

屬性	參數	描述
<CRemoteData>.PrivateKey(String)	String	設定要使用之私密金鑰容器的名稱。
<CRemoteData>.CertificateName(String)	String	設定要使用之私密金鑰的共用名稱。
<CRemoteData>.EncryptionLevel(Byte)	Byte	設定加密層次。 以下為有效的值： ■ -1: 無 ■ 0: pcAnywhere ■ 1: 對稱性 ■ 2: 公開金鑰
<CRemoteData>.DenyLowerEncrypt(Bool)	Bool	定義主控端電腦是否允許與被控端電腦的連線使用較低的加密層次。

屬性	參數	描述
<CRemoteData>.AutoDomain(String)	String	設定自動登入的網域名稱。此選項與 NT 驗證和 Windows 驗證搭配使用。

下表定義 CRemoteDataEx 物件的屬性與傳回值。

在表 2-16 中，以實際值取代角括號中的資訊。

表 2-16 CRemoteDataEx 屬性與傳回值

屬性	回傳值	描述
String = <CRemoteData>.PrivateKey	String	傳回目前使用中私密金鑰容器的名稱。
String = <CRemoteData>.CertificateName	String	傳回使用中私密金鑰容器的共用名稱。
Byte = <CRemoteData>.EncryptionLevel	Byte	傳回以下任一種加密設定： ■ -1: 無 ■ 0: pcAnywhere ■ 1: 對稱性 ■ 2: 公開金鑰
Bool = <CRemoteData>.DenyLowerEncrypt	Bool	傳回拒絕較低層次加密設定的值。
String = <CRemoteData>.AutoDomain	String	傳回自動登入的網域名稱設定。此選項與 NT 驗證和 Windows 驗證搭配使用。

主控端功能的 Visual Basic 範例程式碼

以下的 Visual Basic 範例程式碼會擷取一個主控端資料物件，並修改其屬性：

```
Private Sub Command1_Click()  
Dim RemoteDataManager as Object  
Dim RemoteData as Object  
Dim s as string  
  
'建立 CRemoteDataManager 物件  
Set RemoteDataManager = CreateObject(WINAWSVR.REMOTEDATAMANAGER)  
  
'顯示和變更目前目錄  
s = RemoteDataManager.CurrentDirectory()  
MsgBox ( s )  
RemoteDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
```

```
s = RemoteDataManager.CurrentDirectory()
MsgBox ( s )
```

'擷取被控端資料物件

```
Set RemoteData = RemoteDataManager.RetrieveObjectEx("pod.CHF",
    2, 0)
```

'顯示某些屬性

```
s = RemoteData.AreaCode()
MsgBox (s)
s = RemoteData.PhoneNumber()
MsgBox (s)
```

'設定某些屬性

```
RemoteData.AreaCode = "212"
RemoteData.PhoneNumber = "555-5555"
```

'將物件寫入至磁碟

```
RemoteData.WriteObject (0)
End Sub
```

使用 FindFirst 及 FindNext 方法可顯示目錄中的主控端檔案，如下所示：

```
Private Sub Command5_Click()

Dim RemoteDataManager as Object
Dim RemoteData as Object
Dim s as string

Set RemoteDataManager =
CreateObject("WINAWSVR.REMOTEDATAMANAGER")
RemoteDataManager.ChangeDirectory
("C:\dev\bin.w32\data")
RemoteDataManager.FindFirst "*", s
MsgBox (s)
RemoteDataManager.FindNext s
MsgBox (s)
End Sub
```

建立一個主控端物件。將連線類型設定為 TCP/IP，並將電腦名稱設定為 Host1，然後啟動：

```
Private Sub Command6_Click()  
Dim RemoteDataManager as Object  
Dim RemoteData as Object  
Dim s as string  
  
Set RemoteDataManager =  
CreateObject("WINAWSVR.REMOTEDATAMANAGER")  
MsgBox (RemoteDataManager.CurrentDirectory())  
  
RemoteDataManager.ChangeDirectory ("C:\dev\bin.w32\data")  
MsgBox (RemoteDataManager.CurrentDirectory())  
Set RemoteData = RemoteDataManager.CreateObject("test")  
RemoteData.ConnectionType = "TCP/IP"  
RemoteData.ComputerName = "Host1"  
s = RemoteData.ConnectionType  
MsgBox (s)  
s = RemoteData.ComputerName  
MsgBox (s)  
RemoteData.WriteObject (0)  
End Sub
```

CHostDataManager 方法

CHostDataManager 方法提供存取與控制 CHostData 物件的參數與回傳值。

CHostDataManager 方法	參數或傳回值	描述
CurrentDirectory()	傳回值 = 字串	儲存 pcAnywhere 被控端物件的目前 pcAnywhere 資料目錄完整路徑名稱
FindFirst(Pattern)	參數 = 樣式字串	用於過濾物件檔案的檔案名稱樣式 (使用 [*] 會在目前目錄中尋找所有檔案)。
FindFirst(Name string)	參數 = 名稱字串	傳回符合指定樣式之被控端物件檔案完整路徑名稱的緩衝區
FindFirst	傳回值 = Boolean	如果找到符合指定樣式的被控端物件，則為 TRUE。相符檔案的完整路徑名稱儲存在 Name
FindNext(Name)	參數 = 名稱字串	傳回符合 FindFirst() 原始呼叫中指定樣式之被控端物件檔案完整路徑名稱的緩衝區

CHostDataManager 方法	參數或傳回值	描述
FindNext	傳回值 = Boolean	如果找到另一個符合在 FindFirst() 呼叫中指定樣式的被控端物件檔案，則是 TRUE。相符檔案的完整路徑名稱儲存在 Name
RetrieveObject(Name)	參數 = 名稱字串	要載入的完整被控端物件檔案名稱
RetrieveObject(AccessMode)	參數 = 整數形式的 AccessMode	指定要如何使用此物件。這與密碼保護有關。選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
RetrieveObject(Password)	參數 = 密碼字串	物件密碼。可以是空值
RetrieveObject	傳回值 = 物件	指定檔案中的 CHostData 物件
RetrieveObjectEx(Name)	參數 = 名稱字串	要載入的完整被控端物件檔案名稱
RetrieveObjectEx(AccessMode)	參數 = 整數形式的 AccessMode	指定要如何使用此物件。這與密碼保護有關。選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
RetrieveObjectEx(Password)	參數 = 密碼字串	物件密碼。可以是空值
RetrieveObjectEx	傳回值 = 物件	指定檔案中的 CHostDataEx 物件
CreateObject(Name)	參數 = 名稱字串	建立 CHostData 物件，並將 LPDISPATCH 指標傳回給它。參數是新物件的完整被控端物件檔案名稱
CreateObject	傳回值 = 物件	CHostData
CreateObjectEx(Name)	參數 = 名稱字串	建立 CHostDataEx 物件，並將 LPDISPATCH 指標傳回給它。參數是新物件的完整被控端物件檔案名稱
CreateObjectEx	傳回值 = 物件	CHostDataEx
DeleteObject(Name)	參數 = 名稱字串	要刪除之物件的完整被控端物件檔案名稱
DeleteObject(Password)	參數 = 密碼字串	物件密碼

CHostDataManager 方法	參數或傳回值	描述
DeleteObject	傳回值 = Boolean	如果刪除了物件，則為 TRUE。
Launch(Name)	參數 = 名稱字串	啟動開啟 pcAnywhere 被控端終端機視窗的被控端物件檔案。參數是所需啟動物件的完整被控端物件檔案
啟動	傳回值 = Boolean	如果順利啟動，則為 TRUE

CHostData 屬性

下表說明 CHostData 物件可用的屬性與參數。
在表 2-17 中，以實際值取代角括號中的資訊。

表 2-17 CHostData 屬性與參數

屬性	參數	描述
<CHostData>.PhoneNumber(String)	String	設定被控端電腦的電話號碼。
<CHostData>.UseDialingProperties(Bool)	Bool	設定系統撥號內容。
<CHostData>.RedialCount(Integer)	Integer	設定連續重撥多少次之後便取消呼叫。
<CHostData>.RedialDelay(Integer)	Integer	設定隔幾秒重撥一次。
<CHostData>.Password(String)	String	設定搭配 ExecuteProtection、ReadProtection 與 WriteProtection 設定值使用的被控端物件密碼。 為了安全起見，「pcAnywhere 自動化伺服器」不提供讀取密碼值的功能，因此不會傳回任何密碼值。
<CHostData>.ExecuteProtection(Bool)	Bool	設定執行物件的密碼需求。根據密碼設定。
<CHostData>.ReadProtection(Bool)	Bool	設定檢視被控端物件屬性的密碼需求。根據密碼設定。
<CHostData>.WriteProtection(Bool)	Bool	設定將變更儲存到被控端物件的密碼需求。根據密碼設定。
<CHostData>.LogSession(Bool)	Bool	啟用與停用階段作業記錄。
<CHostData>.RecordFile(String)	String	設定用來記錄使用中階段作業之檔案的位置所在完整目錄與名稱。

屬性	參數	描述
<CHostData>.RecordSession(Bool)	Bool	啟用與停用自動記錄階段作業。
<CHostData>.ReadObject(String)	String	設定物件的密碼。您可以使用此屬性重新整理被控端物件的本機資料複本。
<CHostData>.WriteObject(String)	String	設定物件的密碼。使用此屬性建立被控端物件，或將您對被控端物件所做的變更寫入。

