

Guía de automatización OLE

Guía de automatización OLE

Aviso legal

Copyright © 2008 Symantec Corporation. Reservados todos los derechos.

Symantec, el logotipo de Symantec son marcas comerciales o marcas comerciales registradas de Symantec Corporation o de sus afiliadas en los EE.UU. y en otros países. El resto de los nombres son marcas registradas de sus respectivos propietarios.

Este producto de Symantec puede contener software de otros fabricantes, en cuyo caso Symantec debe brindar atribución al tercero ("Programas de otros fabricantes"). Algunos de los Programas de otros fabricantes están disponibles bajo licencias de código abierto o software libre. El Acuerdo de licencia que acompaña el Software no altera los derechos ni las obligaciones que es posible que tenga bajo las licencias de código abierto o software libre. Consulte el Apéndice del Aviso legal de otros fabricantes adjunto a esta documentación o el archivo Léame de propiedad intelectual de otro fabricante, que acompaña este producto de Symantec para obtener más información acerca del Programa de otros fabricantes.

El producto descrito en este documento se distribuye bajo licencias que restringen su uso, copia, distribución, descompilación e ingeniería inversa. Ninguna parte de este documento se podrá reproducir de ninguna forma y por ningún medio sin la autorización previa, por escrito, de Symantec Corporation y sus licenciarios, si corresponde.

LA DOCUMENTACIÓN SE PROPORCIONA "TAL CUAL" Y SE RENUNCIA A TODOS LOS TÉRMINOS IMPLÍCITOS O EXPRESOS, REPRESENTACIONES Y GARANTÍAS, INCLUIDA CUALQUIER GARANTÍA DE COMERCIABILIDAD Y DE ADECUACIÓN PARA UN FIN CONCRETO, A EXCEPCIÓN DE QUE LAS LIMITACIONES SE DECLAREN NO VÁLIDAS LEGALMENTE. SYMANTEC CORPORATION NO SERÁ RESPONSABLE DE NINGÚN DAÑO INCIDENTAL O CONSECUENTE RELACIONADO CON LA ENTREGA O USO DE ESTA DOCUMENTACIÓN. LA INFORMACIÓN QUE CONTIENE ESTE DOCUMENTO ESTÁ SUJETA A CAMBIOS SIN PREVIO AVISO.

La documentación y el software con licencia se consideran software informático comercial, como están definidos en el artículo 12.212 de las FAR, y están sujetos a derechos restringidos, como están definidos en el artículo 52.227-19 de las FAR, y "Software informático comercial: derechos restringidos" del artículo 227.7202 del Suplemento a las FAR, "Derechos del software informático comercial o Documentación de software comercial para equipos informáticos", según corresponda y cualquier regulación sucesora. Cualquier uso, modificación, edición de reproducción, ejecución, muestra o divulgación de la documentación y el software con licencia por parte del gobierno de los EE.UU. se realizará exclusivamente conforme a las condiciones del presente Acuerdo.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

<http://www.symantec.com.mx>

Fabricado en Irlanda

10 9 8 7 6 5 4 3 2 1

Soporte técnico

El Soporte técnico de Symantec presta servicios de soporte técnico a nivel global. El papel principal del soporte técnico es responder a las consultas específicas sobre funcionalidad y funciones del producto. El grupo de soporte técnico también desarrolla contenido para nuestra Base de conocimientos en línea. El grupo de soporte técnico trabaja en colaboración con las otras áreas funcionales que forman parte de Symantec para responder a sus preguntas de manera oportuna. Por ejemplo, el grupo de soporte técnico trabaja con Ingeniería de productos y con Symantec Security Response para brindar servicios de alerta y actualizaciones de definiciones de virus.

El mantenimiento que brinda Symantec ofrece lo siguiente:

- Una gama de opciones de soporte que le otorga la flexibilidad para seleccionar el volumen de servicio requerido según el tamaño de la organización.
- Servicios de soporte telefónico y basado en web que brindan una respuesta rápida e información de último momento.
- Seguro de actualización que suministra protección automática para la actualización de software.
- Soporte de alcance global disponible las 24 horas del día, los 7 días de la semana.
- Funciones avanzadas, incluidos los Servicios de administración de cuentas

Para obtener más información acerca de los programas de mantenimiento de Symantec, visite nuestro sitio web en la siguiente dirección URL:

www.symantec.com/techsupp/

Contacto con el soporte técnico

Los clientes que tienen un acuerdo de mantenimiento actual pueden acceder a la Información del soporte técnico en la siguiente dirección URL:

www.symantec.com/techsupp/

Antes de contactar al soporte técnico, asegúrese de haber cumplido los requisitos del sistema que se enumeran en la documentación del producto. Asimismo, es necesario estar en el equipo en el cual ocurrió el problema en caso de que sea necesario replicarlo.

Cuando contacta al soporte técnico, tenga la siguiente información disponible:

- Nivel de la versión de producto
- Información del hardware
- Memoria disponible, espacio libre en el disco e información de la NIC

- Sistema operativo
- Versión y nivel del parche
- Topología de redes
- Router, gateway e información de la dirección IP
- Descripción del problema:
 - Mensajes de error y archivos de registro
 - Solución de problemas realizada antes de contactar a Symantec
 - Cambios de configuración del software reciente y cambios de la red

Concesión de licencia y registro

Si el producto de Symantec debe registrarse o requiere una clave de licencia, visite nuestra página web de soporte técnico en la siguiente dirección URL:

www.symantec.com/techsupp/

Servicio al cliente

En la siguiente dirección encontrará información sobre el Servicio al cliente:

www.symantec.com/techsupp/

El Servicio al cliente está disponible para brindarle asistencia en lo que respecta a:

- Cuestiones relacionadas con las licencias de productos o la serialización.
- Actualizaciones de registros de productos, como cambios de direcciones o nombres.
- Información general sobre productos (funciones, disponibilidad de idiomas, representantes locales).
- Información más reciente sobre actualizaciones y mejoras de productos.
- Información acerca del seguro de actualización y los contratos de mantenimiento
- Información sobre los programas de compras de Symantec
- Consejos sobre las opciones que brinda el soporte técnico de Symantec.
- Cuestiones previas a las ventas que no están relacionadas con lo técnico.
- Aspectos relacionados con los CD-ROM o los manuales.

Recursos para acuerdos de mantenimiento

Si desea consultar a Symantec en relación con algún acuerdo de mantenimiento existente, comuníquese con el equipo de administración de acuerdos de mantenimiento de su región:

Pacífico asiático y Japón	contractsadmin@symantec.com
Europa, Oriente medio y África	semea@symantec.com
Norteamérica y América Latina	supportsolutions@symantec.com

Servicios empresariales adicionales

Symantec ofrece un conjunto completo de servicios que le permiten maximizar su inversión en los productos de Symantec y desarrollar su conocimiento, pericia y visión global. De esta manera, podrá administrar sus riesgos empresariales de manera proactiva.

Algunos de los servicios empresariales disponibles son los siguientes:

Soluciones de alerta temprana de Symantec	Estas soluciones brindan una alerta temprana sobre ataques cibernéticos, análisis exhaustivos de amenazas y medidas preventivas para evitar ataques antes de que ocurran.
Servicios de seguridad administrada	Estos servicios lo liberan de la carga que representa administrar y supervisar dispositivos y eventos, lo que garantiza una respuesta rápida ante amenazas reales.
Servicios de consultoría	Los servicios de consultoría de Symantec ponen a su disponibilidad, en el propio lugar de trabajo, la pericia técnica de Symantec y de sus partners de confianza. Los servicios de consultoría de Symantec ofrecen una variedad de opciones personalizables y empaquetadas previamente, que incluyen funciones de evaluación, diseño, supervisión y administración. Cada uno se centra en establecer y mantener la integridad y la disponibilidad de sus recursos de TI.
Servicios educativos	Los servicios educativos brindan una serie completa de programas de entrenamiento técnico, educación y certificación en seguridad, y programas de comunicación de los conocimientos.

Para tener acceso a más información acerca de los servicios empresariales, visite nuestro sitio web en la siguiente dirección URL:

www.symantec.com

Seleccione un país o idioma del índice del sitio.

Contenido

Soporte técnico	4
Capítulo 1	Uso de la automatización OLE con Symantec pcAnywhere
	9
	Acerca de la automatización OLE
	9
	Acerca del servidor de automatización de pcAnywhere
	10
	Acciones que puede realizar con el servidor de automatización de pcAnywhere
	10
	Antes de comenzar
	11
	Registrar automáticamente el motor remoto
	11
	Registrar el motor remoto de forma manual
	12
	Acceso al servidor de automatización de pcAnywhere
	12
	Acceso al servidor de automatización de pcAnywhere con Visual Basic
	12
	Acceso al servidor de automatización de pcAnywhere con Visual C++
	13
	Ejecutar los objetos OLE host y remotos
	16
	Dónde encontrar más información
	16
Capítulo 2	Definiciones de objetos de Visual Basic
	17
	Acerca de los objetos de Visual Basic
	17
	Métodos de CRemoteDataManager
	18
	Propiedades de CRemoteData
	21
	Propiedades del tipo de conexión
	23
	Propiedades de marcado
	25
	Propiedades de un dispositivo COM
	26
	Propiedades de un dispositivo NetBIOS
	29
	Propiedades del dispositivo ISDN mediante CAPI 2.0
	30
	Objeto CRemoteDataEx
	31
	Código de muestra de Visual Basic para las funciones remotas
	32
	Métodos de CHostDataManager
	34
	Propiedades de CHostData
	37
	Propiedades del tipo de conexión
	39
	Método de AssignConnection(ConnectionType)
	42

- Método de UnassignConnection(ConnectionType) 42
- Propiedades de marcado 43
- Propiedades de un dispositivo COM 44
- Propiedades de un dispositivo NetBIOS 47
- Propiedades del dispositivo ISDN mediante CAPI 2.0 48
- Objeto CHostDataEx 48
 - Código de muestra de Visual Basic para las funciones de host 56
- Funciones de Awrem32 58

Capítulo 3 Definiciones de objetos de Visual C++ 61

- Acerca de los objetos de Visual C++ 61
- Métodos de CRemoteDataManager 62
- Objeto CRemoteData 66
 - Métodos Get y Set 66
 - Métodos de detalles de objetos remotos 68
 - Métodos de objetos remotos 71
- Objeto CRemoteDataEx 72
 - Código de muestra de Visual C++ para las funciones remotas 73
- Métodos de CHostDataManager 74
- Objeto CHostData 78
 - Métodos Get y Set 78
 - Métodos de detalles de objetos host 79
 - Métodos de objetos host 83
- Objeto CHostDataEx 85
 - Código de muestra de Visual C++ para las funciones de host 87
- funciones de Awrem32 88

Índice 91

Uso de la automatización OLE con Symantec pcAnywhere

En este capítulo se incluyen los temas siguientes:

- [Acerca de la automatización OLE](#)
- [Acerca del servidor de automatización de pcAnywhere](#)
- [Acciones que puede realizar con el servidor de automatización de pcAnywhere](#)
- [Antes de comenzar](#)
- [Dónde encontrar más información](#)

Acerca de la automatización OLE

La automatización OLE es una tecnología que permite crear una aplicación externa u otro tipo de herramienta de desarrollo (como scripts o macros) que controlan y automatizan cualquier función expuesta dentro de una aplicación.

La automatización OLE consta de los siguientes componentes:

Servidor de automatización OLE

Aplicación o componente de software que expone sus funciones de manera que se pueda tener acceso a ellas o controlarlas mediante otras aplicaciones o herramientas de desarrollo.

El servidor de automatización de pcAnywhere es un ejemplo de servidor de automatización OLE.

Controlador de automatización OLE

Aplicación o herramienta de desarrollo que tiene acceso a los componentes que han sido expuestos por el servidor de automatización OLE y los controla.

Puede utilizar cualquier lenguaje de programación que admita la automatización OLE. Los dos lenguajes de programación más comunes son Microsoft Visual Basic y Microsoft Visual C++.

Una aplicación externa tiene acceso a un servidor de automatización OLE al conectarse con el servidor y luego solicitar acceso a una o más interfaces publicadas. Una interfaz es un punto de entrada que permite el acceso a uno o más métodos o propiedades relacionados. Después de que una aplicación obtiene una interfaz en el servidor, puede llamar a cualquier método de interfaz interna como si fuera parte de la aplicación externa.

Acerca del servidor de automatización de pcAnywhere

El servidor de automatización de pcAnywhere permite que las aplicaciones externas administren los archivos de información de pcAnywhere, del equipo remoto y del host para automatizar las tareas de control remoto y de transferencia de archivos. El servidor de automatización de pcAnywhere funciona como un sustituto programable para la interfaz de usuario de Symantec pcAnywhere y refleja la mayor parte de su comportamiento predeterminado.

Por ejemplo, cuando se crea un objeto host en pcAnywhere, se le asigna el primer módem para TAPI disponible de manera predeterminada. Del mismo modo, cuando se crea un objeto host mediante el servidor de automatización de pcAnywhere y se enumera la lista de conexiones asignadas, el primer módem para TAPI disponible ya está asignado.

Acciones que puede realizar con el servidor de automatización de pcAnywhere

El servidor de automatización de pcAnywhere permite automatizar una gran variedad de tareas de productividad y de administración.

Es posible usar el servidor de automatización de pcAnywhere para hacer las siguientes acciones:

- Distribuir e instalar automáticamente actualizaciones de software en varios equipos de la red.

- Programar transferencias de archivos automáticas entre equipos con fines de almacenamiento y auditoría.
- Agregar automáticamente un nombre a la lista de comunicantes permitidos, o eliminarlo de ella, en cada host de pcAnywhere de la red.

Este documento contiene varios ejemplos, escritos tanto en Visual Basic como en Visual C++, para ilustrar cómo conectarse y utilizar el servidor de automatización de pcAnywhere.

Ver "[Acerca de los objetos de Visual Basic](#)" en la página 17.

Ver "[Acerca de los objetos de Visual C++](#)" en la página 61.

Antes de comenzar

Durante una conexión con el servidor de automatización de pcAnywhere y sus interfaces, los parámetros de identificación, denominados "identificadores únicos globales" (GUID, del inglés Globally Unique Identifier), se transfieren a las funciones API de la biblioteca de automatización. Se asigna un GUID independiente al servidor de automatización de pcAnywhere y a cada interfaz expuesta. Estos GUID deben presentarse en el registro del sistema para conectar una aplicación externa con el servidor de automatización de pcAnywhere y sus interfaces.

