

Guida all'automazione OLE

Guida all'automazione OLE

Note legali

Copyright © 2008 Symantec Corporation. Tutti i diritti riservati.

Symantec e il logo Symantec sono marchi commerciali o marchi registrati di Symantec Corporation o delle relative società affiliate negli Stati Uniti e altri paesi. Altri nomi possono essere marchi dei rispettivi proprietari.

Questo prodotto Symantec può contenere software di terze parti per cui Symantec è obbligata a fornire l'attribuzione a terzi ("Programmi di terze parti"). Alcuni programmi di terze parti sono disponibili con licenza di software libero o gratuito. Il contratto di licenza che accompagna il software non modifica eventuali obbligazioni e diritti derivanti dall'utilizzo di tale software con versione di licenza libera o gratuita. Consultare l'appendice relativa alle note legali di terze parti inclusa nella presente documentazione o il file Leggimi TPIP fornito con il prodotto Symantec per ulteriori informazioni sui programmi di terze parti.

Il prodotto descritto nel presente documento è distribuito in base a licenze che ne limitano utilizzo, copia, distribuzione e decompilazione/ingegnerizzazione inversa. Nessuna parte del presente documento può essere riprodotta in qualsiasi forma e con qualsiasi mezzo senza previa autorizzazione scritta di Symantec Corporation e dei suoi eventuali licenziatari.

LA PRESENTE DOCUMENTAZIONE VIENE FORNITA "COSÌ COM'È" E TUTTE LE CONDIZIONI ESPLICITE O IMPLICITE, LE RAPPRESENTAZIONI E LE GARANZIE, COMPRESSE TUTTE LE GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ, IDONEITÀ PER UN PARTICOLARE SCOPO O NON VIOLAZIONE, SONO DECLINATE, AD ECCEZIONE DELLE SITUAZIONI IN CUI TALI DECLINAZIONI DI RESPONSABILITÀ SONO RITENUTE LEGALMENTE INVALIDE.

SYMANTEC CORPORATION NON POTRÀ ESSERE RITENUTA RESPONSABILE PER DANNI INCIDENTALI O CONSEGUENZIALI IN RELAZIONE ALLE PRESTAZIONI O L'UTILIZZO DELLA PRESENTE DOCUMENTAZIONE. LE INFORMAZIONI CONTENUTE NELLA PRESENTE DOCUMENTAZIONE SONO SOGGETTE A MODIFICA SENZA PREAVVISO.

Il software e la documentazione concessi in licenza sono da considerarsi software commerciale per computer secondo quanto definito dal FAR Sezione 12.212 e soggetti a diritti con restrizioni secondo quanto definito nel FAR Sezione 52.227-19 ("Software commerciale per computer - Diritti con restrizioni") e nel DFARS 227.7202 ("Diritti relativi al software commerciale per computer o alla documentazione del software commerciale per computer") laddove applicabile, e dalle disposizioni successive. Ogni uso, modifica, riproduzione, prestazione, esposizione o divulgazione del software o della documentazione concessi in licenza da parte del governo degli Stati Uniti dovrà avvenire in conformità con i termini stabiliti dal presente accordo.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

<http://www.symantec.it>

Realizzato in Irlanda

10 9 8 7 6 5 4 3 2 1

Supporto tecnico

Il supporto tecnico di Symantec gestisce centri di supporto a livello globale. Il ruolo primario del supporto tecnico è di rispondere alle domande specifiche sulle funzioni e le funzionalità del prodotto. Il supporto tecnico redige inoltre i contenuti della Knowledge Base in linea. Il supporto tecnico lavora in collaborazione con altri reparti di Symantec per rispondere tempestivamente alle domande degli utenti. Ad esempio, il team del supporto tecnico collabora con i reparti Product Engineering e Symantec Security Response per garantire i servizi di avviso e gli aggiornamenti delle definizioni dei virus.