下表說明 CHostData 的屬性與傳回值。

在表 2-18 中，以實際值取代角括號中的資訊。

表 2-18 CHostData 屬性與傳回值

屬性	回傳值	描述
String = <CHostData>.PhoneNumber	String	傳回被控端電腦的電話號碼。
Bool = <CHostData>.UseDialingProperties	Bool	傳回在被控端物件中所設定之系統撥號內容的用途
Integer = <CHostData>.RedialCount	Integer	傳回在被控端物件中所設定的重撥次數
Integer = <CHostData>.RedialDelay	Integer	傳回每兩次重撥之間的時間秒數
Bool = <CHostData>.ExecuteProtection	Bool	傳回 ExecuteProtection 設定的值
Bool = <CHostData>.ReadProtection	Bool	傳回 ReadProtection 設定的值
Bool = <CHostData>.WriteProtection	Bool	傳回 Write Protection 設定的值
Bool = <CHostData>.LogSession	Bool	如果已啟動階段作業記錄，則傳回 TRUE
String = <CHostData>.RecordFile	String	傳回階段作業記錄檔案的完整路徑與名稱
Bool = <CHostData>.RecordSession	Bool	傳回階段作業記錄設定的值

連線類型屬性

下表說明連線類型的屬性與參數。

在表 2-19 中，以實際值取代角括號中的資訊。

表 2-19 連線類型屬性與參數

屬性	參數	描述
<CHostData>.ConnectionType(String)	String	設定被控端的連線類型。傳入的值必須是由 FirstConnectionType() 與 NextConnectionType() 函數所定義的有效連線類型。 以下是有效連線類型的範例： ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN 透過 CAPI 2.0 ■ 數據機名稱 (與出現在電腦上的名稱相同)
<CHostData>.AssignConnection(String)	String	將字串中所定義的連線類型設定為使用中狀態。
<CHostData>.UnassignConnection(String)	String	將在字串中所定義的連線類型設定為非使用中狀態。

下表說明連線類型的屬性與傳回值。

在表 2-20 中，以實際值取代角括號中的資訊。

表 2-20 連線類型屬性與傳回值

屬性	回傳值	描述
String = <CHostData>.ConnectionTypes	String	傳回被控端物件的連線類型。

屬性	回傳值	描述
Integer = <CHostData>.ConnectionType	Integer	傳回可用連線類型的個數。 以下是有效連線類型的範例： ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN 透過 CAPI 2.0 ■ 數據機名稱 (與出現在電腦上的名稱相同)
String = <CHostData>.FirstConnectionType	String	傳回第一個可用的連線類型。
String = <CHostData>.NextConnectionType	String	傳回下一個可用的連線類型。列舉所有連線類型時，可針對在 <CHostData>.ConnectionTypes 中所設定的連線類型個數，依序呼叫此方法。
Bool = <CHostData>.FindConnectionType(ConnectionType)	Bool	若在可用連線類型清單中找到此處指定的連線類型，則傳回 TRUE。
Integer = <CHostData>.MaxAssignedConnections	Integer	傳回可在此被控端上啟動的連線類型個數上限。
String = <CHostData>.FirstAssignedConnection	String	傳回第一個被指派的使用中連線類型。
String = <CHostData>.NextAssignedConnection	String	傳回下一個被指派的使用中連線類型。
Bool = <CHostData>.FindAssignedConnection(ConnectionType)	Bool	若傳入的連線類型符合任何使用中的連線類型，則傳回 TRUE。

AssignConnection(ConnectionType) 方法

AssignConnection(Connection Type) 方法將所要求的連線類型放在被控端物件的指定連線類型的清單上，並且在處理後續的特定裝置方法呼叫時，使它成為目前的連線類型。

如果要求的連線類型已經在指定連線的清單上，指定連線的清單將不會變更。只會將目前的連線類型變更為要求的類型。通常在取得和設定連線專有設定值的過程中，都會在相同的物件上呼叫好幾次 AssignConnection 方法。

如果電腦上有傳入的連線類型且順利指派或已經指派，則 `AssignConnection` 會傳回 `TRUE`。如果電腦上沒有要求的連線類型，或是目前指派的連線數已達允許的上限，則它會傳回 `FALSE`。

`pcAnywhere` 被控端物件最多可以支援兩種指定連線類型。如果 `AssignConnection` 方法偵測到您嘗試超越此限制，便會傳回 `FALSE`。

下表定義 `AssignConnection` 參數：

參數	描述
<code>ConnectionType as string</code>	要指定之連線裝置類型的名稱

下表定義 `AssignConnection` 傳回值：

回傳值	描述
<code>Boolean</code>	如果這個裝置類型是可用的，而且尚未達到所允許的指定連線數上限，則為 <code>TRUE</code>

UnassignConnection(ConnectionType) 方法

如果可順利地將傳入的連線類型從已指定的連線類型清單中移除，`UnassignConnection(ConnectionType)` 方法便會傳回 `TRUE`。

下表定義 `UnassignConnection` 參數：

參數	描述
<code>ConnectionType as string</code>	要取消之連線裝置類型的名稱

下表定義 `UnassignConnection` 傳回值：

回傳值	描述
<code>Boolean</code>	如果順利取消指定了這個裝置類型，則為 <code>TRUE</code>

撥號內容

下表定義設定數據機連線撥號內容的屬性與傳回值。

在表 2-21 中，以實際值取代角括號中的資訊。

表 2-21 數據機撥號內容的屬性與參數

屬性	參數	描述
<CHostData>.AreaCode(String)	String	設定數據機連線的區碼
<CHostData>.CountryCode(String)	String	設定數據機連線的國碼

下表定義數據機撥號內容的屬性與傳回值。

在表 2-22 中，以實際值取代角括號中的資訊。

表 2-22 數據機撥號內容的屬性與傳回值

屬性	傳回值	描述
String = <CHostData>.AreaCode	String	傳回區碼。
String = <CHostData>.CountryCode	String	傳回國碼。
Integer = <CHostData>.CountryCodes	Integer	傳回可用國碼的數量。
String = <CHostData>.FirstCountryCode	String	傳回作業系統中所列的第一個可用國碼。
String = <CHostData>.NextCountryCode	String	傳回下一個可用的國碼。列舉所有國碼時，可針對在 <CHostData>.CountryCodes 中所設定的國碼個數依序呼叫此方法。

COM 裝置屬性

下表定義讓您自訂數據機與其他 COM 連線之通訊埠設定值的屬性與參數。

在表 2-23 中，以實際值取代角括號中的資訊。

表 2-23 COM 裝置屬性與參數

屬性	參數	描述
<CHostData>.ComParity(String)	String	設定通訊同位檢查 以下為有效的值： ■ <無> ■ 奇 ■ 偶 ■ 標記 ■ 空格

屬性	參數	描述
<CHostData>.ComFlowControl(String)	String	設定 COM 連線類型的流量控制 以下為有效的值： ■ <無> ■ XONXOFF ■ RTS/CTS ■ 二者
<CHostData>.ComStartedBy(String)	String	設定 COM 連線類型的起始設定 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI) ■ 接收兩次 <CR> 訊號 ■ 數據機回應
<CHostData>.ComEndedBy(String)	String	設定 COM 連線類型的結束設定 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI)
<CHostData>.ComSpeed(Long)	Long	設定最大 COM 速度設定值 以下為有效的值： ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

下表說明 COM 裝置的屬性與傳回值。

在表 2-24 中，以實際值取代角括號中的資訊。

表 2-24 COM 裝置屬性與傳回值

屬性	傳回值	描述
String = <CHostData>.ComParity	String	傳回以下任一值當做通訊同位檢查： ■ <無> ■ 奇 ■ 偶 ■ 標記 ■ 空格
String = <CHostData>.ComFlowControl	String	傳回被控端物件的 Com Flow 設定值 以下為有效的值： ■ <無> ■ XONXOFF ■ RTS/CTS ■ 二者
String = <CHostData>.ComStartedBy	String	傳回 COM 連線類型的 Com Start 控制 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI) ■ 接收兩次 <CR> 訊號 ■ 數據機回應
String = <CHostData>.ComEndedBy	String	傳回 COM 連線類型的 Com End 控制 以下為有效的值： ■ 保持連接 ■ 載波偵測 (DCD) ■ 清除以便傳送 (CTS) ■ 資料組已準備 (DSR) ■ 鈴聲指示器 (RI)

屬性	傳回值	描述
Long = <CHostData>.ComSpeed	Long	傳回被控端物件 Com Speed 目前的設定值 以下為有效的值： ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

NetBIOS 裝置屬性

下表定義 NetBIOS 網路裝置的屬性與參數。

在表 2-25 中，以實際值取代角括號中的資訊。

表 2-25 NetBIOS 屬性與參數

屬性	參數	描述
<CHostData>.LanaNumber(Integer)	Integer	設定 NetBIOS 連線的 LAN Adapter (LANA) 編號

下表定義 NetBIOS 網路裝置的屬性與回傳值。

在表 2-26 中，以實際值取代角括號中的資訊。

表 2-26 NetBIOS 回傳值

屬性	回傳值	描述
Integer = <CHostData>.LanaNumber	Integer	傳回 NetBIOS 連線 LAN Adapter (LANA) 編號目前的設定值

透過 CAPI 2.0 裝置的 ISDN 屬性

下表定義歐洲 ISDN 連線的屬性與參數。

在表 2-27 中，以實際值取代角括號中的資訊。

表 2-27ISDN 透過 CAPI 2.0 屬性與參數

屬性	參數	描述
<CHostData>.CapiChannelBonding(Bool)	Bool	啟用或停用 ISDN CAPI 裝置的通道連接
<CHostData>.CapiExtensions(String)	String	設定通訊所需的其他 CAPI 延伸功能