Puede registrar las entradas de GUID de forma automática si está ejecutando la aplicación externa en un equipo que tenga instalado Symantec pcAnywhere. De lo contrario, debe registrar las entradas de GUID de forma manual.

Ver "[Registrar automáticamente el motor remoto](#)" en la página 11.

Ver "[Registrar el motor remoto de forma manual](#)" en la página 12.

Registrar automáticamente el motor remoto

Antes de conectarse con otro equipo por primera vez mediante el cliente OLE, debe registrar el motor remoto. Puede hacerlo automáticamente iniciando el objeto remoto mediante Symantec pcAnywhere.

Para registrar automáticamente el motor remoto

- 1 Para abrir Symantec pcAnywhere, realice una de las siguientes acciones:
 - En el escritorio, haga doble clic en el icono de programa de Symantec pcAnywhere.

- En la barra de tareas de Windows, haga clic en **Inicio> Programas> Symantec pcAnywhere**.
- 2 En la ventana del Administrador de pcAnywhere, haga clic en **Remotos**.
- 3 Haga doble clic en un elemento de conexión remota.

Con este proceso se registrará el motor remoto. No es necesario completar la conexión.

Registrar el motor remoto de forma manual

Si pcAnywhere no está instalado en el equipo en el que está ejecutando la aplicación externa, debe registrar los GUID manualmente mediante la ejecución del archivo ejecutable del servidor de automatización de pcAnywhere (Winawsvr.exe). Sólo tiene que ejecutar este archivo una vez para agregar los GUID al registro. El archivo Winawsvr.exe está situado en el directorio de instalación.

Acceso al servidor de automatización de pcAnywhere

Se puede acceder al servidor de automatización de pcAnywhere mediante cualquier lenguaje compatible con la automatización OLE. Los dos lenguajes más populares que son compatibles con la automatización OLE son Visual Basic y Visual C++.

Los principios de codificación de estos dos lenguajes son muy similares, aunque en el entorno de Visual Basic la mayor parte del trabajo de bajo nivel lo ejecuta en segundo plano el sistema de tiempo de ejecución de Visual Basic.

Acceso al servidor de automatización de pcAnywhere con Visual Basic

El lenguaje de programación Visual Basic cuenta con soporte incorporado para interactuar con los servidores de automatización OLE, como el servidor de automatización de pcAnywhere. Cree un proyecto ejecutable estándar y, a continuación, introduzca un código en cada método para tener acceso al servidor de automatización de pcAnywhere. Visual Basic toma las llamadas del método de alto nivel de los archivos de origen y las expande internamente en las llamadas de método de automatización OLE de bajo nivel correspondientes.

Para tener acceso al servidor de automatización de pcAnywhere con Visual Basic

- 1 Agregue dos variables de objeto para cada objeto de pcAnywhere al que desee acceder.

Por ejemplo, al trabajar con objetos remotos, aplique DIM a **RemoteDataManager** y **RemoteDataObject** como objeto

- 2 Utilice RemoteDataManager para adjuntarlo al administrador de datos del objeto remoto.

Por ejemplo, llame al método CreateObject con **WINAWSVR.REMOTEDATAMANAGER** como parámetro.

Visual Basic utilizará el parámetro textual para localizar en el registro el identificador del administrador y devolverá la interfaz a ese administrador.

- 3 Cuando tenga un objeto de administración de datos válido, puede utilizarlo para realizar cualquiera de las siguientes acciones:
 - Determinar el directorio actual.
 - Cambiar a otro directorio.
 - Enumerar los archivos de objetos de datos asociados del directorio actual.
 - Crear, obtener o eliminar un archivo de objetos de datos.
- 4 Después de haber creado u obtenido un objeto de datos, podrá obtener o fijar las propiedades de dicho objeto.

La sintaxis de Visual Basic no utiliza el nombre de una propiedad para diferenciar la obtención y la fijación de su valor. En cambio, es la posición de la propiedad respecto al operador asignado lo que determina si el método implicado es Get o Set.

Los siguientes ejemplos muestran un caso de Get y otro de Set:

- Para obtener el valor de número de teléfono de un objeto, coloque el nombre de la propiedad a la derecha del operador asignado.
 Por ejemplo, s = RemoteData.PhoneNumber(), donde s es una variable de cadena.
- Para definir el número de teléfono, coloque el nombre de la propiedad a la izquierda del operador asignado.
 Por ejemplo, RemoteData.PhoneNumber = "555-1212"

Acceso al servidor de automatización de pcAnywhere con Visual C++

El servidor de automatización de pcAnywhere utiliza bibliotecas de tipos para exponer información acerca de sus interfaces y sus métodos a los clientes de

automatización creados en Visual C++. Estas bibliotecas de tipos utilizan Microsoft Foundation Classes (MFC), que pueden importarse en la aplicación mediante el Asistente para clases de Visual C++.

Las clases del administrador de datos que se ofrecen en las bibliotecas de tipos incluyen funciones necesarias para obtener una interfaz en el servidor de automatización de pcAnywhere y llevar a cabo operaciones de alto nivel en el tipo de objeto asociado de la interfaz.

Utilice el objeto de administración de datos para realizar las siguientes acciones:

- Determinar o cambiar el directorio actual.
- Enumerar la lista de archivos de objetos de datos del directorio actual.
- Crear, obtener o eliminar un objeto determinado.

Una vez creados u obtenidos, los objetos utilizan las clases de objetos de datos asociadas para examinar o modificar cualquiera de las propiedades expuestas. La mayor parte de estas propiedades se exponen mediante dos métodos que comienzan con las palabras Get o Set. Por ejemplo, un usuario llama al método GetPhoneNumber para examinar la propiedad del número de teléfono actual del objeto y llama a SetPhoneNumber para configurarla.

Importación y visualización de clases

El servidor de automatización de pcAnywhere utiliza las siguientes bibliotecas de tipos:

Winawsvr.tlb	Ofrece la información necesaria para conectarse con el servidor de automatización de pcAnywhere y tener acceso a sus interfaces.
Awrem32.tlb	Ofrece la información necesaria para controlar las conexiones de pcAnywhere.

Los siguientes procedimientos explican cómo importar las definiciones de clases desde las bibliotecas de tipos del servidor de automatización de pcAnywhere en la aplicación MFC y, a continuación, cómo ver las clases que se han agregado a la aplicación.

Para importar clases

- 1 En Visual C++, cree una aplicación MFC.
- 2 En el menú Ver, haga clic en **Class Wizard**.
- 3 En el cuadro de diálogo Class Wizard, haga clic en **Add Class** y, a continuación, haga clic en **From a type library**.
- 4 Haga doble clic en **winawsvr.tlb**.

- 5 En el cuadro de diálogo Confirm Classes, haga clic en **OK** para importar todas las definiciones de clases.
- 6 En el cuadro de diálogo Class Wizard, haga clic en **Add Class** y, a continuación, haga clic en **From a type library**.
- 7 Haga doble clic en **awrem32.tlb**.
- 8 En el cuadro de diálogo Confirm Classes, haga clic en **OK** para importar todas las definiciones de clases.
- 9 En el cuadro de diálogo Class Wizard, haga clic en **OK** para completar la importación.

Se agregarán las clases a la aplicación. Estas clases permiten manipular objetos y administrar conexiones.

La importación de las definiciones de clases desde las bibliotecas de tipos también agrega archivos de soporte a la aplicación. Estos archivos contienen definiciones de clases y código fuente de implementación para el servidor de automatización de pcAnywhere.

Ver "[Ver las definiciones de clase y los archivos de implementación](#)" en la página 15.

Para ver las clases agregadas

- 1 En Visual C++, abra la aplicación MFC.
- 2 En la ventana del área de trabajo, haga clic en la ficha **ClassView**.

Ver las definiciones de clase y los archivos de implementación

Cuando importa las bibliotecas de tipos del servidor de automatización de pcAnywhere en la aplicación, se agregan los siguientes archivos:

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

Estos archivos contienen definiciones de clases y código fuente de implementación para el servidor de automatización de pcAnywhere. No necesita modificar estos archivos. No obstante, cada archivo de origen de la aplicación que contiene llamadas a los métodos del servidor de automatización de pcAnywhere debe incluir Winawsvr.h.

Para ver las definiciones de clase y los archivos de implementación

- 1 En Visual C++, abra la aplicación MFC.
- 2 En la ventana del área de trabajo, haga clic en la ficha **FileView** (Vista de archivo).

Ejecutar los objetos OLE host y remotos

Symantec pcAnywhere requiere que todos los objetos host y remotos se almacenen en el directorio de datos predeterminado. Antes de ejecutar un objeto host o remoto que haya creado con OLE, asegúrese de que esté situado en el directorio de datos predeterminado de pcAnywhere. El directorio predeterminado es la carpeta Todos los usuarios en Windows XP y la carpeta ProgramData en Vista.

Dónde encontrar más información

Para obtener más información acerca de la automatización OLE, consulte las siguientes referencias:

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

Definiciones de objetos de Visual Basic

En este capítulo se incluyen los temas siguientes:

- [Acerca de los objetos de Visual Basic](#)
- [Métodos de CRemoteDataManager](#)
- [Propiedades de CRemoteData](#)
- [Objeto CRemoteDataEx](#)
- [Métodos de CHostDataManager](#)
- [Propiedades de CHostData](#)
- [Objeto CHostDataEx](#)
- [Funciones de Awrem32](#)

Acerca de los objetos de Visual Basic

El servidor de automatización de pcAnywhere ofrece los siguientes componentes para admitir la automatización OLE:

Winawsvr	Ofrece la información necesaria para conectarse con el servidor de automatización de pcAnywhere y tener acceso a sus interfaces.
Awrem32	Ofrece la información necesaria para controlar las conexiones de pcAnywhere.

En la [Tabla 2-1](#), se describen los objetos que componen Winawsvr.

Tabla 2-1 Objetos de Winawsvr

Objeto	Descripción	Referencia
CRemoteDataManager	Ofrece los métodos para crear, abrir, modificar, guardar y eliminar los objetos CRemoteData.	Ver " Métodos de CRemoteDataManager " en la página 18.
CRemoteData	Define los parámetros para tener acceso a las funciones remotas de pcAnywhere y controlarlas.	Ver " Propiedades de CRemoteData " en la página 21. Ver " Objeto CRemoteDataEx " en la página 31.
CHostDataManager	Ofrece los métodos para crear, abrir, modificar, guardar y eliminar los objetos CHostData.	Ver " Métodos de CHostDataManager " en la página 34.
CHostData	Define los parámetros para tener acceso a las funciones de host de pcAnywhere y controlarlas.	Ver " Propiedades de CHostData " en la página 37. Ver " Objeto CHostDataEx " en la página 48.

Awrem32 cuenta con un objeto, que está compuesto de ocho interfaces que admiten las sesiones de control remoto y de transferencia de archivos.

Ver "[Funciones de Awrem32](#)" en la página 58.

Algunas funciones ya no se admiten. Sin embargo, las definiciones de objeto se proporcionan para utilizarlas con versiones anteriores de pcAnywhere.

Para las funciones que requieran el uso de contraseñas, será posible configurar sus valores pero no recuperarlos. Esto se debe a motivos de seguridad.

Métodos de CRemoteDataManager

Los métodos de CRemoteDataManager ofrecen parámetros y valores devueltos para tener acceso a los objetos CRemoteData y controlarlos.

Tabla 2-2 Métodos de CRemoteDataManager

Método de CRemoteDataManager	Parámetro o valor devuelto	Descripción
CurrentDirectory()	Valor devuelto = String	Devuelve la ruta completa del directorio actual en el que se almacenan los objetos remotos de pcAnywhere.

Método de CRemoteDataManager	Parámetro o valor devuelto	Descripción
ChangeDirectory(NewDirectory)	Parámetro = NewDirectory	Cambia el directorio actual en el que se almacenan los objetos remotos de pcAnywhere.
FindFirst(Pattern)	Parámetro = Pattern as string	Busca el primer archivo del objeto remoto de pcAnywhere (*.chf) en el directorio actual; filtros del patrón de búsqueda de nombre del archivo basados en el patrón especificado de nombre de archivo (un asterisco [*] busca todos los archivos en el directorio actual).
FindFirst(Name string)	Parámetro = Name as string	Devuelve el nombre y la ruta del archivo de objeto remoto (*.chf) que cumple el patrón especificado.
FindNext(Name)	Parámetro = Name as string	Devuelve el nombre y la ruta del archivo de objeto remoto que cumpla el patrón de búsqueda indicado en la llamada original a FindFirst().
FindNext(Name)	Valor devuelto = Boolean	TRUE si se encuentra algún otro archivo de objeto remoto que cumpla el patrón indicado en la llamada a FindFirst(). El nombre completo de la ruta del archivo se almacena en Name.
RetrieveObject(Name)	Parámetro = Name as string	El nombre y la ruta del archivo de objeto remoto que se debe cargar.
RetrieveObject(AccessMode)	Parámetro = AccessMode as integer	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
RetrieveObject>Password)	Parámetro = Password as string	Contraseña del objeto. Puede tener el valor NULL.

Método de CRemoteDataManager	Parámetro o valor devuelto	Descripción
RetrieveObjectEx(Name)	Parámetro = Name as string	El nombre y la ruta del archivo de objeto remoto que se debe cargar.
RetrieveObjectEx(AccessMode)	Parámetro = AccessMode as integer	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
RetrieveObjectEx>Password)	Parámetro = Password as string	Contraseña del objeto. Puede tener el valor NULL.
RetrieveObjectEx	Valor devuelto = Object	Objeto CRemoteDataEx del archivo indicado.
CreateObject(Name)	Parámetro = Name as string	Crea un objeto CRemoteData y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto remoto para el nuevo objeto.
CreateObject	Valor devuelto = Object	CRemoteData
CreateObjectEx(Name)	Parámetro = Name as string	Crea un objeto CRemoteDataEx y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto remoto para el nuevo objeto.
CreateObjectEx	Valor devuelto = Object	CRemoteDataEx
DeleteObject(Name)	Parámetro = Name as string	El nombre y la ruta del archivo de objeto remoto que se debe eliminar.
DeleteObject>Password)	Parámetro = Password as string	Contraseña del objeto.
DeleteObject	Valor devuelto = Boolean	TRUE si el objeto se ha eliminado.

Propiedades de CRemoteData

Las propiedades y los parámetros que están disponibles para el objeto CRemoteData se muestran de la siguiente manera.