Le offerte di manutenzione Symantec includono:

- Una gamma di opzioni di supporto da cui scegliere il livello di servizio adatto alla propria azienda
- Assistenza telefonica e tramite Web in grado di offrire una risposta rapida e informazioni aggiornate
- La sicurezza dell'aggiornamento automatico del software per la massima protezione
- Assistenza globale disponibile 24 ore al giorno e 7 giorni su 7
- Funzionalità avanzate, compresi i servizi di gestione degli account

Per informazioni sui programmi di assistenza tecnica di Symantec, è possibile visitare il nostro sito Web al seguente URL:

www.symantec.com/techsupp/

Come contattare il supporto tecnico

I clienti con un contratto di assistenza tecnica possono accedere alle informazioni del supporto tecnico al seguente URL:

www.symantec.com/techsupp/

Prima di contattare il supporto tecnico, verificare di disporre dei requisiti di sistema elencati nella documentazione del prodotto. Inoltre, è necessario trovarsi davanti al computer per il quale si è riscontrato il problema, nel caso sia necessario riprodurlo.

Quando si contatta il supporto tecnico, è necessario avere a disposizione le seguenti informazioni:

- Livello della release del prodotto
- Informazioni hardware
- Memoria disponibile, spazio su disco e informazioni NIC

- Sistema operativo
- Versione e livello di patch
- Topologia di rete
- Informazioni relative a router, gateway e indirizzo IP
- Descrizione del problema:
 - Messaggi di errore e file di registro
 - Tentativi di risoluzione dei problemi prima di contattare Symantec
 - Modifiche recenti alla configurazione del software e alla rete

Licenze d'uso e registrazione

Se il prodotto Symantec richiede una registrazione o una chiave di licenza, accedere alla pagina Web del supporto tecnico all'indirizzo:

www.symantec.com/techsupp/

Servizio clienti

Le informazioni sul Servizio clienti sono disponibili al seguente indirizzo:

www.symantec.com/techsupp/

Il Servizio clienti è disponibile per la risoluzione dei seguenti tipi di problemi:

- Domande sulla concessione di licenze o dei numeri di serie
- Aggiornamenti dei dati di registrazione del prodotto, come la modifica di indirizzo o nome
- Informazioni generali sul prodotto (funzionalità, lingua, rivenditori)
- Informazioni più recenti su aggiornamenti e nuove versioni
- Informazioni sugli aggiornamenti di prodotto e i contratti di manutenzione
- Informazioni sui programmi Symantec Buying Programs
- Consigli sulle opzioni di supporto tecnico offerte da Symantec
- Domande di prevendita non tecniche
- Problemi legati ai CD-ROM o ai manuali

Informazioni sul contratto di manutenzione

Se si desidera contattare Symantec relativamente a un contratto di assistenza tecnica esistente, contattare il team di amministrazione del contratto di assistenza tecnica per l'area geografica di appartenenza nel modo seguente:

Asia-Pacifico e Giappone	contractsadmin@symantec.com
Europa, Medio Oriente e Africa	semea@symantec.com
Nord America e America Latina	supportsolutions@symantec.com

Ulteriori servizi per le imprese

Symantec offre una gamma completa di servizi che consentono di ottimizzare l'investimento nei prodotti Symantec e di sviluppare conoscenze, esperienze e capacità di analisi globale, per una gestione proattiva dei rischi aziendali.

I servizi per le imprese offerti includono:

Soluzioni Symantec Early Warning	Queste soluzioni forniscono avvisi di carattere preventivo riguardo ad attacchi cibernetici, un'analisi completa delle minacce e contromisure per prevenirli prima che si verifichino.
Servizi di sicurezza gestiti	Questi servizi eliminano la difficoltà di gestire e di monitorare le periferiche e gli eventi di sicurezza, assicurando una risposta veloce alle minacce reali.
Servizi di consulenza	I servizi di consulenza di Symantec forniscono supporto tecnico in sede da parte di personale esperto Symantec e dei relativi partner. I servizi di consulenza di Symantec offrono varie opzioni preconfezionate e personalizzabili che comprendono la valutazione, la progettazione, l'implementazione, il monitoraggio e la gestione delle funzionalità. Ogni servizio è stato ideato appositamente per stabilire e mantenere l'integrità e la disponibilità delle risorse IT.
Servizi di formazione	I servizi di formazione forniscono una serie completa di risorse per la formazione tecnica, educazione alle best practice, certificazione di sicurezza e programmi di sensibilizzazione.