表 2-28 定義歐洲 ISDN 連線的屬性與傳回值。

表 2-28透過 CAPI 2.0 的 ISDN 屬性與傳回值

屬性	傳回值	描述
Bool= <CHostData>.CapiChannelBonding	Bool	傳回被控端物件中目前 ISDN CAPI 通道連接的設定值
String = <CHostData>.CapiExtensions	String	傳回被控端物件目前的 CAPI 延伸功能清單

CHostDataEx 物件

CHostDataEx 物件所含功能與 CHostData 物件相同，但多了一些額外的功能。

下表說明屬性與參數。

在表 2-29 中，以實際值取代角括號中的資訊。

表 2-29CHostDataEx 屬性與參數

屬性	參數	描述
<CHostData>.CryptPrivateKey(String)	String	設定要使用之私密金鑰容器的名稱。
<CHostData>.CryptCommonName(String)	String	設定要使用之私密金鑰的共用名稱。
<CHostData>.CryptReqLevel(Byte)	Byte	設定加密層次。 以下為有效的值： ■ -1: 無 ■ 0: pcAnywhere ■ 1: 對稱性 ■ 2: 公開金鑰
<CHostData>.CryptRefuseLower(Bool)	Bool	定義被控端電腦是否接受主控端電腦以較低的加密層次進行連線。
<CHostData>.CallersPath(String)	String	設定呼叫者檔案的完整路徑。
<CHostData>.ConfirmConnect(Bool)	Bool	定義系統是否會提示被控端使用者確認連線。

屬性	參數	描述
<CHostData>.ConfirmTimeout(Byte)	Byte	設定系統提示使用者確認連線時間限制的訊息經過幾秒鐘之後才算到期。
<CHostData>.ConfirmDeny(Bool)	Bool	定義提示使用者確認連線的時間線制超過後，是否要結束連線。
<CHostData>.PwCaseSensitive(Bool)	Bool	強迫使用區分大小寫的密碼。
<CHostData>.PwAttempts(Byte)	Byte	設定連續登入失敗多少次之後才結束連線。
<CHostData>.PwTimeout(Byte)	Byte	設定使用者必須完成登入的分鐘數，才可以中斷連線。
<CHostData>.ActiveKbds(Byte)	Byte	定義連線過程中可以使用哪個滑鼠與鍵盤。 以下為有效的值： ■ 0: 被控端和主控端 ■ 1: 被控端 ■ 2: 主控端
<CHostData>.InactiveTimeout(Byte)	Byte	設定連線結束之前，鍵盤和滑鼠可以停用多少分鐘。
<CHostData>.LockSystemWhileWait(Bool)	Bool	設定啟動時鎖定被控端電腦的設定值。
<CHostData>.MinimizeOnLaunch(Bool)	Bool	設定被控端啟動時以最小化執行的選項。
<CHostData>.RunAsService(Bool)	Bool	啟動被控端當作服務執行。
<CHostData>.ConnLostWait(Byte)	Byte	設定等待多少分鐘之後，才允許另一個呼叫者連線。
<CHostData>.ConnLostHostOpts(Bool)	Bool	定義階段作業異常終止時，要等待另一個連線，或取消此被控端的連線。 以下為有效的值： ■ FALSE：等待 ■ TRUE：取消被控端
<CHostData>.EnableConnLostSecurity(Bool)	Bool	啟用或停用異常結束階段作業的結束階段作業安全選項。

屬性	參數	描述
<CHostData>.AuthenticationType(Byte)	Byte	設定驗證類型。 以下為有效的值： ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS Active Directory Services ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID 若未設定任何驗證值，或設定值無效(例如，沒有可用的驗證類型)，則預設使用 pcAnywhere 驗證。
<CHostData>.ConnLostSecurity(Byte)	Byte	設定處理階段作業異常結束的安全選項。 以下為有效的值： ■ 1: 登出使用者 ■ 2: 重新啟動被控端電腦 ■ 3: 鎖定電腦
<CHostData>.CallbkDelay(Byte)	Byte	設定等待多少秒之後，被控端數據機才回撥給主控端。
<CHostData>.EndSessHostOpts(Bool)	Bool	定義階段作業正常結束後，被控端要等待另一個連線，或逕行取消連線。 以下為有效的值： ■ FALSE：等待下一個連線 ■ TRUE：取消被控端
<CHostData>.EnableEndSessSecurity(Bool)	Bool	啟用或停用正常結束階段作業的安全選項。

屬性	參數	描述
<CHostData>.EndSessSecurity(Byte)	Byte	設定正常結束階段作業的安全選項。 以下為有效的值： ■ 1: 登出使用者 ■ 2: 重新啟動被控端電腦 ■ 3: 鎖定電腦
<CHostData>.BlankHost(Bool)	Bool	設定被控端電腦的螢幕清除動作。某些視訊卡並不支援這個選項。
<CHostData>.AllowRemoteMouse(Bool)	Bool	在階段作業期間，停用主控端電腦上的滑鼠。
<CHostData>.RebootOnDisconnect(Bool)	Bool	若此設定為 TRUE，則任何階段作業結束後電腦便會重新啟動。
<CHostData>.PasswordAfterDisc(Bool)	Bool	階段作業結束後將使用者登出。
<CHostData>.LogFailures(Bool)	Bool	記錄密碼失敗的次數。
<CHostData>.AllowDriveSecurity(Bool)	Bool	啟動磁碟機的安全選項。此設定僅適用於 NTFS 檔案系統。
<CHostData>.UseDirectoryServices(Bool)	Bool	啟動使用目錄服務進行驗證。
<CHostData>.DirectoryServiceEntry(String)	String	設定目錄服務設定值。

下表定義 CHostDataEx 物件的屬性與傳回值。

在表 2-30 中，以實際值取代角括號中的資訊。

表 2-30 CHostDataEx 屬性與傳回值

屬性	回傳值	描述
String = <CHostDataEx>.CryptPrivateKey	String	傳回目前使用中的私密金鑰容器。
String = <CHostDataEx>.CryptCommonName	String	傳回使用中私密金鑰容器的共用名稱。
Byte = <CHostDataEx>.CryptReqLevel	Byte	傳回以下任一種加密設定： ■ -1: 無 ■ 0: pcAnywhere ■ 1: 對稱性 ■ 2: 公開金鑰
Bool = <CHostDataEx>.CryptRefuseLower	Bool	傳回拒絕較低層次加密設定的值。
String = <CHostDataEx>.CallersPath	String	傳回呼叫者檔案目前使用中的路徑。

屬性	回傳值	描述
Bool = <CHostDataEx>.ConfirmConnect	Bool	傳回確認連線設定的值。
Byte = <CHostDataEx>.ConfirmTimeout	Byte	傳回拒絕較低層次加密設定的值。
Bool = <CHostDataEx>.ConfirmDeny	Bool	傳回逾時是否切斷連線的設定值。
Bool = <CHostDataEx>.PwCaseSensitive	Bool	傳回要求區分密碼大小寫的設定值。
Byte = <CHostDataEx>.PwAttempts	Byte	傳回限制每次呼叫登入次數的設定值。
Byte = <CHostDataEx>.PwTimeout	Byte	傳回限制完成登入時間的設定值 (以分鐘為單位)。
Byte = <CHostDataEx>.ActiveKbds	Byte	傳回使用中的鍵盤與滑鼠設定值。 以下為有效的值： ■ 0: 被控端和主控端 ■ 1: 被控端 ■ 2: 主控端
Byte = <CHostDataEx>.InactiveTimeout	Byte	傳回無動作時間限制到了之後，再等待多少分鐘才切斷連線。
Bool = <CHostDataEx>.LockSystemWhileWait	Bool	傳回啟動時鎖定被控端電腦的設定值。
Bool = <CHostDataEx>.MinimizeOnLaunch	Bool	傳回被控端啟動時以最小化執行的設定值。
Bool = <CHostDataEx>.RunAsService	Bool	傳回被控端當作服務執行的設定值。
Byte = <CHostDataEx>.ConnLostWait	Byte	傳回等待多少分鐘之後才允許另一個連線。
Bool = <CHostDataEx>.ConnLostHostOpts	Bool	傳回階段作業異常結束時，要等待另一個連線或逕行取消被控端連線。 以下為有效的值： ■ FALSE：等待 ■ TRUE：取消被控端
Bool = <CHostDataEx>.ConnLostWait	Bool	傳回處理異常結束階段作業的安全選項設定值。

屬性	回傳值	描述
Byte = <CHostDataEx>.AuthenticationType	Byte	傳回驗證類型的號碼參照。 以下為有效的值： ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS Active Directory Services ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID 若未設定任何驗證值，或設定值無效(例如，沒有可用的驗證類型)，則預設使用 pcAnywhere 驗證。
Byte = <CHostDataEx>.ConnLostSecurity	Byte	傳回處理異常結束階段作業的安全層次數字代碼。 以下為有效的值： ■ 1: 登出使用者 ■ 2: 重新啟動被控端電腦 ■ 3: 鎖定電腦
Byte = <CHostDataEx>.CallbkDelay	Byte	傳回等待多少秒之後，被控端數據機才回撥給主控端。
Bool = <CHostDataEx>.EndSessHostOpts	Bool	傳回正常結束階段作業後，被控端要等待另一個連線或逕行取消連線。 以下為有效的值： ■ FALSE：等待下一個連線 ■ TRUE：取消被控端
Bool = <CHostDataEx>.EnableEndSessSecurity	Bool	傳回正常結束階段作業後，是否啟動結束階段作業安全選項。