En la [Tabla 2-3](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-3 Propiedades y parámetros de CRemoteData

Propiedad	Parámetro	Descripción
<CRemoteData>.ComputerName(String)	String	Define el nombre del equipo o la dirección IP del equipo host.
<CRemoteData>.PhoneNumber(String)	String	Define el número de teléfono del equipo host.
<CRemoteData>.UseDialingProperties(Bool)	Bool	Define las propiedades de marcado del sistema.
<CRemoteData>.RedialCount(Integer)	Integer	Define el número de intentos de rellamada antes de cancelarla.
<CRemoteData>.RedialDelay(Integer)	Integer	Define el número de segundos que hay que esperar entre los intentos de rellamada.
<CRemoteData>.AutoLoginName(String)	String	Define el nombre del usuario para la conexión automática. Para obtener más información sobre los registros de dominio: Ver " Objeto CRemoteDataEx " en la página 31.
<CRemoteData>.AutoLoginPassword(String)	String	Define la contraseña para las conexiones automáticas en el objeto remoto. Por motivos de seguridad, el servidor de automatización de pcAnywhere no ofrece la posibilidad de leer el valor de la contraseña. No se devuelve el valor de la contraseña.
<CRemoteData>.Password(String)	String	Define la contraseña en el objeto remoto para su uso con la configuración ExecuteProtection, ReadProtection y WriteProtection. Por motivos de seguridad, el servidor de automatización de pcAnywhere no ofrece la posibilidad de leer el valor de la contraseña. No se devuelve el valor de la contraseña.

Propiedad	Parámetro	Descripción
<CRemoteData>.ExecuteProtection(Bool)	Bool	Establece que sea necesaria una contraseña para ejecutar el objeto. Definido por el parámetro Password.
<CRemoteData>.ReadProtection(Bool)	Bool	Establece que sea necesaria una contraseña para ver las propiedades del objeto remoto. Definido por el parámetro Password.
<CRemoteData>.WriteProtection(Bool)	Bool	Establece que sea necesaria una contraseña para guardar los cambios en el objeto remoto. Definido por el parámetro Password.
<CRemoteData>.LogSession(Bool)	Bool	Activa y desactiva el registro de la sesión.
<CRemoteData>.RecordFile(String)	String	Define la ruta y el nombre completo correspondiente a la ubicación del archivo que registra la sesión activa.
<CRemoteData>.RecordSession(Bool)	Bool	Activa y desactiva el registro de sesión automático.
<CRemoteData>.ReadObject(String)	String	Define la contraseña del objeto, que puede utilizarse para actualizar la copia de datos locales del objeto remoto.
<CRemoteData>.WriteObject(String)	String	Define la contraseña del objeto, que se utilizará para crear el objeto remoto o para guardar los cambios que haya realizado en el objeto remoto.

Las propiedades y los valores devueltos de CRemoteData se muestran en la tabla siguiente.

En la [Tabla 2-4](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-4 Propiedades y valores devueltos de CRemoteData

Propiedad	Valor devuelto	Descripción
String = <CRemoteData>.ComputerName	String	Devuelve el nombre del equipo o la dirección IP del equipo host.
String = <CRemoteData>.PhoneNumber	String	Devuelve el número de teléfono del equipo host.

Propiedad	Valor devuelto	Descripción
Bool = <CRemoteData>.UseDialingProperties	Bool	Devuelve las propiedades de marcado del sistema que se han definido en el objeto remoto.
Integer = <CRemoteData>.RedialCount	Integer	Devuelve el número de intentos de rellamada que se haya definido en el objeto remoto.
Integer = <CRemoteData>.RedialDelay	Integer	Devuelve el número de segundos entre intentos de rellamada.
String = <CRemoteData>.AutoLoginName	String	Devuelve el nombre de inicio de sesión que se utiliza para las conexiones automáticas.
Bool = <CRemoteData>.ExecuteProtection	Bool	Devuelve el valor de la configuración de ExecuteProtection.
Bool = <CRemoteData>.ReadProtection	Bool	Devuelve el valor de la configuración de ReadProtection.
Bool = <CRemoteData>.WriteProtection	Bool	Devuelve el valor de la configuración de WriteProtection.
Bool = <CRemoteData>.LogSession	Bool	Devuelve el valor TRUE si está activado el registro de la sesión.
String = <CRemoteData>.RecordFile	String	Devuelve la ruta y el nombre completo del archivo de registro de la sesión.
Bool = <CRemoteData>.RecordSession	Bool	Devuelve el valor de la configuración correspondiente al registro de sesiones.

Propiedades del tipo de conexión

En la siguiente tabla, se definen los parámetros y las propiedades del tipo de conexión.

En la [Tabla 2-5](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-5 Propiedades y parámetros del tipo de conexión

Propiedad	Parámetro	Descripción
<CRemoteData>.ConnectionType(String)	String	Define el tipo de conexión del equipo remoto. El valor debe ser un tipo de conexión válido tal y como se haya definido por las funciones FirstConnectionType() y NextConnectionType(). Los siguientes son ejemplos de tipos de conexión válidos: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN mediante CAPI 2.0 ■ Nombre del módem (tal y como aparezca en el equipo)

En la siguiente tabla, se definen las propiedades y los valores devueltos del tipo de conexión.

En la [Tabla 2-6](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-6 Propiedades y valores devueltos del tipo de conexión

Propiedad	Valor devuelto	Descripción
String = <CRemoteData>.ConnectionTypes	String	Devuelve el tipo de conexión del objeto remoto.

Propiedad	Valor devuelto	Descripción
Integer = <CRemoteData>.ConnectionType	Integer	Devuelve el número de tipos de conexión disponibles. Los siguientes son ejemplos de tipos de conexión válidos: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN mediante CAPI 2.0 ■ Nombre del módem (tal y como aparezca en el equipo)
String = <CRemoteData>.FirstConnectionType	String	Devuelve el primer tipo de conexión disponible.
String = <CRemoteData>.NextConnectionType	String	Devuelve el siguiente tipo de conexión disponible. Se denomina secuencial debido al número de tipos de conexión definido en <CRemoteData>.ConnectionType para enumerar todos los tipos de conexión.
Bool = <CRemoteData>.FindConnectionType(ConnectionType)	Bool	Devuelve TRUE si el tipo de conexión especificado se encuentra en la lista de los tipos de conexión disponibles.

Propiedades de marcado

En la siguiente tabla, se definen las propiedades y los parámetros para configurar las propiedades de marcado para las conexiones por módem.

En la [Tabla 2-7](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-7 Propiedades y parámetros de las propiedades de marcado

Propiedad	Parámetro	Descripción
<CRemoteData>.AreaCode(String)	String	Define las propiedades de marcado del código de área para las conexiones por módem.
<CRemoteData>.CountryCode(String)	String	Define las propiedades de marcado del código del país para las conexiones por módem.

En la siguiente tabla, se definen las propiedades y los valores devueltos de las propiedades de marcado del módem.

En la [Tabla 2-8](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-8 Propiedades y valores devueltos de las propiedades de marcado

Propiedad	Valor devuelto	Descripción
String = <CRemoteData>.AreaCode	String	Devuelve las propiedades de marcado del código de área.
String = <CRemoteData>.CountryCode	String	Devuelve las propiedades de marcado del código del país.
Integer = <CRemoteData>.CountryCodes	Integer	Devuelve el número de códigos de países disponibles.
String = <CRemoteData>.FirstCountryCode	String	Devuelve el primer código de país disponible que aparece en el sistema operativo.
String = <CRemoteData>.NextCountryCode	String	Devuelve el siguiente código de país disponible. Se lo invoca de forma secuencial de acuerdo con el número de códigos de países que se definan en <CRemoteData.CountryCodes> para enumerar todos los códigos de los países.

Propiedades de un dispositivo COM

En la siguiente tabla, se describen las propiedades y los parámetros que permiten personalizar la configuración del puerto para el módem y otras conexiones basadas en COM.

En la [Tabla 2-9](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-9 Propiedades y parámetros del dispositivo COM

Propiedad	Parámetro	Descripción
<CRemoteData>.ComParity(String)	String	Define la paridad de las comunicaciones. Los siguientes valores son válidos: ■ <Ninguna> ■ Impar ■ Par ■ Marca ■ Espacio
<CRemoteData>.ComFlowControl(String)	String	Define el control de flujo de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ <Ninguno> ■ XONXOFF ■ RTS/CTS ■ Ambos
<CRemoteData>.ComStartedBy(String)	String	Define la configuración inicial de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI) ■ Recibir 2 <CR> ■ Respuesta del módem
<CRemoteData>.ComEndedBy(String)	String	Define la configuración final de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI)

Propiedad	Parámetro	Descripción
<CRemoteData>.ComSpeed(Long)	Long	Contiene la configuración de velocidad COM máxima. Los siguientes valores son válidos: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

En la siguiente tabla, se describen las propiedades y los valores devueltos del dispositivo COM.

En la [Tabla 2-10](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-10 Propiedades y valores devueltos del dispositivo COM

Propiedad	Valor devuelto	Descripción
String = <CRemoteData>.ComParity	String	Devuelve uno de los siguientes valores para la paridad de las comunicaciones: ■ <Ninguna> ■ Impar ■ Par ■ Marca ■ Espacio
String = <CRemoteData>.ComFlowControl	String	Devuelve la configuración de Com Flow correspondiente al objeto remoto. Los siguientes valores son válidos: ■ <Ninguno> ■ XONXOFF ■ RTS/CTS ■ Ambos

Propiedad	Valor devuelto	Descripción
String = <CRemoteData>.ComStartedBy	String	Devuelve el control Com Start de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI) ■ Recibir 2 <CR> ■ Respuesta del módem
String = <CRemoteData>.ComEndedBy	String	Devuelve el control Com End de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI)
Long = <CRemoteData>.ComSpeed	Long	Devuelve la configuración actual de Com Speed correspondiente al objeto remoto. Los siguientes valores son válidos: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Propiedades de un dispositivo NetBIOS

En la siguiente tabla, se definen las propiedades y el parámetro para un dispositivo de red NetBIOS.

En la [Tabla 2-11](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-11 Propiedad y parámetro de NetBIOS

Propiedad	Parámetro	Descripción
<CRemoteData>.LanaNumber(Integer)	Integer	Define el número del adaptador LAN (LANA) para las conexiones de NetBIOS.

En la siguiente tabla, se definen la propiedad y el valor devuelto de NetBIOS.

En la [Tabla 2-12](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-12 Propiedad y valor devuelto de NetBIOS

Propiedad	Valor devuelto	Descripción
Integer = <CRemoteData>.LanaNumber	Integer	Devuelve la configuración actual del número del adaptador LAN (LANA) para las conexiones de NetBIOS.

Propiedades del dispositivo ISDN mediante CAPI 2.0

En la siguiente tabla, se definen las propiedades y los parámetros para las conexiones europeas ISDN.

En la [Tabla 2-13](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-13 Propiedades y parámetros de ISDN mediante CAPI 2.0

Propiedad	Parámetro	Descripción
<CRemoteData>.CapiChannelBonding(Bool)	Bool	Activa o desactiva el enlace de canales para los dispositivos ISDN CAPI.
<CRemoteData>.CapiExtensions(String)	String	Define cualquier extensión adicional CAPI necesaria para las comunicaciones.

En la siguiente tabla, se definen las propiedades y los valores devueltos para las conexiones europeas ISDN.

En la [Tabla 2-14](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-14 Propiedades y valores devueltos de ISDN mediante CAPI 2.0

Propiedad	Valor devuelto	Descripción
Bool= <CRemoteData>.CapiChannelBonding	Bool	Devuelve la configuración del enlace de canales actual de ISDN CAPI en el objeto remoto.
String = <CRemoteData>.CapiExtensions	String	Devuelve la lista actual de extensiones CAPI desde el objeto remoto.

Objeto CRemoteDataEx

El objeto CRemoteDataEx contiene las mismas funciones que el objeto CRemoteData con algunas funciones adicionales.

En la siguiente tabla, se describen las propiedades y los parámetros.

En la [Tabla 2-15](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-15 Parámetros de CRemoteDataEx

Propiedad	Parámetro	Descripción
<CRemoteData>.PrivateKey(String)	String	Define el nombre del contenedor de clave privada que desee utilizar.
<CRemoteData>.CertificateName(String)	String	Define el nombre común de la clave privada que desee utilizar.
<CRemoteData>.EncryptionLevel(Byte)	Byte	Define el nivel de cifrado. Los siguientes valores son válidos: ■ -1: Ninguno ■ 0: pcAnywhere ■ 1: Simétrico ■ 2: Clave pública
<CRemoteData>.DenyLowerEncrypt(Bool)	Bool	Define si el equipo remoto permite una conexión a un equipo host que utiliza un nivel inferior de cifrado.
<CRemoteData>.AutoDomain(String)	String	Define el nombre de dominio para las conexiones automáticas. Esta opción se utiliza con la autenticación de NT y de Windows.

En la tabla siguiente, se describen las propiedades y los valores devueltos del objeto CRemoteDataEx.

En la [Tabla 2-16](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-16 Propiedades y valores devueltos de CRemoteDataEx

Propiedad	Valor devuelto	Descripción
String = <CRemoteData>.PrivateKey	String	Devuelve el nombre del contenedor de clave privada actualmente activo.
String = <CRemoteData>.CertificateName	String	Devuelve el nombre común del contenedor de clave privada activo.
Byte = <CRemoteData>.EncryptionLevel	Byte	Devuelve uno de los siguientes valores de cifrado: ■ -1: Ninguno ■ 0: pcAnywhere ■ 1: Simétrico ■ 2: Clave pública
Bool = <CRemoteData>.DenyLowerEncrypt	Bool	Devuelve el valor de la configuración para denegar el cifrado más bajo.
String = <CRemoteData>.AutoDomain	String	Devuelve la configuración de nombre de dominio para las conexiones automáticas. Esta opción se utiliza con la autenticación de NT y de Windows.

Código de muestra de Visual Basic para las funciones remotas

El siguiente código de muestra de Visual Basic recupera un objeto de datos remoto y modifica sus propiedades:

```
Private Sub Command1_Click()  
Dim RemoteDataManager as Object  
Dim RemoteData as Object  
Dim s as string  
  
'Crea un objeto CRemoteDataManager  
Set RemoteDataManager = CreateObject (WINAWSVR.REMOTEDATAMANAGER)  
  
'muestra y cambia el directorio actual  
s = RemoteDataManager.CurrentDirectory()  
MsgBox (s)
```

```
RemoteDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
s = RemoteDataManager.CurrentDirectory()
MsgBox (s)

'obtiene un objeto de datos remoto
Set RemoteData = RemoteDataManager.RetrieveObjectEx("pod.CHF",
2, 0)

'muestra algunas propiedades
s = RemoteData.AreaCode()
MsgBox (s)
s = RemoteData.PhoneNumber()
MsgBox (s)

'establece algunas propiedades
RemoteData.AreaCode = "212"
RemoteData.PhoneNumber = "555-5555"

'escribe objetos en el disco
RemoteData.WriteObject (0)
End Sub
```

Utilice los métodos FindFirst y FindNext para mostrar el archivo remoto en un directorio de esta manera:

```
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
```

Cree un objeto remoto. Establezca el tipo de conexión en TCP/IP y el nombre del equipo en "Host1" y, a continuación, ejecútelo de la siguiente manera:

```
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
```

Métodos de CHostDataManager

Los métodos de CHostDataManager ofrecen parámetros y valores devueltos para tener acceso a los objetos CHostData y controlarlos.