Per accedere a ulteriori informazioni sui servizi per le imprese, visitare il sito Web al seguente URL:

<http://www.symantec.com>

Selezionare il paese o la lingua dall'indice del sito.

Sommario

Supporto tecnico	4
Capitolo 1 Uso di Automazione OLE con Symantec pcAnywhere	9
Informazioni su Automazione OLE	9
Informazioni sul Server di automazione di pcAnywhere	10
Attività possibili con il Server di automazione di pcAnywhere	10
Prima di iniziare	11
Registrazione automatica del motore remoto	11
Registrazione manuale del motore remoto	12
Accesso al Server di automazione di pcAnywhere	12
Accesso al Server di automazione di pcAnywhere con Visual Basic	12
Accesso al Server di automazione di pcAnywhere con Visual C++	13
Avvio di oggetti OLE host e remoti	16
Dove ottenere ulteriori informazioni	16
Capitolo 2 Definizioni degli oggetti Visual Basic	17
Informazioni sugli oggetti Visual Basic	17
Metodi di CRemoteDataManager	18
Proprietà CRemoteData	21
Proprietà dei tipi di connessione	23
Proprietà di composizione	25
Proprietà per le periferiche COM	26
Proprietà per le periferiche NetBIOS	30
Proprietà per le periferiche ISDN mediante CAPI 2.0	30
oggetto CRemoteDataEx	31
Codice di esempio di Visual Basic per le funzionalità remote	32
Metodi di CHostDataManager	34
Proprietà CHostData	37
Proprietà dei tipi di connessione	39
Metodo AssignConnection(ConnectionType)	42
Metodo UnassignConnection(ConnectionType)	42
Proprietà di composizione	43

- Proprietà per le periferiche COM 44
- Proprietà per le periferiche NetBIOS 47
- Proprietà per le periferiche ISDN mediante CAPI 2.0 48
- oggetto CHostDataEx 48
 - Codice di esempio di Visual Basic per le funzionalità dell'host 56
- funzioni Awrem32 58

Capitolo 3	Definizioni degli oggetti Visual C++ 61
	Informazioni sugli oggetti Visual C++ 61
	Metodi di CRemoteDataManager 62
	oggetto CRemoteData 66 - Metodi Get e Set 66 - Metodi relativi ai dettagli degli oggetti remoti 69 - Metodi dell'oggetto remoto 71
	oggetto CRemoteDataEx 73 Codice di esempio di VisualC++ per le funzionalità del remoto 73
	Metodi di CHostDataManager 74
	oggetto CHostData 78 - Metodi Get e Set 78 - Metodi relativi ai dettagli degli oggetti host 79 - Metodi dell'oggetto host 84
	oggetto CHostDataEx 86 Codice di esempio di Visual C++ per le funzionalità dell'host 88
	funzioni Awrem32 88
Indice	93

Uso di Automazione OLE con Symantec pcAnywhere

Il capitolo contiene i seguenti argomenti:

- [Informazioni su Automazione OLE](#)
- [Informazioni sul Server di automazione di pcAnywhere](#)
- [Attività possibili con il Server di automazione di pcAnywhere](#)
- [Prima di iniziare](#)
- [Dove ottenere ulteriori informazioni](#)

Informazioni su Automazione OLE

Automazione OLE è una tecnologia che consente di creare un'applicazione o un altro strumento di programmazione esterno, ad esempio uno script o una macro, per controllare e automatizzare qualsiasi funzione esposta all'interno di un'applicazione.

Automazione OLE è costituita dai seguenti componenti:

Server di Automazione OLE: un'applicazione o un componente software che espone le proprie funzionalità in modo che siano accessibili o controllabili da altre applicazioni o strumenti di programmazione.