屬性	回傳值	描述
Byte = <CHostDataEx>.EndSessSecurity	Byte	傳回為正常結束階段作業所設定的安全選項。 以下為有效的值： ■ 1: 登出使用者 ■ 2: 重新啟動被控端電腦 ■ 3: 鎖定電腦
Bool = <CHostDataEx>.BlankHost	Bool	傳回所設定的螢幕清除設定選項。
Bool = <CHostDataEx>.AllowRemoteMouse	Bool	傳回主控端使用者在階段作業過程中，是否可以使用鍵盤與滑鼠。
Bool = <CHostDataEx>.RebootOnDisconnect	Bool	傳回階段作業結束後，被控端電腦是否會重新啟動。
Bool = <CHostDataEx>.PasswordAfterDisc	Bool	傳回階段作業結束後，是否要將使用者登出。
Bool = <CHostDataEx>.LogFailures	Bool	傳回是否有設定記錄密碼失敗的選項。
Bool = <CHostDataEx>.AllowDriveSecurity	Bool	傳回是否已經啟動磁碟機安全選項。
Bool = <CHostDataEx>.UseDirectoryServices	Bool	傳回是否已經啟動目錄服務選項。
String = <CHostDataEx>.DirectoryServiceEntry	String	傳回目錄服務設定值。

被控端功能的 Visual Basic 程式碼範例

以下 Visual Basic 程式碼範例會擷取一個被控端資料物件並修改其屬性：

```
Private Sub Command1_Click()  
Dim HostDataManager as Object  
Dim HostData as Object  
Dim s as string  
  
'建立 CHostDataManager 物件  
Set HostDataManager = CreateObject(WINAWSVR.BEHOSTDATAMANAGER)  
  
'顯示和變更目前目錄  
s = HostDataManager.CurrentDirectory()  
MsgBox ( s )  
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")  
s = HostDataManager.CurrentDirectory()  
MsgBox ( s )
```

```
'擷取被控端資料物件
Set HostData = HostDataManager.RetrieveObject("pod.BHF", 2, 0)

'顯示某些屬性
s = HostData.AreaCode()
MsgBox (s)
s = HostData.PhoneNumber()
MsgBox (s)
'設定某些屬性
RemoteData.HostData = "212"
RemoteData.HostData = "555-5555"

'將物件寫入至磁碟
HostData.WriteObject (0)
End Sub
```

使用 **FindFirst** 及 **FindNext** 方法可顯示目錄中的被控端檔案，如下所示：

```
Private Sub Command5_Click()

Dim HostDataManager as Object
Dim HostData as Object
Dim s as string

Set HostDataManager = CreateObject("WINAWSVR.BEHOSTDATAMANAGER")
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
HostDataManager.FindFirst "*", s
MsgBox (s)
HostDataManager.FindNext s
MsgBox (s)

End Sub
```

Awrem32 函數

Awrem32 函數提供處理被控端與主控端電腦之間連線的參數及回傳值。

Awrem32 函數	參數或傳回值	描述
awConnect(FileName)	參數 = 名稱字串	含有關於被控端電腦資訊的完整 .chf 檔案名稱
awConnect	傳回值 = Boolean	如果已執行指令，則為 TRUE
awDisconnect()	傳回值 = Boolean	呼叫此函數後，呼叫的程式必須刪除物件 (C++ - delete IAwrem32X*、VB - ObjectName = Nothing;)
FileXferFromHost(HostFile)	參數 = 字串形式的 HostFile	含有要從被控端電腦複製的完整路徑及檔案名稱
FileXferFromHost(RemoteFile)	參數 = 字串形式的 RemoteFile	含有完整的目的地路徑及檔案名稱。HostFile 及 RemoteFile 字串不需完全相同
FileXferFromHost	傳回值 = Boolean	如果已執行指令，則為 TRUE
FileXferToHost(HostFile)	參數 = 字串形式的 HostFile	從主控端電腦將檔案複製到被控端電腦。參數可包含萬用字元 參數含有完整目的地路徑和檔案名稱
FileXferToHost(RemoteFile)	參數 = 字串形式的 RemoteFile	含有要從主控端電腦複製的完整路徑及檔案名稱。HostFile 及 RemoteFile 字串不需完全相同
CreateFolderOnHost(FolderName)	Parmeter = 字串形式的 FolderName	含有磁碟機及路徑，以便在被控端電腦上建立資料夾
CreateFolderOnHost	傳回值 = Boolean	如果已執行指令，則為 TRUE
ExecuteHostFile(FileName)	參數 = 字串形式的 FileName	執行被控端電腦上的現有檔案。此函數只會執行批次檔、指令及執行檔。它不會執行與執行檔相關聯的檔案。例如，如果您執行 .doc 檔案，此函數不會開啟 Microsoft Word。 參數含有被控端電腦檔案的完整路徑
ExecuteHostFile	傳回值 = Boolean	如果已執行指令，則為 TRUE
GetError()	傳回值 = 字串	傳回以 Awrem32 產生的最後一個錯誤

Awrem32 函數	參數或傳回值	描述
ConnectionStatus()	傳回值 = Short	將目前的連線狀態傳回被控端電腦 可能的值包括： ■ -1 = 失去連線 ■ 0 = 無連線 ■ 1 = 階段作業已連線

Visual C++ 物件定義

本章包含以下主題：

- [關於 Visual C++ 物件](#)
- [CRemoteDataManager 方法](#)
- [CRemoteData 物件](#)
- [CRemoteDataEx 物件](#)
- [CHostDataManager 方法](#)
- [CHostData 物件](#)
- [CHostDataEx 物件](#)
- [Awrem32 函數](#)

關於 Visual C++ 物件

「pcAnywhere 自動化伺服器」提供下列支援 OLE 自動化的元件：

Winawsvr	提供連線到「pcAnywhere 自動化伺服器」與存取其介面所需的資訊
Awrem32	提供控制 pcAnywhere 連線所需的資訊

在表 3-1 中說明的物件構成了 Winawsvr。

表 3-1 Winawsvr 物件

物件	描述	參考文件
CRemoteDataManager	提供用來建立、開啟、修改、儲存及刪除 CRemoteData 物件的方法	請參閱第 52 頁的「 CRemoteDataManager 方法 」。
CRemoteData	定義存取與控制 pcAnywhere 主控端功能的參數	請參閱第 54 頁的「 CRemoteData 物件 」。 請參閱第60頁的「 CRemoteDataEx物件 」。
CHostDataManager	提供用來建立、開啟、修改、儲存及刪除 CHostData 物件的方法	請參閱第 62 頁的「 CHostDataManager 方法 」。
CHostData	定義存取與控制 pcAnywhere 被控端功能的參數	請參閱第 54 頁的「 CRemoteData 物件 」。 請參閱第60頁的「 CRemoteDataEx物件 」。

Awrem32 有一個物件是由 8 個介面組成，支援遠端遙控與檔案傳輸階段作業。
請參閱第 74 頁的「[Awrem32 函數](#)」。

某些功能已不再支援。但是，物件定義仍可供給先前的版本使用。

對於要求密碼的功能，可以設定密碼，但無法擷取。這是為了安全起見。

CRemoteDataManager 方法

CRemoteDataManager 方法提供存取與控制 CRemoteData 物件的參數與回傳值。

方法	參數或傳回值	描述
BSTR CurrentDirectory();	參數 = BSTR	儲存 pcAnywhere 主控端物件的目前 pcAnywhere 資料目錄完整路徑名稱
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	參數 = LPCTSTR lpszNewDirectory 傳回值 = BOOL	變更儲存 pcAnywhere 主控端物件的目前目錄。參數是現有目錄的名稱 如果成功，傳回值是 TRUE
BOOL FindFirst(LPCTSTR lpszPattern, BSTR FAR*);	參數 = LPCTSTR lpszPattern	用於過濾物件檔案的檔案名稱樣式 (使用 [*] 會在目前目錄中尋找所有檔案)。
BOOL FindFirst(pbstrFullQualName);	參數 = BSTR FAR * pbstrFullQualName	傳回符合指定樣式之主控端物件檔案完整路徑名稱的緩衝區。
BOOL FindFirst	傳回值 = BOOL	如果找到符合指定樣式的主控端物件，則為 TRUE。相符檔案的完整路徑名稱儲存在 pbstrFullQualName

方法	參數或傳回值	描述
BOOL FindNext(BSTR FAR* pbstrFullQualName);	參數 = BSTR FAR* pbstrFullQualName 傳回值 = BOOL	成功呼叫 FindFirst() 以取得目前目錄中的主控端物件檔案名稱之後，就可以呼叫 FindNext() 以尋找符合樣式的下一個檔案 (若有的話) 如果找到另一個符合在 FindFirst() 呼叫中指定樣式的主控端物件檔案，則傳回值是 TRUE。相符檔案的完整路徑名稱儲存在 pbstrFullQualName
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName	按照檔案名稱擷取 CRemoteData 物件 參數是要載入的完整主控端物件檔案名稱
LPDISPATCH RetrieveObject(short wAccessMode);	參數 = short wAccessMode	指定要如何使用此物件。這與密碼保護有關。選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword	物件密碼。可以是空值
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName	按照檔案名稱擷取 CRemoteDataEx 物件 參數是要載入的完整主控端物件檔案名稱
LPDISPATCH RetrieveObjectEx(short wAccessMode);	參數 = short wAccessMode	指定要如何使用此物件。這與密碼保護有關。選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword	物件密碼。可以是空值
LPDISPATCH RetrieveObjectEx	傳回值 = LPDISPATCH	指向 OLE 分派物件的指標。物件是 CRemoteDataEx 物件。 請參閱 主控端功能的 Visual C++ 程式碼範例 。