Método de CHostDataManager	Parámetro o valor devuelto	Descripción
CurrentDirectory()	Valor devuelto = String	El nombre completo de la ruta del directorio de datos actual de pcAnywhere, en el cual se almacenan los objetos host de pcAnywhere
FindFirst(Pattern)	Parámetro = Pattern as string	Patrón de búsqueda de nombre de archivo para filtrar los archivos de objetos (un asterisco [*] busca todos los archivos del directorio actual).

Método de CHostDataManager	Parámetro o valor devuelto	Descripción
FindFirst(Name string)	Parámetro = Name as string	Devuelve el nombre y la ruta del archivo de objeto host que cumple el patrón especificado.
FindFirst	Valor devuelto = Boolean	TRUE si se encuentra un archivo de objeto host que cumpla el patrón de búsqueda indicado. El nombre completo de la ruta del archivo se almacena en Name.
FindNext(Name)	Parámetro = Name as string	Devuelve el nombre y la ruta del archivo de objeto host que cumpla el patrón de búsqueda indicado en la llamada original a FindFirst().
FindNext	Valor devuelto = Boolean	TRUE si se encuentra algún otro archivo de objeto host que cumpla el patrón indicado en la llamada a FindFirst(). El nombre completo de la ruta del archivo se almacena en Name.
RetrieveObject(Name)	Parámetro = Name as string	El nombre y la ruta del archivo de objeto host que se debe cargar.
RetrieveObject(AccessMode)	Parámetro = AccessMode as integer	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
RetrieveObject>Password)	Parámetro = Password as string	Contraseña del objeto. Puede tener el valor NULL.
RetrieveObject	Valor devuelto = Object	Objeto CHostData del archivo indicado.
RetrieveObjectEx(Name)	Parámetro = Name as string	El nombre y la ruta del archivo de objeto host que se debe cargar.

Método de CHostDataManager	Parámetro o valor devuelto	Descripción
RetrieveObjectEx(AccessMode)	Parámetro = AccessMode as integer	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
RetrieveObjectEx(Password)	Parámetro = Password as string	Contraseña del objeto. Puede tener el valor NULL.
RetrieveObjectEx	Valor devuelto = Object	Objeto CHostDataEx del archivo indicado
CreateObject(Name)	Parámetro = Name as string	Crea un objeto CHostData y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto host para el nuevo objeto.
CreateObject	Valor devuelto = Object	CHostData
CreateObjectEx(Name)	Parámetro = Name as string	Crea un objeto CHostDataEx y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto host para el nuevo objeto.
CreateObjectEx	Valor devuelto = Object	CHostDataEx
DeleteObject(Name)	Parámetro = Name as string	El nombre y la ruta del archivo de objeto host que se debe eliminar.
DeleteObject(Password)	Parámetro = Password as string	Contraseña del objeto.
DeleteObject	Valor devuelto = Boolean	TRUE si el objeto se ha eliminado.

Método de CHostDataManager	Parámetro o valor devuelto	Descripción
Launch(Name)	Parámetro = Name as string	Ejecuta un archivo de objeto host que se encargará de abrir la ventana del terminal host de pcAnywhere. El parámetro es el nombre y la ruta del archivo de objeto host que se iniciará.
Iniciar	Valor devuelto = Boolean	TRUE si el objeto se ejecuta correctamente.

Propiedades de CHostData

En la siguiente tabla, se describen las propiedades y los parámetros que están disponibles para el objeto CHostData.

En la [Tabla 2-17](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-17 Propiedades y parámetros de CHostData

Propiedad	Parámetro	Descripción
<CHostData>.PhoneNumber(String)	String	Define el número de teléfono del equipo host.
<CHostData>.UseDialingProperties(Bool)	Bool	Define las propiedades de marcado del sistema.
<CHostData>.RedialCount(Integer)	Integer	Define el número de intentos de rellamada antes de cancelarla.
<CHostData>.RedialDelay(Integer)	Integer	Define el número de segundos que hay que esperar entre los intentos de rellamada.
<CHostData>.Password(String)	String	Define la contraseña en el objeto host para su uso con la configuración de ExecuteProtection, ReadProtection y WriteProtection. Por motivos de seguridad, el servidor de automatización de pcAnywhere no ofrece la posibilidad de leer el valor de la contraseña. No se devuelve el valor de la contraseña.
<CHostData>.ExecuteProtection(Bool)	Bool	Establece que sea necesaria una contraseña para ejecutar el objeto. Definido por el parámetro Password.

Propiedad	Parámetro	Descripción
<CHostData>.ReadProtection(Bool)	Bool	Establece que sea necesaria una contraseña para ver las propiedades del objeto host. Definido por el parámetro Password.
<CHostData>.WriteProtection(Bool)	Bool	Establece que sea necesaria una contraseña para guardar los cambios en el objeto host. Definido por el parámetro Password.
<CHostData>.LogSession(Bool)	Bool	Activa y desactiva el registro de la sesión.
<CHostData>.RecordFile(String)	String	Define la ruta y el nombre completo correspondiente a la ubicación del archivo que registra la sesión activa.
<CHostData>.RecordSession(Bool)	Bool	Activa y desactiva el registro de sesión automático.
<CHostData>.ReadObject(String)	String	Define la contraseña del objeto, que se utilizará para actualizar la copia de datos locales del objeto host.
<CHostData>.WriteObject(String)	String	Define la contraseña del objeto, que se utilizará para crear el objeto host o para guardar los cambios que haya realizado en el objeto host.

En la siguiente tabla, se describen las propiedades de CHostData y los valores devueltos.

En la [Tabla 2-18](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-18 Propiedades y valores devueltos de CHostData

Propiedad	Valor devuelto	Descripción
String = <CHostData>.PhoneNumber	String	Devuelve el número de teléfono del equipo host.
Bool = <CHostData>.UseDialingProperties	Bool	Devuelve el uso de las propiedades de marcado de sistema que están definidas en el objeto host.
Integer = <CHostData>.RedialCount	Integer	Devuelve el número de intentos de rellamada que se haya definido en el objeto host.
Integer = <CHostData>.RedialDelay	Integer	Devuelve el número de segundos entre intentos de rellamada.

Propiedad	Valor devuelto	Descripción
Bool = <CHostData>.ExecuteProtection	Bool	Devuelve el valor de la configuración de ExecuteProtection.
Bool = <CHostData>.ReadProtection	Bool	Devuelve el valor de la configuración de ReadProtection.
Bool = <CHostData>.WriteProtection	Bool	Devuelve el valor de la configuración de WriteProtection.
Bool = <CHostData>.LogSession	Bool	Devuelve el valor TRUE si está activado el registro de la sesión.
String = <CHostData>.RecordFile	String	Devuelve la ruta y el nombre completo del archivo de registro de la sesión.
Bool = <CHostData>.RecordSession	Bool	Devuelve el valor de la configuración correspondiente al registro de sesiones.

Propiedades del tipo de conexión

En la siguiente tabla, se describen los parámetros y las propiedades del tipo de conexión.

En la [Tabla 2-19](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-19 Propiedades y parámetros del tipo de conexión

Propiedad	Parámetro	Descripción
<CHostData>.ConnectionType(String)	String	Define el tipo de conexión del host. El valor debe ser un tipo de conexión válido tal y como se haya definido por las funciones FirstConnectionType() y NextConnectionType(). Los siguientes son ejemplos de tipos de conexión válidos: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN mediante CAPI 2.0 ■ Nombre del módem (tal y como aparezca en el equipo)
<CHostData>.AssignConnection(String)	String	Establece el tipo de conexión que está definido en la cadena para activar el estado.
<CHostData>.UnassignConnection(String)	String	Establece el tipo de conexión que está definido en la cadena para desactivar el estado.

En la siguiente tabla, se describen las propiedades y los valores devueltos del tipo de conexión.

En la [Tabla 2-20](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-20 Propiedades y valores devueltos del tipo de conexión

Propiedad	Valor devuelto	Descripción
String = <CHostData>.ConnectionTypes	String	Devuelve el tipo de conexión del objeto host.

Propiedad	Valor devuelto	Descripción
Integer = <CHostData>.ConnectionType	Integer	Devuelve el número de tipos de conexión disponibles. Los siguientes son ejemplos de tipos de conexión válidos: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN mediante CAPI 2.0 ■ Nombre del módem (tal y como aparezca en el equipo)
String = <CHostData>.FirstConnectionType	String	Devuelve el primer tipo de conexión disponible.
String = <CHostData>.NextConnectionType	String	Devuelve el siguiente tipo de conexión disponible. Se denomina secuencial debido al número de tipos de conexión definido en <CHostData>.ConnectionTypes para enumerar todos los tipos de conexión.
Bool = <CHostData>.FindConnectionType(ConnectionType)	Bool	Devuelve TRUE si el tipo de conexión especificado se encuentra en la lista de los tipos de conexión disponibles.
Integer = <CHostData>.MaxAssignedConnections	Integer	Devuelve el número máximo de tipos de conexión que pueden estar activos en este host.
String = <CHostData>.FirstAssignedConnection	String	Devuelve el primer tipo de conexión activa asignada disponible.
String = <CHostData>.NextAssignedConnection	String	Devuelve el siguiente tipo de conexión activa asignada disponible.
Bool = <CHostData>.FindAssignedConnection(ConnectionType)	Bool	Devuelve TRUE si el tipo de conexión concuerda con cualquiera de los tipos de conexión activos.

Método de AssignConnection(ConnectionType)

El método de AssignConnection(Connection Type) coloca el tipo de conexión solicitado en la lista de objetos host correspondientes a los tipos de conexiones asignadas y lo transforma en el tipo de conexión actual cuando se procesan las siguientes llamadas de métodos específicos del dispositivo.

Si el tipo de conexión solicitado ya se encuentra en la lista de conexiones asignadas, esta lista no cambia. Sólo el tipo de conexión actual se cambia al tipo solicitado. Es normal llamar al método de AssignConnection varias veces sobre el mismo objeto mientras se están obteniendo y fijando los valores específicos de la conexión.

AssignConnection devuelve TRUE si el tipo de conexión ya existe en el equipo y se asigna correctamente o ya está asignado. Devuelve FALSE si el tipo de conexión solicitado no existe en el equipo o la cuenta de conexión asignada actualmente ya tiene el número máximo permitido de tipos de conexión.

Un objeto host de pcAnywhere puede admitir hasta dos tipos de conexión asignados. El método de AssignConnection devuelve FALSE si detecta cualquier intento de sobrepasar este límite.

En la siguiente tabla, se define el parámetro de AssignConnection:

Parámetro	Descripción
ConnectionType as string	El nombre de un tipo de dispositivo de conexión que se debe asignar.

En la siguiente tabla, se define el valor devuelto de AssignConnection:

Valor devuelto	Descripción
Booleano	TRUE si el tipo de dispositivo está disponible y el número de conexiones máximas permitidas no se ha alcanzado aún.

Método de UnassignConnection(ConnectionType)

El método de UnassignConnection(Connection Type) devuelve TRUE si el tipo de conexión se ha eliminado correctamente de la lista de tipos de conexiones asignados.

En la siguiente tabla, se define el parámetro de UnassignConnection:

Parámetro	Descripción
ConnectionType as string	El nombre de un tipo de dispositivo de conexión que debe dejar de estar asignado.

En la siguiente tabla, se define el valor devuelto de UnassignConnection:

Valor devuelto	Descripción
Booleano	TRUE si se elimina con éxito la asignación del tipo de dispositivo.

Propiedades de marcado

En la siguiente tabla, se definen las propiedades y los valores devueltos para configurar las propiedades de marcado para las conexiones por módem.

En la [Tabla 2-21](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-21 Propiedades y parámetros de las propiedades de marcado del módem

Propiedad	Parámetro	Descripción
<CHostData>.AreaCode(String)	String	Define el código de área de las conexiones por módem.
<CHostData>.CountryCode(String)	String	Define el código del país para las conexiones por módem.

En la siguiente tabla, se definen las propiedades y los valores devueltos de las propiedades de marcado del módem.

En la [Tabla 2-22](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-22 Propiedades y valores devueltos de las propiedades de marcado del módem

Propiedad	Valor devuelto	Descripción
String = <CHostData>.AreaCode	String	Devuelve el código de área.
String = <CHostData>.CountryCode	String	Devuelve el código del país.
Integer = <CHostData>.CountryCodes	Integer	Devuelve el número de códigos de países disponibles.
String = <CHostData>.FirstCountryCode	String	Devuelve el primer código de país disponible que aparece en el sistema operativo.

Propiedad	Valor devuelto	Descripción
String = <CHostData>.NextCountryCode	String	Devuelve el siguiente código de país disponible. Se lo invoca de forma secuencial de acuerdo con el número de códigos de países definido en <CHostData>.CountryCodes para enumerar todos los códigos de países.

Propiedades de un dispositivo COM

En la siguiente tabla, se definen las propiedades y los parámetros que permiten personalizar la configuración del puerto para el módem y otras conexiones basadas en COM.

En la [Tabla 2-23](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-23 Propiedades y parámetros del dispositivo COM

Propiedad	Parámetro	Descripción
<CHostData>.ComParity(String)	String	Define la paridad de las comunicaciones. Los siguientes valores son válidos: ■ <Ninguna> ■ Impar ■ Par ■ Marca ■ Espacio
<CHostData>.ComFlowControl(String)	String	Define el control de flujo de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ <Ninguno> ■ XONXOFF ■ RTS/CTS ■ Ambos

Propiedad	Parámetro	Descripción
<CHostData>.ComStartedBy(String)	String	Define la configuración inicial de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI) ■ Recibir 2 <CR> ■ Respuesta del módem
<CHostData>.ComEndedBy(String)	String	Define la configuración final de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI)
<CHostData>.ComSpeed(Long)	Long	Define la configuración de velocidad COM máxima. Los siguientes valores son válidos: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

En la siguiente tabla, se describen las propiedades y los valores devueltos del dispositivo COM.