Il Server di automazione di pcAnywhere è un esempio di server di Automazione OLE.

Controller di Automazione OLE: un'applicazione o uno strumento di programmazione che accede e controlla i componenti esposti dal server di Automazione OLE.

È possibile utilizzare qualsiasi linguaggio di programmazione che supporti Automazione OLE. I due linguaggi di programmazione più comuni sono Microsoft Visual Basic e Microsoft Visual C++.

Durante l'accesso a un server di Automazione OLE, un'applicazione esterna si collega al server, quindi richiede l'accesso a una o più interfacce pubbliche. Un'interfaccia è un punto di ingresso che consente l'accesso a uno o più metodi o proprietà correlati. Una volta ottenuta un'interfaccia al server, l'applicazione richiama qualsiasi metodo di interfaccia interno come se facesse parte dell'applicazione esterna stessa.

Informazioni sul Server di automazione di pcAnywhere

Il Server di automazione di pcAnywhere consente alle applicazioni esterne di gestire i file remoti e host di pcAnywhere in modo da automatizzare le operazioni di controllo remoto e di trasferimento file. Il Server di automazione di pcAnywhere funziona come alternativa programmabile all'interfaccia utente di Symantec pcAnywhere e ne rispetta molti aspetti del funzionamento predefinito.

Se, ad esempio, si crea un oggetto host in pcAnywhere, per impostazione predefinita viene assegnata la prima periferica TAPI modem disponibile. Analogamente, se tramite il Server di automazione di pcAnywhere si crea un oggetto host, quindi si enumera l'elenco di connessioni assegnate, la prima periferica TAPI modem disponibile è già assegnata.

Attività possibili con il Server di automazione di pcAnywhere

Il Server di automazione di pcAnywhere consente di automatizzare una serie di attività di amministrazione e produttività.

È possibile utilizzare il Server di automazione di pcAnywhere per fare quanto segue:

- distribuzione e installazione automatiche degli aggiornamenti software su più computer della rete
- pianificazione dei trasferimenti automatici dei file tra computer a scopo di controllo o archiviazione

- aggiunta o rimozione automatica di un nome dall'elenco dei chiamanti consentiti in ogni host pcAnywhere della rete

Questo manuale contiene molti esempi, sia in Visual Basic che in Visual C++, che illustrano le modalità di connessione e di uso del Server di automazione di pcAnywhere.

Vedere ["Informazioni sugli oggetti Visual Basic"](#) a pagina 17.

Vedere ["Informazioni sugli oggetti Visual C++"](#) a pagina 61.

Prima di iniziare

Durante la connessione al Server di automazione di pcAnywhere e alle relative interfacce, i parametri identificatori, noti anche come GUID (Globally Unique Identifiers, Identificatori globalmente univoci), vengono trasmessi alle funzioni API della libreria di automazione. Un identificatore GUID separato viene assegnato al Server di automazione di pcAnywhere e a ogni interfaccia esposta. Questi identificatori devono essere presenti nel registro di sistema per collegare un'applicazione esterna al Server di automazione di pcAnywhere e alle relative interfacce.

Se si esegue l'applicazione esterna su un computer in cui è installato Symantec pcAnywhere, è possibile registrare automaticamente le voci GUID. In caso contrario, è necessario registrarle manualmente.

Vedere ["Registrazione automatica del motore remoto"](#) a pagina 11.

Vedere ["Registrazione manuale del motore remoto"](#) a pagina 12.

Registrazione automatica del motore remoto

Prima di effettuare la connessione a un altro computer per la prima volta mediante il client OLE, è necessario eseguire la registrazione automatica del motore remoto. È possibile eseguire questa operazione automaticamente avviando un oggetto remoto mediante Symantec pcAnywhere.

Per eseguire automaticamente la registrazione del motore remoto

- 1 Eseguire una delle operazioni seguenti per aprire Symantec pcAnywhere:
 - Sul desktop, fare doppio clic sull'icona del programma Symantec pcAnywhere.