方法	參數或傳回值	描述
LPDISPATCH CreateObject(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName 傳回值 = LPDISPATCH	建立 CRemoteData 物件，並將 LPDISPATC 指標傳回。 參數是新物件的完整主控端物件檔案名稱 傳回值是指向 OLE 分派物件的指標。該 物件是 CRemoteData 物件。 請參閱 主控端功能的 Visual C++ 程式碼 範例 。
LPDISPATCH CreateObjectEx(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName 傳回值 = LPDISPATCH	建立 CRemoteDataEx 物件，並將 LPDISPATCH 指標傳回 參數是新物件的完整主控端物件檔案名稱 傳回值是指向 OLE 分派物件的指標。物 件是 CRemoteDataEx 物件。 請參閱 主控端功能的 Visual C++ 程式碼 範例 。
BOOL DeleteObject(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName	刪除主控端物件檔案。 參數是所需刪除物件的完整主控端物件檔 案名稱
BOOL DeleteObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword	物件密碼。可以是空值
BOOL DeleteObject	BOOL	如果刪除了物件，則為 TRUE。
BOOL Launch(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName 傳回值 = BOOL	啟動開啟 pcAnywhere 主控端終端機視窗 的主控端物件檔案。 參數是所需啟動物件的完整主控端物件檔 案 如果成功啟動物件，則傳回值是 TRUE

CRemoteData 物件

使用此物件可修改主控端物件資料。

Get 及 Set 方法

下列方法可用來取得及設定 CRemoteData 物件的屬性。

電腦名稱是啟動主控端物件時所要呼叫之 pcAnywhere 被控端電腦的名稱。

```
BSTR GetComputerName();  
void SetComputerName(LPCTSTR lpszNewValue);
```

電話號碼是用來與 **pcAnywhere** 被控端電腦建立數據機連線時的撥接號碼。

```
BSTR GetPhoneNumber();  
void SetPhoneNumber(LPCTSTR lpszNewValue);
```

指出要使用 **TAPI** 撥號屬性 (位置資訊) (**TRUE**)，還是要完全按照所出現的電話號碼字串來撥號 (**FALSE**)。

```
BOOL GetUseDialingProperties();  
void SetUseDialingProperties(BOOL bNewValue);
```

如果要使用撥號屬性，這個就是所要呼叫之號碼的區碼。

```
BSTR GetAreaCode();  
void SetAreaCode(LPCTSTR lpszNewValue);
```

如果要使用撥號屬性，這個就是所要呼叫之號碼的國碼。

```
BSTR GetCountryCode();  
void SetCountryCode(LPCTSTR lpszNewValue);
```

如果呼叫失敗，要重試撥接此號碼的次數。

```
short GetRedialCount();  
void SetRedialCount(short nNewValue);
```

每次嘗試重撥之間的等待時間 (以秒計)。

```
short GetRedialDelay();  
void SetRedialDelay(short nNewValue);
```

建立好連線時，要傳送給被控端的登入名稱。如果左邊是空的，就會提示使用者輸入連線的登入名稱。

```
BSTR GetAutoLoginName();  
void SetAutoLoginName(LPCTSTR lpszNewValue);
```

建立好連線時，要傳送給被控端的登入密碼。如果左邊是空的，就會提示使用者輸入連線的密碼。

```
BSTR GetAutoLoginPassword();  
void SetAutoLoginPassword(LPCTSTR lpszNewValue);
```

此物件的密碼。

```
BSTR GetPassword();  
void SetPassword(LPCTSTR lpszNewValue);
```

只有在有使用密碼的情況下，才能啟動此物件 (TRUE)。

```
BOOL GetExecuteProtection();  
void SetExecuteProtection(BOOL bNewValue);
```

只有在有提供正確密碼的情況下，才能檢視此物件 (TRUE)。

```
BOOL GetReadProtection();  
void SetReadProtection(BOOL bNewValue);
```

只有在有提供正確密碼的情況下，才能寫入此物件 (TRUE)。

```
BOOL GetWriteProtection();  
void SetWriteProtection(BOOL bNewValue);
```

控制是否要記錄使用此物件的階段作業。

```
BOOL GetLogSession();  
void SetLogSession(BOOL bNewValue);
```

控制是否要從頭開始記錄使用此物件的階段作業。

```
BOOL GetRecordSession();  
void SetRecordSession(BOOL bNewValue);
```

使用此物件之階段作業的記錄檔名稱。

```
BSTR GetRecordFile();  
void SetRecordFile(LPCTSTR lpszNewValue);
```

設定連線類型。

```
BOOL GetRunOnConnect();  
void SetRunOnConnect(BOOL bNewValue);
```

可用的連線類型如下：

- COM1
- COM2
- COM3
- COM4
- ISDN 透過 CAPI 2.0
- NetBIOS

- SPX
- DEFAULT TAPI
- TCP/IP

TAPI 裝置的名稱也可以作為連線類型。DEFAULT TAPI 會使用在系統上所找到的第一個 TAPI 裝置。若要使用特定的 TAPI 裝置，請使用 `FirstConnectionType()` / `NextConnectionType()` 以搜尋可用的裝置。

主控端物件細節方法

指派好主控端物件的連線類型時，便會將裝置細節設定為有效的預設值。

下列連線類型具有可在應用程式中設定的進階架構選項：

- COM 裝置
- NetBIOS 裝置
- 透過 CAPI 2.0 (僅適用於歐洲的 ISDN) 裝置的 ISDN

COM 裝置細節

設定通訊同位檢查層次。

```
BSTR GetComParity();  
void SetComParity(LPCTSTR lpszNewValue);
```

通訊同位檢查值如下：

- 無
- 奇
- 偶
- 標記
- 空格

設定流量控制層次。

```
BSTR GetComFlowControl();  
void SetComFlowControl(LPCTSTR lpszNewValue);
```

流量控制值如下：

- <無>
- XONXOFF
- RTS/CTS

■ 二者

設定連線起始設定。

```
BSTR GetComStartedBy();  
void SetComStartedBy(LPCTSTR lpszNewValue);
```

連線起始值如下：

- 保持連接
- 載波偵測 (DCD)
- 清除以便傳送 (CTS)
- 資料組已準備 (DSR)
- 鈴聲指示器 (RI)
- 接收兩次 <CR> 訊號
- 數據機回應

設定連線結束值。

```
BSTR GetComEndedBy();  
void SetComEndedBy(LPCTSTR lpszNewValue);
```

連線結束值如下：

- 保持連接
- 載波偵測 (DCD)
- 清除以便傳送 (CTS)
- 資料組已準備 (DSR)
- 鈴聲指示器 (RI)

設定連線速度。

```
long GetComSpeed();  
void SetComSpeed(long nNewValue);
```

連線速度值如下：

- 110
- 300
- 600
- 1200
- 2400

- 4800
- 9600
- 19200
- 38400
- 57600
- 115200

NetBIOS 裝置細節

設定要用在此連線的 LAN Adapter (LANA) 號碼。

```
short GetLanaNumber();  
void SetLanaNumber(short nNewValue);
```

透過 CAPI 2.0 裝置的 ISDN 細節

如果是 TRUE，啟用通道連結 (在一個連線上使用 2 個 ISDN 通道)。

```
BOOL GetCapiChannelBonding();  
void SetCapiChannelBonding(BOOL bNewValue);
```

設定通訊所需的任何其他 CAPI 延伸功能。

```
BSTR GetCapiExtensions();  
void SetCapiExtensions(LPCTSTR lpszNewValue);
```

主控端物件方法

下表說明主控端物件的正常方法 (用來取得與設定屬性)：

方法	參數或傳回值	描述
short ConnectionTypes();	傳回值 = Short	此電腦上可用之連線類型數量
BSTR FirstConnectionType(); 與 BSTR NextConnectionType();	傳回值 = BSTR	FirstConnectionType() 及 NextConnectionType() 是用來反覆尋找 可用的連線類型。這些函數會傳回 BSTR，這是可用連線類型的名稱。您可 以搭配 SetConnectionType() 函數使用這 些傳回的連線類型。 傳回值是所支援連線裝置類型的名稱

方法	參數或傳回值	描述
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	參數 = LPCTSTR lpszConnectionType 傳回值 = BOOL	連線裝置類型的名稱 如果在此電腦上找到可以使用的連線類型，則傳回值是 TRUE
short CountryCodes();	傳回值 = Short	傳回可用國碼的數量
BSTR FirstCountryCode(); 與 BSTR NextCountryCode();	傳回值 = BSTR	FirstCountryCode() 及 NextCountryCode() 是用來反覆尋找可用的國碼。這些函數會傳回 BSTR，也就是可用國碼的名稱。您可以搭配 SetCountryCode() 函數使用這些傳回的值 傳回值是第一個或第二個國碼字串
BOOL ReadObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword 傳回值 = BOOL	從主控端物件檔案讀取物件資料 參數是物件密碼 如果成功讀取物件，則傳回值是 TRUE
BOOL WriteObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword 傳回值 = BOOL	將物件資料寫出主控端物件檔案。 參數是物件密碼 如果成功寫出物件，則傳回值是 TRUE

CRemoteDataEx 物件

CRemoteDataEx 物件含有與 CRemoteData 物件相同的功能，還有下列額外的 Get 與 Set 方法：

```
BSTR GetPrivateKey(); //傳回 PrivateKey 資訊
void SetPrivateKey(LPCTSTR lpszNewValue);
BSTR GetCertificationName(); //傳回認證名稱
void SetCertificationName(LPCTSTR lpszNewValue);
short GetEncryptionLevel(); //傳回加密層次值
void SetEncryptionLevel(short nNewValue);
BOOL GetDenyLowerEncrypt(); //傳回 DenyLowerEncrypt 值
void SetDenyLowerEncrypt(BOOL bNewValue);
BSTR GetAutoDomain(); //傳回 AutoDomain 值
void SetAutoDomain(LPCTSTR lpszNewValue);
```