En la [Tabla 2-24](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-24 Propiedades y valores devueltos del dispositivo COM

Propiedad	Valor devuelto	Descripción
String = <CHostData>.ComParity	String	Devuelve uno de los siguientes valores para la paridad de las comunicaciones: ■ <Ninguna> ■ Impar ■ Par ■ Marca ■ Espacio
String = <CHostData>.ComFlowControl	String	Devuelve la configuración de Com Flow correspondiente al objeto host. Los siguientes valores son válidos: ■ <Ninguno> ■ XONXOFF ■ RTS/CTS ■ Ambos
String = <CHostData>.ComStartedBy	String	Devuelve el control Com Start de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI) ■ Recibir 2 <CR> ■ Respuesta del módem
String = <CHostData>.ComEndedBy	String	Devuelve el control Com End de los tipos de conexión basados en COM. Los siguientes valores son válidos: ■ Siempre conectado ■ Detección de portadora (DCD) ■ Preparado para enviar (CTS) ■ Datos preparados (DSR) ■ Indicador de timbre (RI)

Propiedad	Valor devuelto	Descripción
Long = <CHostData>.ComSpeed	Long	Devuelve la configuración actual de Com Speed correspondiente al objeto host. Los siguientes valores son válidos: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Propiedades de un dispositivo NetBIOS

En la siguiente tabla, se definen las propiedades y el parámetro para los dispositivos de red NetBIOS.

En la [Tabla 2-25](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-25 Propiedad y parámetro de NetBIOS

Propiedad	Parámetro	Descripción
<CHostData>.LanaNumber(Integer)	Integer	Define el número del adaptador LAN (LANA) para las conexiones de NetBIOS.

En la siguiente tabla, se definen la propiedad y el valor devuelto para los dispositivos de red NetBIOS.

En la [Tabla 2-26](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-26 Valores devueltos de NetBIOS

Propiedad	Valor devuelto	Descripción
Integer = <CHostData>.LanaNumber	Integer	Devuelve la configuración actual del número del adaptador LAN (LANA) para las conexiones de NetBIOS.

Propiedades del dispositivo ISDN mediante CAPI 2.0

En la siguiente tabla, se definen las propiedades y los parámetros para las conexiones europeas ISDN.

En la [Tabla 2-27](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-27 Propiedades y parámetros de ISDN mediante CAPI 2.0

Propiedad	Parámetro	Descripción
<CHostData>.CapiChannelBonding(Bool)	Bool	Activa o desactiva el enlace de canales para los dispositivos ISDN CAPI.
<CHostData>.CapiExtensions(String)	String	Define todas las extensiones adicionales CAPI necesarias para las comunicaciones.

En la [Tabla 2-28](#), se definen las propiedades y los valores devueltos para las conexiones ISDN europeas.

Tabla 2-28 Propiedades y valores devueltos de ISDN mediante CAPI 2.0

Propiedad	Valor devuelto	Descripción
Bool= <CHostData>.CapiChannelBonding	Bool	Devuelve la configuración del enlace de canales actual de ISDN CAPI en el objeto host.
String = <CHostData>.CapiExtensions	String	Devuelve la lista actual de extensiones CAPI del objeto host.

Objeto CHostDataEx

El objeto CHostDataEx contiene las mismas funciones que el objeto CHostData con algunas funciones adicionales.

En la siguiente tabla, se describe las propiedades y los parámetros.

En la [Tabla 2-29](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-29 Propiedades y parámetros de CHostDataEx

Propiedad	Parámetro	Descripción
<CHostData>.CryptPrivateKey(String)	String	Define el nombre del contenedor de clave privada que desee utilizar.

Propiedad	Parámetro	Descripción
<CHostData>.CryptCommonName(String)	String	Define el nombre común de la clave privada que desee utilizar.
<CHostData>.CryptReqLevel(Byte)	Byte	Define el nivel de cifrado. Los siguientes valores son válidos: ■ -1: Ninguno ■ 0: pcAnywhere ■ 1: Simétrico ■ 2: Clave pública
<CHostData>.CryptRefuseLower(Bool)	Bool	Define si el equipo host acepta conexiones procedentes de un equipo remoto que utiliza un nivel inferior de cifrado.
<CHostData>.CallersPath(String)	String	Define la ruta de acceso completa de los archivos del comunicante.
<CHostData>.ConfirmConnect(Bool)	Bool	Define si se le solicitará al usuario host confirmar las conexiones.
<CHostData>.ConfirmTimeout(Byte)	Byte	Define el número de segundos antes de solicitar la confirmación de que se agota el límite de tiempo de conexión.
<CHostData>.ConfirmDeny(Bool)	Bool	Define si la conexión debería finalizar si la solicitud para confirmar el límite del tiempo de conexión se agota.
<CHostData>.PwCaseSensitive(Bool)	Bool	Obliga al uso de contraseñas que distingan entre mayúsculas y minúsculas.
<CHostData>.PwAttempts(Byte)	Byte	Define el número de intentos de inicio de sesión fallidos consecutivos que están permitidos antes de finalizar la conexión.
<CHostData>.PwTimeout(Byte)	Byte	Define el número de minutos que un usuario tiene para completar el inicio de sesión antes de que finalice la conexión.

Propiedad	Parámetro	Descripción
<CHostData>.ActiveKbds(Byte)	Byte	Define qué mouse y teclado estarán activos durante la conexión. Los siguientes valores son válidos: ■ 0: Host y remoto ■ 1: Host ■ 2: Remoto
<CHostData>.InactiveTimeout(Byte)	Byte	Define el número de minutos en los que el teclado y el mouse pueden estar inactivos antes de que finalice la conexión.
<CHostData>.LockSystemWhileWait(Bool)	Bool	Define el equipo host bloqueado al iniciar la configuración.
<CHostData>.MinimizeOnLaunch(Bool)	Bool	Define la ejecución de la opción de inicio del host minimizado.
<CHostData>.RunAsService(Bool)	Bool	Activa el host para ejecutarlo como un servicio.
<CHostData>.ConnLostWait(Byte)	Byte	Define el número de minutos que tendrá que esperar antes de permitir que otro comunicante se conecte.
<CHostData>.ConnLostHostOpts(Bool)	Bool	Define si esperar o no otra conexión, o cancelar el host si la sesión finaliza de forma incorrecta. Los siguientes valores son válidos: ■ FALSE: esperar ■ TRUE: cancelar host
<CHostData>.EnableConnLostSecurity(Bool)	Bool	Activa o desactiva el final de las opciones de seguridad para las sesiones que finalicen de forma incorrecta.

Propiedad	Parámetro	Descripción
<CHostData>.AuthenticationType(Byte)	Byte	Define el tipo de autenticación. Los siguientes valores son válidos: ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS (Servicios Active Directory) ■ 7: LDAP de Microsoft ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID La autenticación de pcAnywhere se utiliza de manera predeterminada si no se ha definido ningún valor de autenticación o si el valor definido no es válido (por ejemplo, el tipo de autenticación no está disponible).
<CHostData>.ConnLostSecurity(Byte)	Byte	Define las opciones de seguridad para manejar un final de sesión incorrecto. Los siguientes valores son válidos: ■ 1: Cerrar sesión del usuario ■ 2: Reiniciar el equipo host ■ 3: Bloquear el equipo
<CHostData>.CallbkDelay(Byte)	Byte	Define el número de segundos que tendrá que esperar antes de que el módem host llame de nuevo al equipo remoto.
<CHostData>.EndSessHostOpts(Bool)	Bool	Define si el host espera otra conexión o si se ha cancelado después de un final normal de la sesión. Los siguientes valores son válidos: ■ FALSE: esperar la siguiente conexión ■ TRUE: cancelar host
<CHostData>.EnableEndSessSecurity(Bool)	Bool	Activa o desactiva las opciones de seguridad para una finalización de sesión normal.

Propiedad	Parámetro	Descripción
<CHostData>.EndSessSecurity(Byte)	Byte	Define las opciones de seguridad para una finalización de sesión normal. Los siguientes valores son válidos: ■ 1: Cerrar sesión del usuario ■ 2: Reiniciar el equipo host ■ 3: Bloquear el equipo
<CHostData>.BlankHost(Bool)	Bool	Apaga la pantalla del equipo host. Algunas tarjetas de video no admiten esta opción.
<CHostData>.AllowRemoteMouse(Bool)	Bool	Desactiva el uso del mouse en el equipo remoto durante una sesión.
<CHostData>.RebootOnDisconnect(Bool)	Bool	Obliga a reiniciar el equipo después de cada finalización de sesión si el valor es TRUE.
<CHostData>.PasswordAfterDisc(Bool)	Bool	Desconecta al usuario después de finalizar la sesión.
<CHostData>.LogFailures(Bool)	Bool	Registra los intentos fallidos de introducción de la contraseña.
<CHostData>.AllowDriveSecurity(Bool)	Bool	Activa las opciones de seguridad de la unidad. Esta configuración sólo es válida en sistemas de archivos NTFS.
<CHostData>.UseDirectoryServices(Bool)	Bool	Activa el uso de los servicios de directorios para la autenticación.
<CHostData>.DirectoryServiceEntry(String)	String	Define la configuración de los servicios de directorios.

En la tabla siguiente, se describen las propiedades y los valores devueltos para el objeto CHostDataEx.

En la [Tabla 2-30](#), sustituya la información entre los corchetes angulares por los valores reales.

Tabla 2-30 Propiedades y valores devueltos de CHostDataEx

Propiedad	Valor devuelto	Descripción
String = <CHostDataEx>.CryptPrivateKey	String	Devuelve el contenedor de clave privada activo en ese momento.

Propiedad	Valor devuelto	Descripción
String = <CHostDataEx>.CryptCommonName	String	Devuelve el nombre común del contenedor de clave privada activo.
Byte = <CHostDataEx>.CryptReqLevel	Byte	Devuelve uno de los siguientes valores de cifrado: ■ -1: Ninguno ■ 0: pcAnywhere ■ 1: Simétrico ■ 2: Clave pública
Bool = <CHostDataEx>.CryptRefuseLower	Bool	Devuelve el valor de la configuración para denegar el cifrado más bajo.
String = <CHostDataEx>.CallersPath	String	Devuelve la ruta de acceso activa actualmente a los archivos del comunicante.
Bool = <CHostDataEx>.ConfirmConnect	Bool	Devuelve el valor de la configuración de confirmación de conexión.
Byte = <CHostDataEx>.ConfirmTimeout	Byte	Devuelve el valor de la configuración para denegar el cifrado más bajo.
Bool = <CHostDataEx>.ConfirmDeny	Bool	Devuelve el valor de la desconexión si se ha agotado el tiempo de espera.
Bool = <CHostDataEx>.PwCaseSensitive	Bool	Devuelve el valor de la configuración para que las contraseñas distingan entre mayúsculas y minúsculas.
Byte = <CHostDataEx>.PwAttempts	Byte	Devuelve el valor de la configuración del límite de intentos de inicio de sesión por llamada.
Byte = <CHostDataEx>.PwTimeout	Byte	Devuelve el valor de la configuración del tiempo límite para completar el inicio de sesión (en minutos).
Byte = <CHostDataEx>.ActiveKbds	Byte	Devuelve la configuración del mouse y del teclado activo. Los siguientes valores son válidos: ■ 0: Host y remoto ■ 1: Host ■ 2: Remoto

Propiedad	Valor devuelto	Descripción
Byte = <CHostDataEx>.InactiveTimeout	Byte	Devuelve el número de minutos que tendrá que esperar antes de desconectar si el límite de tiempo de inactividad se agota.
Bool = <CHostDataEx>.LockSystemWhileWait	Bool	Devuelve el equipo host bloqueado al iniciar la configuración.
Bool = <CHostDataEx>.MinimizeOnLaunch	Bool	Devuelve la configuración de inicio del host minimizado al inicio.
Bool = <CHostDataEx>.RunAsService	Bool	Devuelve el host en ejecución como un valor de servicio.
Byte = <CHostDataEx>.ConnLostWait	Byte	Devuelve el número de minutos que tendrá que esperar antes de permitir otra conexión.
Bool = <CHostDataEx>.ConnLostHostOpts	Bool	Devuelve si esperar o no otra conexión o cancelar el host si la sesión finaliza de una forma incorrecta. Los siguientes valores son válidos: ■ FALSE: esperar ■ TRUE: cancelar host
Bool = <CHostDataEx>.ConnLostWait	Bool	Devuelve el valor de la opción de seguridad que está establecida sobre cómo tratar el final incorrecto de la sesión.

Propiedad	Valor devuelto	Descripción
Byte = <CHostDataEx>.AuthenticationType	Byte	Devuelve la referencia numérica del tipo de autenticación. Los siguientes valores son válidos: ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS (Servicios Active Directory) ■ 7: LDAP de Microsoft ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID La autenticación de pcAnywhere se utiliza de manera predeterminada si no se ha definido ningún valor de autenticación o si el valor definido no es válido (por ejemplo, el tipo de autenticación no está disponible).
Byte = <CHostDataEx>.ConnLostSecurity	Byte	Devuelve la representación numérica del nivel de seguridad que está establecida sobre cómo tratar el final incorrecto de la sesión. Los siguientes valores son válidos: ■ 1: Cerrar sesión del usuario ■ 2: Reiniciar el equipo host ■ 3: Bloquear el equipo
Byte = <CHostDataEx>.CallbkDelay	Byte	Devuelve el número de segundos que tendrá que esperar antes de que el módem host llame de nuevo al equipo remoto.
Bool = <CHostDataEx>.EndSessHostOpts	Bool	Devuelve si el host espera otra conexión o si se ha cancelado después de un final normal de la sesión. Los siguientes valores son válidos: ■ FALSE: esperar la siguiente conexión ■ TRUE: cancelar host

Propiedad	Valor devuelto	Descripción
Bool = <CHostDataEx>.EnableEndSessSecurity	Bool	Devuelve si el final de las opciones de seguridad de la sesión están activas para un final normal de la sesión.
Byte = <CHostDataEx>.EndSessSecurity	Byte	Devuelve la opción de seguridad definida para un final normal de la sesión. Los siguientes valores son válidos: ■ 1: Cerrar sesión del usuario ■ 2: Reiniciar el equipo host ■ 3: Bloquear el equipo
Bool = <CHostDataEx>.BlankHost	Bool	Devuelve la opción de apagar la pantalla que haya definido.
Bool = <CHostDataEx>.AllowRemoteMouse	Bool	Devuelve si el usuario remoto cuenta con el control del teclado y el mouse durante una sesión.
Bool = <CHostDataEx>.RebootOnDisconnect	Bool	Devuelve si el equipo host se ha reiniciado después de que la sesión finalizara.
Bool = <CHostDataEx>.PasswordAfterDisc	Bool	Devuelve si el usuario host ha cerrado la sesión después de que la sesión finalizara.
Bool = <CHostDataEx>.LogFailures	Bool	Devuelve si la opción de registro de los fallos en la introducción de contraseñas está definida.
Bool = <CHostDataEx>.AllowDriveSecurity	Bool	Devuelve si la opción de seguridad de la unidad está habilitada.
Bool = <CHostDataEx>.UseDirectoryServices	Bool	Devuelve si la opción de los servicios de directorio está habilitada.
String = <CHostDataEx>.DirectoryServiceEntry	String	Devuelve la configuración del servicio de directorio.