主控端功能的 Visual C++ 程式碼範例

下列的 C++ 函數範例會建立一個主控端物件，將其連線類型設定為 TCP/IP，將電腦名稱設定為傳入函數中的 TCP/IP 位址，然後啟動主控端物件：

```
    BOOL LaunchTCPRemote(LPCTSTR lpszAddress)
    {
        BOOL bReturn = FALSE;

        CRemoteDataManager remoteDM;

        CRemoteData remoteData;

        // 首先，建立 CRemoteDataManager
        remoteDM.CreateDispatch( _T( "WINAWSVR.RemoteDataManager" ) );
        // 然後，建立 CRemoteData 並加以連接
        remoteData.AttachDispatch( remoteDM.CreateObject("Test", 0) );

        // 此時，設定所需的屬性
        remoteData.SetConnectionType("TCP/IP");

        remoteData.SetComputerName(lpszAddress);

        // 儲存物件資料
        if (remoteData.WriteObject(0))
        {
            // 並加以啟動
            if (remoteData.Launch())
                bReturn = TRUE;
        }
        // 釋放主控端物件。
        remoteData.ReleaseDispatch();

        remoteDM.ReleaseDispatch( _T( "WINAWSVR.RemoteDataManager" ) );

        return bReturn;
    }
```

CHostDataManager 方法

下表說明 CHostDataManager 方法，包括存取與控制 CHostData 物件的參數與傳回值：

方法	參數或傳回值	描述
BSTR CurrentDirectory();	傳回值 = BSTR	傳回儲存 pcAnywhere 被控端物件的目前目錄完整路徑名稱
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	參數 = LPCTSTR lpszNewDirectory 傳回值 = BOOL	變更儲存 pcAnywhere 被控端物件的目前目錄 參數是現有目錄的名稱 如果成功，傳回值是 TRUE
BOOL FindFirst(LPCTSTR lpszPattern);	參數 = LPCTSTR lpszPattern 傳回值 = BOOL	根據指定的檔案名稱樣式，尋找目前目錄中的第一個 pcAnywhere 被控端物件檔案 (*.bhf) 參數是用於過濾物件檔案的檔案名稱樣式 (使用 [*] 會在目前目錄中尋找所有被控端檔案) 如果找到符合指定樣式的被控端物件，則傳回值是 TRUE。相符檔案的完整路徑名稱儲存在 pbstrFullQualName
BOOL FindFirst(BSTR FAR* pbstrFullQualName);	參數 = BSTR FAR * pbstrFullQualName	傳回符合指定樣式之主控端物件檔案完整路徑名稱的緩衝區。 如果找到符合指定樣式的被控端物件，則傳回值是 TRUE。相符檔案的完整路徑名稱存放在 pbstrFullQualName。
BOOL FindNext(BSTR FAR* pbstrFullQualName);	參數 = BSTR FAR * pbstrFullQualName 傳回值 = BOOL	成功呼叫 FindFirst() 以取得目前目錄中的被控端物件檔案名稱之後，就可以呼叫 FindNext() 以尋找符合樣式的下一個檔案 (若有的話) 參數是符合 FindFirst() 原始呼叫中指定樣式之被控端物件檔案完整路徑名稱的傳回緩衝區 如果找到另一個符合在 FindFirst() 呼叫中指定樣式的被控端物件檔案，則傳回值是 TRUE。相符檔案的完整路徑名稱儲存在 pbstrFullQualName

方法	參數或傳回值	描述
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName	按照檔案名稱擷取 CHostData 物件。 參數是要載入的完整被控端物件檔案名稱
LPDISPATCH RetrieveObject(short wAccessMode);	參數 = short wAccessMode	指定要如何使用此物件。這與密碼保護有關。選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword	物件密碼。可以是空值
LPDISPATCH RetrieveObject	傳回值 = LPDISPATCH	指向 OLE 分派物件的指標。該物件是 CHostData 物件。 請參閱 被控端功能 Visual C++ 的程式碼範例 。
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName	按照檔案名稱擷取 CHostDataEx 物件 參數是要載入的完整被控端物件檔案名稱
LPDISPATCH RetrieveObjectEx(short wAccessMode);	參數 = short wAccessMode	指定要如何使用此物件。這與密碼保護有關。選項如下： ■ 0 = 不指定 ■ 1 = 只檢視 ■ 2 = 檢視及修改 ■ 3 = 執行
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword	物件密碼。可以是空值
LPDISPATCH RetrieveObjectEx	傳回值 = LPDISPATCH	指向 OLE 分派物件的指標。此物件是 CHostDataEx 物件 請參閱 被控端功能 Visual C++ 的程式碼範例 。

方法	參數或傳回值	描述
LPDISPATCH CreateObject(LPCTSTR lpszName);	參數 = LPCTSTR lpszFQName 傳回值 = LPDISPATCH	建立 CHostData 物件，並將 LPDISPATCH 指標傳回 參數是新物件的完整被控端物件檔案名稱 傳回值是指向 OLE 分派物件的指標。此物件是 CHostData 物件 請參閱 被控端功能 Visual C++ 的程式碼範例 。
LPDISPATCH CreateObjectEx(LPCTSTR lpszName);	參數 = LPCTSTR lpszFQName 傳回值 = LPDISPATCH	建立 CHostDataEx 物件，並將 LPDISPATCH 指標傳回 參數是新物件的完整被控端物件檔案名稱 傳回值是指向 OLE 分派物件的指標。此物件是 CHostDataEx 物件。 請參閱 被控端功能 Visual C++ 的程式碼範例 。
BOOL DeleteObject(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName	刪除被控端物件檔案 參數是所需刪除物件的完整被控端物件檔案名稱
BOOL DeleteObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword	物件密碼。可以是空值
BOOL DeleteObject	傳回值 = BOOL	如果刪除了物件，則為 TRUE。
BOOL Launch(LPCTSTR lpszFQName);	參數 = LPCTSTR lpszFQName 傳回值 = BOOL	啟動開啟 pcAnywhere 被控端終端機視窗的被控端物件檔案 參數是所需啟動物件的完整被控端物件檔案 如果成功啟動物件，則傳回值是 TRUE

CHostData 物件

使用此物件以修改被控端物件資料。

Get 及 Set 方法

下列方法可用來取得及設定 CHostData 物件的屬性。

電話號碼是用來與 pcAnywhere 主控端電腦建立數據機連線的撥接號碼。

```
BSTR GetPhoneNumber();  
void SetPhoneNumber(LPCTSTR lpszNewValue);
```

指出要使用 TAPI 撥號屬性 (位置資訊) (TRUE)，還是要完全按照所出現的電話號碼字串來撥號 (FALSE)。

```
BOOL GetUseDialingProperties();  
void SetUseDialingProperties(BOOL bNewValue);
```

如果要使用撥號屬性，這個就是所要呼叫之號碼的區碼。

```
BSTR GetAreaCode();  
void SetAreaCode(LPCTSTR lpszNewValue);
```

如果要使用撥號屬性，這個就是所要呼叫之號碼的國碼。

```
BSTR GetCountryCode();  
void SetCountryCode(LPCTSTR lpszNewValue);
```

如果呼叫失敗，要重試撥接此號碼的次數。

```
short GetRedialCount();  
void SetRedialCount(short nNewValue);
```

每次嘗試重撥之間的等待時間 (以秒計)。

```
short GetRedialDelay();  
void SetRedialDelay(short nNewValue);
```

控制是否要記錄使用此物件的階段作業。

```
BOOL GetLogSession();  
void SetLogSession(BOOL bNewValue);
```

控制是否要從頭開始記錄使用此物件的階段作業。

```
BOOL GetRecordSession();  
void SetRecordSession(BOOL bNewValue);
```

使用此物件之階段作業的記錄檔名稱。

```
BSTR GetRecordFile();  
void SetRecordFile(LPCTSTR lpszNewValue);
```

被控端物件細節方法

指派好被控端物件的連線類型時，便會將裝置細節設定為有效的預設值。

下列連線類型具有可在應用程式中設定的進階架構選項：

- COM 裝置
- NetBIOS 裝置
- NASI/NCSI 裝置
- 透過 CAPI 2.0 (僅適用於歐洲的 ISDN) 裝置的 ISDN

COM 裝置細節

下列程式碼將所要求的連線類型放在被控端物件的指定連線類型的清單上，並且在處理後續的特定裝置方法呼叫時，使它成為現行的連線類型：

```
BOOL AssignConnection(LPCTSTR lpszNewValue);
```

如果要求的連線類型已經在指定連線的清單上，指定連線的清單將不會變更。只會將目前的連線類型變更為要求的類型。通常在取得和設定連線專有設定值的過程中，都會在相同的物件上呼叫好幾次 `AssignConnection` 方法。

如果電腦上有傳入的連線類型且順利指派或已經指派，則 `AssignConnection` 會傳回 `TRUE`。如果電腦上沒有要求的連線類型，或是目前指派的連線數已達允許的上限，則它會傳回 `FALSE`。

`pcAnywhere` 被控端物件最多可以支援兩種指定連線類型。如果 `AssignConnection` 方法偵測到您嘗試超越此限制，便會傳回 `FALSE`。