Código de muestra de Visual Basic para las funciones de host

El siguiente código de muestra de Visual Basic recupera un objeto de datos host y modifica sus propiedades:

```
Private Sub Command1_Click()  
Dim HostDataManager as Object  
Dim HostData as Object
```

```
Dim s as string

'Crea un objeto CHostDataManager
Set HostDataManager = CreateObject(WINAWSVR.BEHOSTDATAMANAGER)

'muestra y cambia el directorio actual
s = HostDataManager.CurrentDirectory()
MsgBox (s)
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
s = HostDataManager.CurrentDirectory()
MsgBox (s)

'obtiene un objeto de datos remoto
Set HostData = HostDataManager.RetrieveObject("pod.BHF", 2, 0)

'muestra algunas propiedades
s = HostData.AreaCode()
MsgBox (s)
s = HostData.PhoneNumber()
MsgBox (s)
'establece algunas propiedades
RemoteData.HostData = "212"
RemoteData.HostData = "555-5555"

'escribe objetos en el disco
HostData.WriteObject (0)
End Sub
```

Utilice los métodos FindFirst y FindNext para mostrar el archivo host en un directorio de esta manera:

```
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")
RemoteDataManager.FindFirst "*", s
MsgBox (s)
HostDataManager.FindNext s
MsgBox (s)

End Sub
```

Funciones de Awrem32

Las funciones de Awrem32 ofrecen los parámetros y valores devueltos para gestionar las conexiones entre el equipo host y remoto.

Función de Awrem32	Parámetro o valor devuelto	Descripción
awConnect(FileName)	Parámetro = Name as string	El nombre y la ruta del archivo .chf que contiene la información acerca del equipo host.
awConnect	Valor devuelto = Boolean	TRUE si se ejecuta el comando.
awDisconnect()	Valor devuelto = Boolean	Después de llamar a esta función, el programa que ejecuta la llamada debe eliminar el objeto (C++ - delete IAwrem32X*, VB - set ObjectName = Nothing;).
FileXferFromHost(HostFile)	Parámetro = HostFile as string	Contiene el nombre de archivo y la ruta completos que se van a copiar desde el equipo host.
FileXferFromHost(RemoteFile)	Parámetro = RemoteFile as string	Contiene el nombre de archivo y la ruta completos. No es necesario que las cadenas HostFile y RemoteFile sean idénticas.
FileXferFromHost	Valor devuelto = Boolean	TRUE si se ejecuta el comando.

Función de Awrem32	Parámetro o valor devuelto	Descripción
FileXferToHost(HostFile)	Parámetro = HostFile as string	Copia un archivo desde el equipo remoto en el equipo host. Los parámetros pueden contener caracteres comodín. El parámetro contiene la ruta de destino y el nombre de archivo completos.
FileXferToHost(RemoteFile)	Parámetro = RemoteFile as string	Contiene el nombre de archivo y la ruta completos que se van a copiar desde el equipo remoto. No es necesario que las cadenas HostFile y RemoteFile sean idénticas.
CreateFolderOnHost(FolderName)	Parámetro = FolderName as string	Contiene la unidad y la ruta donde se debe crear la carpeta en el equipo host.
CreateFolderOnHost	Valor devuelto = Boolean	TRUE si se ejecuta el comando.
ExecuteHostFile(FileName)	Parámetro = FileName as string	Ejecuta un archivo existente en el equipo host. Esta función sólo permite ejecutar archivos por lotes, de comandos y ejecutables. No permite ejecutar archivos asociados con ejecutables. Por ejemplo, esta función no abre Microsoft Word si ejecuta un archivo .doc. El parámetro contiene la ruta completa al archivo en el equipo host.
ExecuteHostFile	Valor devuelto = Boolean	TRUE si se ejecuta el comando.
GetError()	Valor devuelto = String	Devuelve el último error generado en Awrem32.
ConnectionStatus()	Valor devuelto = Short	Devuelve el estado actual de la conexión al equipo host. Entre los valores posibles se incluyen los siguientes: ■ -1 = Conexión perdida ■ 0 = Sin conexión ■ 1 = Sesión conectada

Definiciones de objetos de Visual C++

En este capítulo se incluyen los temas siguientes:

- [Acerca de los objetos de Visual C++](#)
- [Métodos de CRemoteDataManager](#)
- [Objeto CRemoteData](#)
- [Objeto CRemoteDataEx](#)
- [Métodos de CHostDataManager](#)
- [Objeto CHostData](#)
- [Objeto CHostDataEx](#)
- [funciones de Awrem32](#)

Acerca de los objetos de Visual C++

El servidor de automatización de pcAnywhere ofrece los siguientes componentes para admitir la automatización OLE:

Winawsvr	Ofrece la información necesaria para conectarse con el servidor de automatización de pcAnywhere y tener acceso a sus interfaces.
Awrem32	Ofrece la información necesaria para controlar las conexiones de pcAnywhere.

Los objetos que forman Winawsvr se describen en la [Tabla 3-1](#).

Tabla 3-1 Objetos de Winawsvr

Object	Descripción	Referencia
CRemoteDataManager	Ofrece los métodos para crear, abrir, modificar, guardar y eliminar los objetos CRemoteData.	Ver " Métodos de CRemoteDataManager " en la página 62.
CRemoteData	Define los parámetros para tener acceso a las funciones remotas de pcAnywhere y controlarlas.	Ver " Objeto CRemoteData " en la página 66. Ver " Objeto CRemoteDataEx " en la página 72.
CHostDataManager	Ofrece los métodos para crear, abrir, modificar, guardar y eliminar los objetos CHostData.	Ver " Métodos de CHostDataManager " en la página 74.
CHostData	Define los parámetros para tener acceso a las funciones de host de pcAnywhere y controlarlas.	Ver " Objeto CRemoteData " en la página 66. Ver " Objeto CRemoteDataEx " en la página 72.

Awrem32 cuenta con un objeto, que está compuesto de ocho interfaces que admiten las sesiones de control remoto y de transferencia de archivos.

Ver "[funciones de Awrem32](#)" en la página 88.

Algunas funciones ya no se admiten. Sin embargo, las definiciones de objeto se proporcionan para utilizarlas con versiones anteriores.

Para las funciones que requieran el uso de contraseñas, será posible configurar sus valores pero no recuperarlos. Esto se debe a motivos de seguridad.

Métodos de CRemoteDataManager

Los métodos de CRemoteDataManager ofrecen parámetros y valores devueltos para tener acceso a los objetos CRemoteData y controlarlos.

Método	Parámetro o valor devuelto	Descripción
BSTR CurrentDirectory();	Parámetro = BSTR	El nombre completo de la ruta del directorio de datos actual de pcAnywhere, en el cual se almacenan los objetos remotos de pcAnywhere

Método	Parámetro o valor devuelto	Descripción
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Parámetro = LPCTSTR lpszNewDirectory Valor devuelto = BOOL	Cambia el directorio actual en el que se almacenan los objetos remotos de pcAnywhere. El parámetro es el nombre de un directorio existente. El valor devuelto es TRUE si es correcto.
BOOL FindFirst(LPCTSTR lpszPattern, BSTR FAR*);	Parámetro = LPCTSTR lpszPattern	Patrón de búsqueda de nombre de archivo para filtrar los archivos de objetos (un asterisco [*] busca todos los archivos del directorio actual).
BOOL FindFirst(pbstrFullQualName);	Parámetro = BSTR FAR * pbstrFullQualName	Devuelve el nombre y la ruta del archivo de objeto remoto que cumple el patrón especificado.
BOOL FindFirst	Valor devuelto = BOOL	TRUE si se encuentra un archivo de objeto remoto que cumpla el patrón de búsqueda indicado. El nombre completo de la ruta del archivo coincidente se almacena en pbstrFullQualName.
BOOL FindNext(BSTR FAR* pbstrFullQualName);	Parámetro = BSTR FAR * pbstrFullQualName Valor devuelto = BOOL	Cuando FindFirst() se ha llamado correctamente para obtener el nombre de un archivo de objeto remoto del directorio actual, se puede llamar a FindNext() para encontrar el siguiente archivo que cumpla el patrón, si existe alguno. El valor devuelto es TRUE si se encuentra otro archivo de objeto remoto que cumpla el patrón indicado en la llamada a FindFirst(). El nombre completo de la ruta del archivo coincidente se almacena en pbstrFullQualName.
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName	Obtiene un objeto de CRemoteData según el nombre de archivo. El parámetro es el nombre y la ruta del archivo de objeto remoto que se debe cargar.

Método	Parámetro o valor devuelto	Descripción
LPDISPATCH RetrieveObject(short wAccessMode);	Parámetro = short wAccessMode	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword	Contraseña del objeto. Puede tener el valor NULL.
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName	Obtiene un objeto CRemoteDataEx según el nombre de archivo. El parámetro es el nombre y la ruta del archivo de objeto remoto que se debe cargar.
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parámetro = short wAccessMode	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword	Contraseña del objeto. Puede tener el valor NULL.
LPDISPATCH RetrieveObjectEx	Valor devuelto = LPDISPATCH	Puntero a un objeto OLE dual. El objeto es un objeto CRemoteDataEx. Consulte Código de muestra de Visual C++ para las funciones remotas.

Método	Parámetro o valor devuelto	Descripción
LPDISPATCH CreateObject(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName Valor devuelto = LPDISPATCH	Crea un objeto CRemoteData y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto remoto para el nuevo objeto. El valor devuelto es un puntero a un objeto OLE dual. El objeto es un objeto CRemoteData. Consulte Código de muestra de Visual C++ para las funciones remotas.
LPDISPATCH CreateObjectEx(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName Valor devuelto = LPDISPATCH	Crea un objeto CRemoteDataEx y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto remoto para el nuevo objeto. El valor devuelto es un puntero a un objeto OLE dual. El objeto es un objeto CRemoteDataEx. Consulte Código de muestra de Visual C++ para las funciones remotas.
BOOL DeleteObject(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName	Elimina un archivo de objeto remoto. El parámetro es el nombre y la ruta del archivo de objeto remoto que se debe eliminar.
BOOL DeleteObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword	Contraseña del objeto. Puede tener el valor NULL.
BOOL DeleteObject	BOOL	TRUE si el objeto se ha eliminado.

Método	Parámetro o valor devuelto	Descripción
BOOL Launch(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName	Ejecuta un archivo de objeto remoto que se encargará de abrir la ventana del terminal remoto de pcAnywhere.
	Valor devuelto = BOOL	El parámetro es el nombre y la ruta del archivo de objeto remoto que se debe iniciar.
		El valor devuelto es TRUE si el objeto se inicia correctamente.

Objeto CRemoteData

Utilice este objeto para modificar los datos de un objeto remoto.

Métodos Get y Set

Los siguientes métodos se utilizan para obtener y configurar las propiedades del objeto CRemoteData.

El nombre del equipo es el nombre del host de pcAnywhere al que se debe llamar cuando se inicia el objeto remoto.

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

El número de teléfono es el número que se debe marcar para establecer una conexión por módem con un host de pcAnywhere.

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

Indica si se deben utilizar las propiedades de marcado TAPI (información de la ubicación) (TRUE), o bien, si la cadena del número de teléfono debe utilizarse exactamente como aparece (FALSE).

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

Si se van a utilizar las propiedades de marcado, éste es el código de área del número al que se debe llamar.

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

Si se van a utilizar las propiedades de marcado, éste es el código de país del número al que se debe llamar.

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

La cantidad de veces que se debe intentar marcar este número si falla la llamada.

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

El tiempo de espera (en segundos) entre los intentos de llamada.

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

El nombre que se debe enviar al host para iniciar la sesión cuando se establece una conexión. Si este campo se deja en blanco, al usuario se le solicitará un nombre de registro al iniciar la conexión.

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

La contraseña que se debe enviar al host para registrarse cuando se establece una conexión. Si este campo se deja en blanco, al usuario se le solicitará una contraseña al iniciar la conexión.

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

La contraseña para este objeto.

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

El objeto sólo se podrá ejecutar si se utiliza la contraseña (TRUE).

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

El objeto sólo se podrá ver si se introduce la contraseña correcta (TRUE).

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

El objeto sólo se podrá escribir si se introduce la contraseña correcta (TRUE).

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

Determina si en las sesiones en las que se utiliza este objeto se va a efectuar el registro.

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

Determina si las sesiones que utilizan este objeto se grabarán desde el comienzo.

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

El nombre del archivo de grabación para las sesiones que utilicen este objeto.

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

Define los tipos de conexión.

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

Los siguientes tipos de conexión están disponibles:

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

También puede utilizarse como tipo de conexión el nombre de un dispositivo TAPI. DEFAULT TAPI utiliza el primer dispositivo TAPI que se encuentre en el sistema. Para utilizar un dispositivo TAPI específico, utilice FirstConnectionType() / NextConnectionType() para buscar los dispositivos disponibles.

Métodos de detalles de objetos remotos

Cuando a un objeto remoto se le asigna un tipo de conexión, los detalles del dispositivo se configuran en los valores válidos predeterminados.

Los siguientes tipos de conexión cuentan con opciones de configuración avanzadas que pueden definirse en la aplicación:

- Dispositivos COM
- Dispositivos de NetBIOS
- Dispositivos ISDN mediante CAPI 2.0 (sólo para ISDN europea)

Detalles del dispositivo COM

Define el nivel de paridad de las comunicaciones.

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

Los valores de paridad de las comunicaciones son los siguientes:

- Ninguna
- Impar
- Par
- Marca
- Espacio

Define el nivel de control de flujo.

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

Los valores de control de flujo son los siguientes:

- <Ninguno>
- XONXOFF
- RTS/CTS
- AMBOS

Define el ajuste de inicio de conexión.