可用的連線類型如下：

- COM1
- COM2
- COM3
- COM4
- SPX
- NetBIOS
- TCP/IP
- ISDN 透過 CAPI 2.0
- DEFAULT TAPI

TAPI 裝置的名稱也可以作為連線類型。DEFAULT TAPI 會使用在系統中所找到的第一個 TAPI 裝置。若要使用特定的 TAPI 裝置，請使用 FirstConnectionType() 及 NextConnectionType() 來搜尋可用的裝置。

下列程式碼會取消指派連線類型。在取消指派連線類型之後，其餘已指派的連線(如果有的話)就會變成用於後續裝置專有方法呼叫的目前連線類型。

```
BOOL UnassignConnection(LPCTSTR lpszNewValue);
```

下列程式碼設定通訊同位檢查層次：

```
BSTR GetComParity();  
void SetComParity(LPCTSTR lpszNewValue);
```

通訊同位檢查值如下：

- 無
- 奇
- 偶
- 標記
- 空格

下列程式碼設定流量控制層次：

```
BSTR GetComFlowControl();void SetComFlowControl(LPCTSTR  
lpszNewValue);
```

流量控制值如下：

- <無>
- XONXOFF
- RTS/CTS
- 二者

下列程式碼設定連線起始值：

```
BSTR GetComStartedBy();  
void SetComStartedBy(LPCTSTR lpszNewValue);
```

連線起始值如下：

- 保持連接
- 載波偵測 (DCD)
- 清除以便傳送 (CTS)

- 資料組已準備 (DSR)
- 鈴聲指示器 (RI)
- 接收兩次 <CR> 訊號
- 數據機回應

下列程式碼設定連線結束值：

```
BSTR GetComEndedBy();  
void SetComEndedBy(LPCTSTR lpszNewValue);
```

連線結束值如下：

- 保持連接
- 載波偵測 (DCD)
- 清除以便傳送 (CTS)
- 資料組已準備 (DSR)
- 鈴聲指示器 (RI)

下列程式碼設定連線速度：

```
long GetComSpeed();  
void SetComSpeed(long nNewValue);
```

連線速度值如下：

- 110
- 300
- 600
- 1200
- 2400
- 4800
- 9600
- 19200
- 38400
- 57600
- 115200

NetBIOS 裝置細節

設定要用在此連線的 LAN Adapter (LANA) 號碼。

```
short GetLanaNumber();void SetLanaNumber(short nNewValue);
```

NASI/NCSI 裝置細節

設定 NASI 伺服器的使用者名稱。

```
BSTR GetNasiUserName();  
void SetNasiUserName(LPCTSTR lpszNewValue);
```

設定 NASI 伺服器的使用者密碼。

```
BSTR GetNasiPassword();  
void SetNasiPassword(LPCTSTR lpszNewValue);
```

設定 NASI 階段作業名稱。

```
BSTR GetNasiSessionName();  
void SetNasiSessionName(LPCTSTR lpszNewValue);  
BOOL GetNasiSessionNameAvailable();  
void SetNasiSessionNameAvailable(BOOL bNewValue);
```

指定要使用的 NASI 伺服器。

```
BOOL NasiServer();  
BSTR GetNasiServerName();  
void SetNasiServerName(LPCTSTR lpszNewValue);
```

指定要使用的 NASI 服務。

```
BOOL NasiService();  
BSTR GetNasiServiceName();  
void SetNasiServiceName(LPCTSTR lpszNewValue);
```

指定要使用的 NASI 埠。

```
BOOL NasiPort();  
BSTR GetNasiPortName();  
void SetNasiPortName(LPCTSTR lpszNewValue);  
BOOL GetNasiSelectOnConnect();  
void SetNasiSelectOnConnect(BOOL bNewValue);
```

透過 CAPI 2.0 裝置的 ISDN 細節

如果是 TRUE，啟用通道連結(在一個連線上使用 2 個 ISDN 通道)。

```
    BOOL GetCapiChannelBonding();  
void SetCapiChannelBonding(BOOL bNewValue);
```

設定通訊所需的任何其他 CAPI 延伸功能。

```
    BSTR GetCapiExtensions();  
void SetCapiExtensions(LPCTSTR lpszNewValue);
```

被控端物件方法

下表說明物件的正常方法 (用來取得與設定屬性)：

方法	參數或傳回值	描述
short ConnectionTypes();	傳回值 = Short	傳回此電腦上可用的連線類型數量
BSTR FirstConnectionType(); 與 BSTR NextConnectionType();	傳回值 = BSTR	FirstConnectionType() 及 NextConnectionType() 是用來反覆尋找可 用的連線類型。這些函數會傳回 BSTR，這 是可用連線類型的名稱。您可以搭配 SetConnectionType() 函數使用這些傳回的 連線類型 傳回值是所支援連線裝置類型的名稱
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	參數 = LPCTSTR lpszConnectionTyp 傳回值 = BOOL	參數是連線裝置類型的名稱 如果使用這個裝置類型，則傳回值是 TRUE
short MaxAssignedConnections()	傳回值 = Short	傳回可以同時指派的連線類型數量上限(目 前是兩個)
short AssignedConnections()	傳回值 = Short	傳回所指派之連線類型的個數。
BSTR FirstAssignedConnection(); 與 BSTR NextAssignedConnection();	傳回值 = BSTR	FirstAssignedConnection() 及 NextAssignedConnection() 會在所指派的 連線之間反覆執行。這些函數會傳回 BSTR，這是所指派之連線類型的名稱。您 可以搭配 AssignConnection() 函數使用這 些傳回的連線類型 傳回值是所支援連線裝置類型的名稱
BOOL FindAssignedConnection (LPCTSTR lpszConnectionType);	參數 = LPCTSTR lpszConnectionTyp 傳回值 = BOOL	參數是連線裝置類型的名稱 如果目前指派這個裝置類型，則傳回值是 TRUE

方法	參數或傳回值	描述
short CountryCodes();	傳回值 = Short	傳回可用國碼的數量
BSTR FirstCountryCode(); 與 BSTR NextCountryCode();	傳回值 = BSTR	FirstCountryCode() 與 NextCountryCode() 會在可用的國碼清單之間反覆執行。這些函數會傳回 BSTR，也就是可用國碼的名稱。您可以搭配 SetCountryCode() 函數使用這些傳回的值。 傳回值是第一個或第二個國碼字串
BOOL ReadObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword 傳回值 = BOOL	從被控端物件檔案讀取物件資料 參數是物件密碼 如果成功讀取物件，則傳回值是 TRUE
BOOL WriteObject(LPCTSTR lpszPassword);	參數 = LPCTSTR lpszPassword 傳回值 = BOOL	將物件資料寫出至被控端物件檔案 參數是物件密碼 如果成功寫出物件，則傳回值是 TRUE

CHostDataEx 物件

CHostDataEx 物件含有與 CHostData 物件相同的功能，還有下列額外的 Get 與 Set 方法：

```
BOOL GetReadProtection();  
void SetReadProtection(BOOL bNewValue);  
BOOL GetWriteProtection();  
void SetWriteProtection(BOOL bNewValue);  
BSTR GetPassword(); //Returns "NOT IMPLEMENTED"  
void SetPassword(LPCTSTR lpszNewValue);  
BSTR GetCallersPath();  
void SetCallersPath(LPCTSTR lpszNewValue);  
BOOL GetConfirmConnect();  
void SetConfirmConnect(BOOL bNewValue);  
short GetConfirmTimeout();  
void SetConfirmTimeout(short nNewValue);  
BOOL GetConfirmDeny();  
void SetConfirmDeny(BOOL bNewValue);  
BOOL GetPwCaseSensitive();  
void SetPwCaseSensitive(BOOL bNewValue);  
short GetPwAttempts();  
void SetPwAttempts(short nNewValue);  
short GetPwTimeout();
```

```
void SetPwTimeout(short nNewValue);
short GetActiveKbds();
void SetActiveKbds(short nNewValue); //設定 ActiveKbds
short GetInactiveTimeout();
void SetInactiveTimeout(short nNewValue);
short GetCryptReqLevel();
void SetCryptReqLevel(short nNewValue);
BOOL GetCryptRefuseLower();
void SetCryptRefuseLower(BOOL bNewValue);
short GetAuthenticationType();
void SetAuthenticationType(short nNewValue);
BOOL GetLockSystemWhileWait();
void SetLockSystemWhileWait(BOOL bNewValue);
BOOL GetMinimizeOnLaunch();
void SetMinimizeOnLaunch(BOOL bNewValue);
BOOL GetRunAsService();
void SetRunAsService(BOOL bNewValue);
short GetConnLostWait();
void SetConnLostWait(short nNewValue);
BOOL GetConnLostHostOpts();
void SetConnLostHostOpts(BOOL bNewValue);
BOOL GetEnableConnLostSecurity();
void SetEnableConnLostSecurity(BOOL bNewValue);
short GetConnLostSecurity();
void SetConnLostSecurity(short nNewValue);
short GetCallbkDelay();
void SetCallbkDelay(short nNewValue);
BOOL GetEndSessHostOpts();
void SetEndSessHostOpts(BOOL bNewValue);
BOOL GetEnableEndSessSecurity();
void SetEnableEndSessSecurity(BOOL bNewValue);
short GetEndSessSecurity();
void SetEndSessSecurity(short nNewValue);
BSTR GetCryptPrivateKey();
void SetCryptPrivateKey(LPCTSTR lpszNewValue);
BSTR GetCryptCommonName();
void SetCryptCommonName(LPCTSTR lpszNewValue);
BOOL GetBlankHost();
void SetBlankHost(BOOL bNewValue);
BOOL GetAllowRemoteMouse();
void SetAllowRemoteMouse(BOOL bNewValue);
short GetRebootOnDisconnect();
void SetRebootOnDisconnect(short nNewValue);
```

```
BOOL GetPasswordAfterDisc();  
void SetPasswordAfterDisc(BOOL bNewValue);  
BOOL GetLogFailures();  
void SetLogFailures(BOOL bNewValue);  
BOOL GetAllowDriveSecurity();  
void SetAllowDriveSecurity(BOOL bNewValue);  
BOOL GetExecuteProtection();  
void SetExecuteProtection(BOOL bNewValue);
```