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

Los valores de inicio de conexión son los siguientes:

- Siempre conectado
- Detección de portadora (DCD)

- Preparado para enviar (CTS)
- Datos preparados (DSR)
- Indicador de timbre (RI)
- Recibir 2 <CR>
- Respuesta del módem

Define los valores de fin de conexión.

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

Los valores de fin de conexión son los siguientes:

- Siempre conectado
- Detección de portadora (DCD)
- Preparado para enviar (CTS)
- Datos preparados (DSR)
- Indicador de timbre (RI)

Define la velocidad de la conexión.

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

Los valores de la velocidad de conexión son los siguientes:

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

Detalles del dispositivo de NetBIOS

Define el número de adaptador LAN (LANA) que se debe utilizar para esta conexión.

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

Detalles del dispositivo ISDN mediante CAPI 2.0

Activa el enlace de canales (utiliza dos canales ISDN para una conexión) si el valor es TRUE.

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

Define todas las extensiones adicionales CAPI necesarias para las comunicaciones.

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

Métodos de objetos remotos

En la siguiente tabla, se enumeran los métodos normales del objeto remoto (no se utilizan para obtener y definir las propiedades):

Método	Parámetro o valor devuelto	Descripción
short ConnectionTypes();	Valor devuelto = Short	La cantidad de tipos de conexión disponibles actualmente en este equipo.
BSTR FirstConnectionType(); y BSTR NextConnectionType();	Valor devuelto = BSTR	FirstConnectionType() y NextConnectionType() se utilizan para explorar los tipos de conexión disponibles. Las funciones devuelven una cadena BSTR, que es el nombre de un tipo de conexión disponible. Estos tipos de conexión devueltos se pueden utilizar con la función SetConnectionType(). El valor devuelto es el nombre de un tipo de dispositivo de conexión admitido.

Método	Parámetro o valor devuelto	Descripción
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Parámetro = LPCTSTR lpszConnectionTyp Valor devuelto = BOOL	El nombre de un tipo de dispositivo de conexión. El valor devuelto es TRUE si el tipo de conexión ya existe en el equipo.
short CountryCodes();	Valor devuelto = Short	Devuelve el número de códigos de países disponibles.
BSTR FirstCountryCode(); y BSTR NextCountryCode();	Valor devuelto = BSTR	FirstCountryCode() y NextCountryCode() se utilizan para explorar los códigos de países disponibles. Las funciones devuelven una cadena BSTR, que es el nombre de un código de país disponible. Estos valores devueltos se pueden utilizar con la función SetCountryCode(). El valor devuelto es la primera o siguiente cadena del código de país.
BOOL ReadObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword Valor devuelto = BOOL	Lee los datos de objetos de los archivos de objetos remotos. El parámetro es la contraseña del objeto. El valor devuelto es TRUE si el objeto se lee correctamente.
BOOL WriteObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword Valor devuelto = BOOL	Escribe los datos de objetos en el archivo de objeto remoto. El parámetro es la contraseña del objeto. El valor devuelto es TRUE si el objeto se escribe correctamente.

Objeto CRemoteDataEx

El objeto CRemoteDataEx contiene las mismas funciones que el objeto CRemoteData con los siguientes métodos Get y Set adicionales:

```
BSTR GetPrivateKey(); // Devuelve la información de la clave privada
void SetPrivateKey(LPCTSTR lpszNewValue);
BSTR GetCertificationName(); // Devuelve el nombre de la certificación
void SetCertificationName(LPCTSTR lpszNewValue);
short GetEncryptionLevel(); // Devuelve el valor del nivel de cifrado
```

```
void SetEncryptionLevel(short nNewValue);  
BOOL GetDenyLowerEncrypt(); // Devuelve el valor de DenyLowerEncrypt  
void SetDenyLowerEncrypt(BOOL bNewValue);  
BSTR GetAutoDomain(); // Devuelve el valor de AutoDomain  
void SetAutoDomain(LPCTSTR lpszNewValue);
```

Código de muestra de Visual C++ para las funciones remotas

La siguiente función de muestra de C++ crea un objeto remoto, establece su tipo de conexión en TCP/IP, establece el nombre del equipo en la dirección TCP/IP introducida en la función e inicia el objeto remoto.

```
BOOL LaunchTCPRemote(LPCTSTR lpszAddress)  
{  
    BOOL bReturn = FALSE;  
  
    CRemoteDataManager remoteDM;  
  
    CRemoteData remoteData;  
  
    // Primero se crea CRemoteDataManager  
    remoteDM.CreateDispatch( _T( "WINAWSVR.RemoteDataManager" ) );  
    // Después se crea CRemoteData y se lo adjunta  
    remoteData.AttachDispatch( remoteDM.CreateObject("Prueba", 0) );  
  
    // Por último, se configuran las propiedades necesarias  
    remoteData.SetConnectionType("TCP/IP");  
  
    remoteData.SetComputerName(lpszAddress);  
  
    // Se guardan los datos del objeto  
    if (remoteData.WriteObject(0))  
    {  
        // Y se inicia  
        if (remoteData.Launch())  
            bReturn = TRUE;  
    }  
    // Se libera el objeto remoto.  
    remoteData.ReleaseDispatch();
```

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

return bReturn;
}
```

Métodos de CHostDataManager

En la tabla siguiente, se describen los métodos de CHostDataManager, incluso los parámetros y los valores devueltos para acceder y controlar los objetos CHostData:

Método	Parámetro o valor devuelto	Descripción
BSTR CurrentDirectory();	Valor devuelto = BSTR	Devuelve la ruta completa del directorio actual en el que se almacenan los objetos host de pcAnywhere.
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Parámetro = LPCTSTR lpszNewDirectory Valor devuelto = BOOL	Cambia el directorio actual en el que se almacenan los objetos host de pcAnywhere. El parámetro es el nombre de un directorio existente. El valor devuelto es TRUE si es correcto.
BOOL FindFirst(LPCTSTR lpszPattern);	Parámetro = LPCTSTR lpszPattern Valor devuelto = BOOL	Busca el primer archivo de objeto host de pcAnywhere (*.bhf) del directorio actual, basándose en el patrón de búsqueda de nombre del archivo indicado. El parámetro es el patrón de búsqueda de nombre de archivo para filtrar los archivos de objetos (un asterisco [*] busca todos los archivos host del directorio actual). El valor devuelto es TRUE si se encuentra un archivo de objeto host que coincide con el patrón especificado. El nombre completo de la ruta del archivo coincidente se almacena en pbstrFullQualName.

Método	Parámetro o valor devuelto	Descripción
BOOL FindFirst(BSTR FAR* pbstrFullQualName);	Parámetro = BSTR FAR * pbstrFullQualName	Devuelve el nombre y la ruta del archivo de objeto remoto que cumple el patrón especificado. El valor devuelto es TRUE si se encuentra un archivo de objeto host que cumpla con el patrón especificado. El nombre completo de la ruta del archivo coincidente se almacena en pbstrFullQualName.
BOOL FindNext(BSTR FAR* pbstrFullQualName);	Parámetro = BSTR FAR * pbstrFullQualName Valor devuelto = BOOL	Cuando FindFirst() se ha llamado correctamente para obtener el nombre de un archivo de objeto host del directorio actual, se puede llamar a FindNext() para encontrar el siguiente archivo que cumpla el patrón, si existe alguno. El parámetro es el búfer devuelto del nombre completo de la ruta del archivo de objeto host que cumple el patrón de búsqueda indicado en la llamada original a FindFirst(). El valor devuelto es TRUE si se encuentra otro archivo de objeto host que cumpla el patrón indicado en la llamada a FindFirst(). El nombre completo de la ruta del archivo coincidente se almacena en pbstrFullQualName.
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName	Obtiene un objeto CHostData según el nombre de archivo. El parámetro es el nombre y la ruta del archivo de objeto host que se debe cargar.

Método	Parámetro o valor devuelto	Descripción
LPDISPATCH RetrieveObject(short wAccessMode);	Parámetro = short wAccessMode	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword	Contraseña del objeto. Puede tener el valor NULL.
LPDISPATCH RetrieveObject	Valor devuelto = LPDISPATCH	Puntero a un objeto OLE dual. El objeto es un objeto CHostData. Consulte Código de muestra de Visual C++ para las funciones de host.
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName	Obtiene un objeto CHostDataEx según el nombre de archivo. El parámetro es el nombre y la ruta del archivo de objeto host que se debe cargar.
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parámetro = short wAccessMode	Indica el modo de utilización de este objeto, que se relaciona con la protección mediante contraseña. Las opciones son las siguientes: ■ 0 = No se ha especificado ■ 1 = Sólo ver ■ 2 = Ver y modificar ■ 3 = Ejecutar
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword	Contraseña del objeto. Puede tener el valor NULL.
LPDISPATCH RetrieveObjectEx	Valor devuelto = LPDISPATCH	Puntero a un objeto OLE dual. El objeto es un objeto CHostDataEx. Consulte Código de muestra de Visual C++ para las funciones de host.

Método	Parámetro o valor devuelto	Descripción
LPDISPATCH CreateObject(LPCTSTR lpszName);	Parámetro = LPCTSTR lpszFQName Valor devuelto = LPDISPATCH	Crea un objeto CHostData y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto host para el nuevo objeto. El valor devuelto es un puntero a un objeto OLE dual. El objeto es un objeto CHostData. Consulte Código de muestra de Visual C++ para las funciones de host.
LPDISPATCH CreateObjectEx(LPCTSTR lpszName);	Parámetro = LPCTSTR lpszFQName Valor devuelto = LPDISPATCH	Crea un objeto CHostDataEx y devuelve un puntero LPDISPATCH hacia él. El parámetro es el nombre y la ruta del archivo de objeto host para el nuevo objeto. El valor devuelto es un puntero a un objeto OLE dual. El objeto es un objeto CHostDataEx. Consulte Código de muestra de Visual C++ para las funciones de host.
BOOL DeleteObject(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName	Elimina un archivo de objeto host. El parámetro es el nombre y la ruta del archivo de objeto host que se eliminará.
BOOL DeleteObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword	Contraseña del objeto. Puede tener el valor NULL.
BOOL DeleteObject	Valor devuelto = BOOL	TRUE si el objeto se ha eliminado.
BOOL Launch(LPCTSTR lpszFQName);	Parámetro = LPCTSTR lpszFQName Valor devuelto = BOOL	Ejecuta un archivo de objeto host, que se encargará de abrir la ventana del terminal host de pcAnywhere. El parámetro es el nombre y la ruta del archivo de objeto host que se iniciará. El valor devuelto es TRUE si el objeto se inicia correctamente.

Objeto CHostData

Utilice este objeto para modificar los datos de un objeto host.

Métodos Get y Set

Los siguientes métodos se utilizan para obtener y configurar las propiedades de un objeto CHostData.

El número de teléfono es el número que se debe marcar para establecer una conexión mediante módem con un equipo remoto de pcAnywhere.

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

Indica si se deben utilizar las propiedades de marcado TAPI (información de la ubicación) (TRUE), o bien, si la cadena del número de teléfono debe utilizarse exactamente como aparece (FALSE).

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

Si se van a utilizar las propiedades de marcado, éste es el código de área del número al que se debe llamar.

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

Si se van a utilizar las propiedades de marcado, éste es el código de país del número al que se debe llamar.

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

La cantidad de veces que se debe intentar marcar este número si falla la llamada.

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

El tiempo de espera (en segundos) entre los intentos de llamada.

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

Determina si en las sesiones en las que se utiliza este objeto se va a efectuar el registro.

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

Determina si las sesiones que utilizan este objeto se grabarán desde el comienzo.

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

El nombre del archivo de grabación para las sesiones que utilicen este objeto.

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

Métodos de detalles de objetos host

Cuando a un objeto host se le asigna un tipo de conexión, los detalles del dispositivo se configuran en los valores válidos predeterminados.

Los siguientes tipos de conexión cuentan con opciones de configuración avanzadas que pueden definirse en la aplicación:

- Dispositivos COM
- Dispositivos de NetBIOS
- Dispositivos de NASI/NCSS
- Dispositivos ISDN mediante CAPI 2.0 (sólo para ISDN europea)

Detalles del dispositivo COM

El siguiente código sitúa el tipo de conexión solicitado en la lista de tipos de conexión asignados al objeto host y lo transforma en el tipo de conexión actual cuando se estén procesando las siguientes llamadas de métodos específicos del dispositivo:

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

Si el tipo de conexión solicitado ya se encuentra en la lista de conexiones asignadas, esta lista no cambia. Solo el tipo de conexión actual se cambia al tipo solicitado. Es normal llamar al método de AssignConnection varias veces sobre el mismo objeto mientras se están obteniendo y fijando los valores específicos de la conexión.

AssignConnection devuelve TRUE si el tipo de conexión ya existe en el equipo y se asigna correctamente o ya está asignado. Devuelve FALSE si el tipo de conexión solicitado no existe en el equipo o la cuenta de conexión asignada actualmente ya tiene el número máximo permitido de tipos de conexión.

Un objeto host de pcAnywhere puede admitir hasta dos tipos de conexión asignados. El método de AssignConnection devuelve FALSE si detecta cualquier intento de sobrepasar este límite.

Los siguientes tipos de conexión están disponibles:

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

También puede utilizarse como tipo de conexión el nombre de un dispositivo TAPI. DEFAULT TAPI utiliza el primer dispositivo TAPI que se encuentre en el sistema. Para utilizar un dispositivo TAPI específico, utilice FirstConnectionType() y NextConnectionType() para buscar los dispositivos disponibles.

El siguiente código deja de asignar un tipo de conexión. Después de quitar la asignación de un tipo de conexión, la conexión que queda (si hubiera una) pasa a ser el tipo de conexión actual para las siguientes llamadas de métodos específicos del dispositivo.

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

El siguiente código define los niveles de paridad de las comunicaciones:

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

Los valores de paridad de las comunicaciones son los siguientes:

- Ninguna
- Impar
- Par
- Marca
- Espacio

El siguiente código define los niveles del control de flujo:

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

Los valores de control de flujo son los siguientes:

- <Ninguno>
- XONXOFF
- RTS/CTS
- AMBOS

El siguiente código define los valores de inicio de conexión:

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

Los valores de inicio de conexión son los siguientes:

- Siempre conectado
- Detección de portadora (DCD)
- Preparado para enviar (CTS)
- Datos preparados (DSR)
- Indicador de timbre (RI)
- Recibir 2 <CR>
- Respuesta del módem

El siguiente código define los valores de fin de conexión:

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

Los valores de fin de conexión son los siguientes:

- Siempre conectado
- Detección de portadora (DCD)
- Preparado para enviar (CTS)
- Datos preparados (DSR)
- Indicador de timbre (RI)

El siguiente código define la velocidad de la conexión:

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

Los valores de la velocidad de conexión son los siguientes:

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

Detalles del dispositivo de NetBIOS

Define el número de adaptador LAN (LANA) que se debe utilizar para esta conexión.

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

Detalles del dispositivo NASI/NCSI

Define el nombre de usuario para el servidor NASI.

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

Define la contraseña de usuario para el servidor NASI.

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

Define el nombre de la sesión NASI.

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

Especifica el servidor NASI que se debe utilizar.

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

Especifica el servicio NASI que se debe utilizar.

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

Especifica el puerto NASI que se debe utilizar.

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

Detalles del dispositivo ISDN mediante CAPI 2.0

Activa el enlace de canales (utiliza dos canales ISDN para una conexión) si el valor es TRUE.

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

Define todas las extensiones adicionales CAPI necesarias para las comunicaciones.

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

Métodos de objetos host

En la siguiente tabla, se describen los métodos normales del objeto (no se utilizan para obtener y definir las propiedades):

Método	Parámetro o valor devuelto	Descripción
short ConnectionTypes();	Valor devuelto = Short	Devuelve la cantidad de tipos de conexión disponibles actualmente en este equipo.

Método	Parámetro o valor devuelto	Descripción
BSTR FirstConnectionType(); y BSTR NextConnectionType();	Valor devuelto = BSTR	FirstConnectionType() y NextConnectionType() se utilizan para explorar los tipos de conexión disponibles. Las funciones devuelven una cadena BSTR, que es el nombre de un tipo de conexión disponible. Estos tipos de conexión devueltos se pueden utilizar con la función SetConnectionType(). El valor devuelto es el nombre de un tipo de dispositivo con conexión admitida.
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Parámetro = LPCTSTR lpszConnectionTyp Valor devuelto = BOOL	El parámetro es el nombre de un tipo de dispositivo de conexión. El valor devuelto es TRUE si este tipo de dispositivo está disponible.
short MaxAssignedConnections()	Valor devuelto = Short	Devuelve el número máximo de tipos de conexión que se pueden asignar de forma simultánea (en este momento, dos).
short AssignedConnections()	Valor devuelto = Short	Devuelve el número de tipos de conexión asignados.
BSTR FirstAssignedConnection (); y BSTR NextAssignedConnection ();	Valor devuelto = BSTR	FirstAssignedConnection() y NextAssignedConnection() se utilizan para explorar la lista de conexiones asignadas. Las funciones devuelven una cadena BSTR, que es el nombre de un tipo de conexión asignado. Estos tipos de conexión devueltos se pueden utilizar con la función AssignConnection(). El valor devuelto es el nombre de un tipo de dispositivo con conexión admitida.
BOOL FindAssignedConnection (LPCTSTR lpszConnectionType);	Parámetro = LPCTSTR lpszConnectionTyp Valor devuelto = BOOL	El parámetro es el nombre de un tipo de dispositivo de conexión. El valor devuelto es TRUE si este tipo de dispositivo está disponible en ese momento.
short CountryCodes();	Valor devuelto = Short	Devuelve el número de códigos de países disponibles.

Método	Parámetro o valor devuelto	Descripción
BSTR FirstCountryCode(); y BSTR NextCountryCode();	Valor devuelto = BSTR	FirstCountryCode() y NextCountryCode() se utilizan para explorar la lista de códigos de países disponibles. Las funciones devuelven una cadena BSTR, que es el nombre de un código de país disponible. Estos valores devueltos se pueden utilizar con la función SetCountryCode(). El valor devuelto es la primera o la siguiente cadena del código de país.
BOOL ReadObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword Valor devuelto = BOOL	Lee los datos de objetos del archivo de objeto host. El parámetro es la contraseña del objeto. El valor devuelto es TRUE si el objeto se lee correctamente.
BOOL WriteObject(LPCTSTR lpszPassword);	Parámetro = LPCTSTR lpszPassword Valor devuelto = BOOL	Escribe los datos de objetos en el archivo de objeto host. El parámetro es la contraseña del objeto. El valor devuelto es TRUE si el objeto se escribe correctamente.

Objeto CHostDataEx

El objeto CHostDataEx contiene las mismas funciones que el objeto CHostData con los siguientes métodos Get y Set adicionales:

```
BOOL GetReadProtection();  
void SetReadProtection(BOOL bNewValue);  
BOOL GetWriteProtection();  
void SetWriteProtection(BOOL bNewValue);  
BSTR GetPassword(); //Devuelve "NO IMPLEMENTADO"  
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); // Configura 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);
```

Código de muestra de Visual C++ para las funciones de host

La siguiente función de muestra de Visual C++ crea un objeto host, establece su tipo de conexión en TCP/IP, establece el nombre del equipo en la dirección TCP/IP introducida en la función e inicia el objeto host.

```
BOOL LaunchTCPHost(LPCTSTR lpszAddress)  
{  
    BOOL bReturn = FALSE;  
  
    CHostDataManager hostDM;  
    CHostData hostData;  
  
    // Primero se crea CHostDataManager  
    hostDM.CreateDispatch( _T( "WINAWSVR.BeHostDataManager" ) );  
  
    // Después se crea CRemoteData y se lo adjunta  
    hostData.AttachDispatch(hostDM.CreateObject("Prueba", 0) );  
  
    // Por último, se configuran las propiedades necesarias  
    hostData.SetConnectionType("TCP/IP");  
  
    // Se guardan los datos del objeto
```

```
if (hostData.WriteObject(0))
{
// Y se inicia
if (hostData.Launch())
    bReturn = TRUE;
}

// Se libera el objeto host.
hostData.ReleaseDispatch();

return (bReturn);
}
```

funciones de Awrem32

La tabla siguiente describe las funciones de Awrem32, incluso los parámetros y los valores devueltos para gestionar conexiones entre un host y un equipo remoto:

Método	Parámetro o valor devuelto	Descripción
boolean awConnect(BSTR FileName);	Parámetro = Name as string	Crea la conexión con el equipo host.
	Valor devuelto = Boolean	El parámetro es el nombre y la ruta del archivo .chf que contiene información sobre el equipo host.
		El valor devuelto ejecuta el comando.
boolean awDisconnect();	Valor devuelto = Boolean	Desconecta el equipo host. Después de llamar a esta función, el programa que ejecuta la llamada debe eliminar el objeto (C++ - delete IAwrem32X*, VB - set ObjectName = Nothing;).

Método	Parámetro o valor devuelto	Descripción
boolean FileXferFromHost(BSTR HostFile);	Parámetro = HostFile as string	Copia un archivo desde el equipo host en el equipo remoto. Los parámetros pueden contener caracteres comodín. El parámetro contiene el nombre de archivo y la ruta completos que se van a copiar desde el equipo host. No es necesario que las cadenas HostFile y RemoteFile sean idénticas.
boolean FileXferFromHost(BSTR RemoteFile);	Parámetro = RemoteFile as string	Copia un archivo desde el equipo host en el equipo remoto. Los parámetros pueden contener caracteres comodín. El parámetro contiene la ruta de destino y el nombre de archivo completos. No es necesario que las cadenas HostFile y RemoteFile sean idénticas.
boolean FileXferFromHost	Valor devuelto = Boolean	TRUE si se ejecuta el comando.
boolean FileXferToHost(BSTR HostFile);	Parámetro = HostFile as string	Copia un archivo desde el equipo remoto en el equipo host. Los parámetros pueden contener caracteres comodín. El parámetro contiene la ruta de destino y el nombre de archivo completos. No es necesario que las cadenas HostFile y RemoteFile sean idénticas.
boolean FileXferToHost(BSTR RemoteFile);	Parámetro = RemoteFile as string	Copia un archivo desde el equipo remoto en el equipo host. Los parámetros pueden contener caracteres comodín. El parámetro contiene el nombre de archivo y la ruta completos que se van a copiar desde el equipo remoto. No es necesario que las cadenas HostFile y RemoteFile sean idénticas.
boolean FileXferToHost	Valor devuelto = Boolean	TRUE si se ejecuta el comando.

Método	Parámetro o valor devuelto	Descripción
boolean CreateFolderOnHost(BSTR FolderName);	Parámetro = FolderName as string Valor devuelto = Boolean	Crea una nueva carpeta en el equipo host. Esta función crea una carpeta temporal en el equipo remoto y, a continuación, la copia en el equipo host. El parámetro contiene la unidad y la ruta donde se debe crear la carpeta en el equipo host. El valor devuelto es TRUE si el comando se ejecutó.
boolean ExecuteHostFile(BSTR FileName);	Parámetro = FileName as string Valor devuelto = Boolean	Ejecuta un archivo existente en el equipo host. Esta función sólo permite ejecutar archivos por lotes, de comandos y ejecutables. No permite ejecutar archivos asociados con ejecutables. Por ejemplo, no abre Microsoft Word si se ejecuta un archivo .doc. El parámetro contiene la ruta completa al archivo en el equipo host. El valor devuelto es TRUE si el comando se ejecutó.
BSTR GetError();	Valor devuelto = String	Devuelve el último error producido en Awrem32 como cadena.
short ConnectionStatus();	Valor devuelto = Short	Devuelve el estado actual de la conexión al equipo host. Entre los valores posibles se incluyen los siguientes: ■ -1 = Conexión perdida ■ 0 = Sin conexión ■ 1 = Sesión conectada

Índice

A

- archivos
 - en ejecución en el host 59, 90
- Automatización OLE.. *Ver* Servidor de automatización de pcAnywhere
- Awrem32
 - funciones 58, 88

B

- bibliotecas API 14
- bibliotecas de tipos 14

C

- C++. *Ver* Visual C++
- carpetas
 - creación en el host 59, 90
- códigos de países
 - en host 84
 - en remoto 72
- conexiones
 - devolución de errores 90
 - devolución de estado 59, 90
 - finalizar 58, 88
 - inicio 37, 58, 66, 88
- conexiones CAPI
 - propiedades del host 48, 83
 - propiedades remotas 30, 71
- conexiones de red
 - en host
 - propiedades de NetBIOS 47, 82
 - en remoto
 - propiedades de NetBIOS 29, 71
- conexiones NetBIOS
 - propiedades del host 47, 82
 - propiedades remotas 29, 71
- conexiones por módem
 - en host
 - códigos de países 84
 - dispositivos TAPI 80
 - propiedades de COM 44, 79

- propiedades de marcado 43
- en remoto
 - códigos de países 72
 - dispositivos TAPI 68
 - propiedades de COM 26, 69
 - propiedades de marcado 25
- configuración del cifrado
 - en host 48, 85
 - en remoto 31–32, 72
- contraseñas
 - en objetos host 85
 - en objetos remotos 72
- control de flujo 27, 44, 69, 80
- controladores de automatización
 - acerca de Visual Basic 12
 - acerca de Visual C++ 13
- CRemoteData
 - utilización de Visual Basic 21
 - utilización de Visual C++ 66
- CRemoteDataEx
 - utilización de Visual Basic 31
 - utilización de Visual C++ 72
- CRemoteDataManager
 - utilización de Visual Basic 18
 - utilización de Visual C++ 62

CH

- CHostData
 - utilización de Visual Basic 37
 - utilización de Visual C++ 78
- CHostDataEx
 - utilización de Visual Basic 48
 - utilización de Visual C++ 85
- CHostDataManager
 - utilización de Visual Basic 34
 - utilización de Visual C++ 74

D

- definiciones de clases
 - importación 14
 - ver 15

- dispositivos. *Ver* tipos de conexión
- dispositivos COM
 - detalles del dispositivo host 44, 79
 - detalles del dispositivo remoto 26, 69
- dispositivos NASI 82
- dispositivos NCSI 82
- dispositivos TAPI
 - en host 80
 - en remoto 68
- dispositivos.. *Ver* tipos de conexión

E

- errores
 - devolver 59, 90
- estado
 - devolver 59, 90

G

- GUID 11

I

- identificadores. *Ver* GUID
- ISDN
 - conexiones CAPI
 - propiedades del host 48, 83
 - propiedades remotas 30, 71

M

- motor remoto
 - registro automático 11
 - registro manual 12
- muestras de código
 - Visual Basic 32, 56
 - Visual C++ 73, 87

O

- objetos host
 - búsqueda 35–36, 74, 76
 - contraseñas 85
 - creación 77
 - detalles del dispositivo 79
 - directorios 34, 74
 - eliminar 36, 77
 - inicio 37, 77
 - métodos 83
 - propiedades de marcado 43

- objetos remotos
 - archivos 19–20, 34, 63–64
 - contraseñas 72
 - crear 65
 - detalles del dispositivo 68
 - directorios 18, 62
 - eliminar 65
 - métodos 71
 - propiedades de marcado 25

P

- paridad 27, 44, 69, 80

S

- Servidor de automatización de pcAnywhere
 - acceso con Visual Basic 12
 - acceso con Visual C++ 14
 - acerca del 10
 - bibliotecas de tipos 14
 - registro de GUID 11
 - usos de ejemplo 10
- servidor de automatización.. *Ver* Servidor de automatización de pcAnywhere

T

- tipos de conexión
 - asignación 42
 - propiedades del host 39, 79, 83
 - propiedades remotas 23, 68, 71
 - sin asignación 42
- transferencia de archivos
 - al host 59, 89
 - desde el host 58, 89

V

- Visual Basic
 - acceso al servidor de automatización de pcAnywhere 12
 - funciones de Awrem32 58
 - muestras de código 32, 56
 - objeto CHostData 37
 - objeto CHostDataEx 48
 - objeto CHostDataManager 34
 - objeto CRemoteData 21
 - objeto CRemoteDataEx 31
 - objeto CRemoteDataManager 18

Visual C++

- acceso al servidor de automatización de
pcAnywhere 14
- funciones de Awrem32 88
- importación de clases 14
- incluir Winawsvr.h 15
- muestras de código 73, 87
- objeto CHostData 78
- objeto CHostDataEx 85
- objeto CHostDataManager 74
- objeto CRemoteData 66
- objeto CRemoteDataEx 72
- Objeto CRemoteDataManager 62

W**Winawsvr**

- archivo de encabezado 15
- ejecutable 12
- objetos
 - CHostData 37, 78
 - CHostDataEx 48, 85
 - CHostDataManager 34, 74
 - CRemoteData 21, 66
 - CRemoteDataEx 31, 72
 - CRemoteDataManager 18, 62