被控端功能 Visual C++ 的程式碼範例

下列的 Visual C++ 函數範例會建立一個被控端物件，將其連接類型設定為 TCP/IP，將電腦名稱設定為傳入函數中的 TCP/IP 位址，然後啟動被控端物件：

```
BOOL LaunchTCPHost(LPCTSTR lpszAddress)  
{  
    BOOL bReturn = FALSE;  
  
    CHostDataManager hostDM;  
    CHostData hostData;  
  
    // 首先，建立 CHostDataManager  
    hostDM.CreateDispatch( _T( "WINAHSV.BEHostDataManager" ) );  
  
    // 然後，建立 CRemoteData 並加以連接  
    hostData.AttachDispatch( hostDM.CreateObject( "Test", 0 ) );  
  
    // 此時，設定所需的屬性  
    hostData.SetConnectionType( "TCP/IP" );  
  
    // 儲存物件資料  
    if ( hostData.WriteObject( 0 ) )  
    {  
        // 並加以啟動  
        if ( hostData.Launch() )  
            bReturn = TRUE;  
    }  
}
```

```
// 釋放 Host 物件。  
hostData.ReleaseDispatch();  
  
return (bReturn);  
}
```

Awrem32 函數

下表說明 Awrem32 函數，包括處理被控端與主控端電腦之間連線的參數及傳回值。

方法	參數或傳回值	描述
boolean awConnect(BSTR FileName);	參數 = 名稱字串 傳回值 = Boolean	建立與被控端電腦的連線 參數含有關於被控端電腦資訊的完整 .chf 檔案名稱 傳回值會執行指令
boolean awDisconnect();	傳回值 = Boolean	與被控端電腦中斷連線 呼叫此函數之後，呼叫程式必須刪除物 件 (C++ - 刪除 IAwrem32X*、VB - ObjectName = Nothing;)
boolean FileXferFromHost(BSTR HostFile);	參數 = 字串形式的 HostFile	從被控端電腦將檔案複製到主控端電腦。 參數可包含萬用字元 參數含有要從被控端電腦複製的完整路 徑及檔案名稱。HostFile 及 RemoteFile 字串不需完全相同
boolean FileXferFromHost(BSTR RemoteFile);	參數 = 字串形式的 RemoteFile	從被控端電腦將檔案複製到主控端電腦。 參數可包含萬用字元 參數含有完整目的地路徑和檔案名稱。 HostFile 及 RemoteFile 字串不需完全相 同。
boolean FileXferFromHost	傳回值 = Boolean	如果已執行指令，則為 TRUE
boolean FileXferToHost(BSTR HostFile);	參數 = 字串形式的 HostFile	從主控端電腦將檔案複製到被控端電腦。 參數可包含萬用字元 參數含有完整目的地路徑和檔案名稱。 HostFile 及 RemoteFile 字串不需完全相 同

方法	參數或傳回值	描述
boolean FileXferToHost(BSTR RemoteFile);	參數 = 字串形式的 RemoteFile 傳回值 = Boolean	從主控端電腦將檔案複製到被控端電腦。 參數可包含萬用字元 參數含有要從主控端電腦複製的完整路 徑及檔案名稱。HostFile 及 RemoteFile 字串不需完全相同
boolean FileXferToHost	傳回值 = Boolean	如果已執行指令，則為 TRUE
boolean CreateFolderOnHost(BSTR FolderName);	參數 = 字串形式的 FolderName 傳回值 = Boolean	在被控端電腦上建立新資料夾。此函數 會在主控端電腦上建立暫存資料夾，然 後將該資料夾複製到被控端電腦 參數含有磁碟機及路徑，以便在被控端 電腦上建立資料夾 如果執行指令，傳回值是 TRUE
boolean ExecuteHostFile(BSTR FileName);	參數 = 字串形式的 FileName 傳回值 = Boolean	執行被控端電腦上的現有檔案。此函數 只會執行批次檔、指令及執行檔。它不 會執行與執行檔相關聯的檔案。例如， 如果您執行 .doc 檔案，便不會開啟 Microsoft Word 參數含有被控端電腦檔案的完整路徑 如果執行指令，傳回值是 TRUE
BSTR GetError();	傳回值 = 字串	以字串的形式傳回 Awrem32 中最後一個 錯誤。
short ConnectionStatus();	傳回值 = Short	將目前的連線狀態傳回被控端電腦 可能的值包括： ■ -1 = 失去連線 ■ 0 = 無連線 ■ 1 = 階段作業已連線

索引

A

API 程式庫 13
Awrem32
 函數 47, 74

C

C++ 。, 請參閱 Visual C++
CAPI 連線
 主控端屬性 25, 59
 被控端內容 39, 69
CHostData
 使用 Visual Basic 31
 使用 Visual C++ 64
CHostDataEx
 使用 Visual Basic 40
 使用 Visual C++ 71
CHostDataManager
 使用 Visual Basic 29
 使用 Visual C++ 62
COM 裝置
 主控端裝置細節 22, 57
 被控端裝置細節 36, 66
CRemoteData
 使用 Visual Basic 18
 使用 Visual C++ 54
CRemoteDataEx
 使用 Visual Basic 26
 使用 Visual C++ 60
CRemoteDataManager
 使用 Visual Basic 16
 使用 Visual C++ 52

G

GUID 10

I

ISDN
 CAPI 連線
 主控端屬性 25, 59
 被控端內容 39, 69

N

NASI 裝置 69
NCSI 裝置 69
NetBIOS 連線
 主控端屬性 25, 59
 被控端內容 39, 69

O

OLE 自動化。 , 請參閱 pcAnywhere 自動化伺服器

P

pcAnywhere 自動化伺服器
 使用 Visual Basic 存取 11
 使用 Visual C++ 存取 12
 註冊 GUID 10
 範例用途 10
 關於 10
 類型程式庫 13

T

TAPI 裝置
 在主控端上 57
 在被控端上 67

V

Visual Basic
 Awrem32 函數 47
 CHostData 物件 31
 CHostDataEx 物件 40
 CHostDataManager 物件 29
 CRemoteData 物件 18
 CRemoteDataEx 物件 26
 CRemoteDataManager 物件 16
 存取 pcAnywhere 自動化伺服器 11
 程式碼範例 27, 46
Visual C++
 Awrem32 函數 74
 CHostData 物件 64
 CHostDataEx 物件 71
 CHostDataManager 物件 62

CRemoteData 物件 54
 CRemoteDataEx 物件 60
 CRemoteDataManager 物件 52
 引用 Winawsvr.h 14
 存取 pcAnywhere 自動化伺服器 12
 程式碼範例 60, 73
 匯入類別 13

W

Winawsvr

物件
 CHostData 31, 64
 CHostDataEx 40, 71
 CHostDataManager 29, 62
 CRemoteData 18, 54
 CRemoteDataEx 26, 60
 CRemoteDataManager 16, 52
 執行檔 11
 標頭檔案 14

五劃

主控端引擎

手動註冊 11
 自動註冊 11

主控端物件

方法 59
 目錄 16, 52
 刪除 54
 建立 54
 密碼 60
 裝置細節 57
 撥號內容 21
 檔案 16-17, 29, 52-53

加密設定

在主控端上 26-27, 60
 在被控端上 40, 71

六劃

同位元 22, 36, 57, 67

自動化伺服器。請參閱 pcAnywhere 自動化伺服器
 自動化控制器

關於 Visual Basic 11
 關於 Visual C++ 12

八劃

狀態

傳回 49, 75

九劃

流量控制 22, 37, 57, 67

十一劃

國碼

在主控端上 60
 在被控端上 71

密碼

在主控端物件上 60
 在被控端物件上 71

被控端物件

方法 70
 目錄 29, 62
 刪除 30, 64
 建立 64
 密碼 71
 啟動 31, 64
 尋找 29-30, 62-63
 裝置細節 66
 撥號內容 35

連線

啟動 31, 48, 54, 74
 結束 48, 74
 傳回狀態 49, 75
 傳回錯誤 75

連線類型

主控端屬性 20, 57, 59
 取消指定 35
 指派 34
 被控端內容 32, 66, 70

十二劃

程式碼範例

Visual Basic 27, 46
 Visual C++ 60, 73

十三劃

裝置。請參閱 連線類型

資料夾

在被控端上建立 48, 75

十四劃

數據機連線

在主控端上
 COM 屬性 22, 57
 TAPI 裝置 57
 國碼 60

- 撥號內容 21
- 在被控端上
 - COM 屬性 36, 66
 - TAPI 裝置 67
 - 國碼 71
 - 撥號內容 35
- 網路連線
 - 在主控端上
 - NetBIOS 屬性 25, 59
 - 在被控端上
 - NetBIOS 屬性 39, 69

十六劃

- 錯誤
 - 傳回 48, 75

十七劃

- 檔案
 - 在被控端上執行 48, 75
- 檔案傳輸
 - 到被控端 48, 74
 - 從被控端 48, 74

十九劃

- 識別碼, 請參閱 GUID
- 類別定義
 - 匯入 13
 - 檢視 14
- 類型程式庫 13