- Nella barra delle applicazioni di Windows, fare clic su **Start > Programmi > Symantec pcAnywhere**.
 - 2 Nella finestra di pcAnywhere Manager, fare clic su **Remoti**.
 - 3 Fare doppio clic su un elemento di connessione remota.
- Questo processo registra il motore remoto. Non è necessario completare la connessione.

Registrazione manuale del motore remoto

Se pcAnywhere non è installato nel computer in cui è in esecuzione l'applicazione esterna, è necessario registrare manualmente gli identificatori GUID mediante il file eseguibile del Server di automazione di pcAnywhere (Winawsvr.exe). È sufficiente eseguire tale file una volta per aggiungere tutti gli identificatori GUID al registro di sistema. Il file Winawsvr.exe si trova nella directory di installazione.

Accesso al Server di automazione di pcAnywhere

Per accedere al Server di automazione di pcAnywhere, è possibile utilizzare qualsiasi linguaggio che supporti Automazione OLE. I due linguaggi più comuni che supportano Automazione OLE sono Visual Basic e Visual C++.

I principi di codifica di questi due linguaggi sono simili, sebbene in ambiente Visual Basic la maggior parte delle attività di basso livello venga eseguita in background dal sistema runtime di Visual Basic.

Accesso al Server di automazione di pcAnywhere con Visual Basic

Il linguaggio di programmazione Visual Basic include un supporto incorporato per l'interazione con i server di Automazione OLE, come il Server di automazione di pcAnywhere. Creare un progetto eseguibile standard, quindi immettere in ciascun metodo il codice necessario per accedere al Server di automazione di pcAnywhere. Visual Basic preleva le chiamate ai metodi di alto livello nei file di origine e li espande internamente nelle chiamate ai metodi di automazione OLE di basso livello corrispondenti.

Per accedere al Server di automazione di pcAnywhere con Visual Basic

- 1 Aggiungere due variabili di oggetto per ciascuno degli oggetti di pcAnywhere ai quali si desidera accedere.

Ad esempio, durante l'uso di oggetti remoti, impostare il DIM di **RemoteDataManager** e di **RemoteDataObject** come oggetto.

- 2 Utilizzare RemoteDataManager per collegarsi alla gestione dati dell'oggetto remoto.

Ad esempio, impostare la chiamata al metodo CreateObject con il parametro **WINAWSVR.REMOTEDATAMANAGER**.

Visual Basic utilizza il parametro testuale per individuare nel registro l'identificatore di gestione, quindi restituisce l'interfaccia alla funzione di gestione.

- 3 Una volta disponibile un oggetto di gestione dati valido, utilizzarlo per una qualsiasi delle operazioni seguenti:
 - determinazione della directory corrente
 - spostamento in un'altra directory
 - enumerazione dei file di oggetti di dati associati nella directory corrente
 - creazione, recupero o eliminazione di un file di oggetti dati.
- 4 Dopo aver creato o recuperato un oggetto di dati, è possibile richiamarne o impostarne le proprietà.

La sintassi Visual Basic non utilizza un nome di proprietà per distinguere il richiamo dall'impostazione del relativo valore. Per stabilire se la chiamata del metodo sottostante è Get o Set, viene invece utilizzata la posizione della proprietà in relazione all'operatore di assegnazione.

I seguenti esempi illustrano una chiamata Get e una Set.

- Per ottenere il valore del numero di telefono di un oggetto, posizionare il nome della proprietà a destra dell'operatore di assegnazione.
Ad esempio, `s = RemoteData.PhoneNumber()`, dove `s` è una variabile stringa.
- Per impostare il numero di telefono, posizionare il nome della proprietà a sinistra dell'operatore di assegnazione.
Ad esempio, `RemoteData.PhoneNumber = "555-1212"`

Accesso al Server di automazione di pcAnywhere con Visual C++

Il Server di automazione di pcAnywhere utilizza le librerie dei tipi per esporre le informazioni sulle interfacce e i metodi ai client di automazione scritti in Visual

C++. Queste librerie utilizzano le classi MFC (Microsoft Foundation Classes), che possono essere importate nell'applicazione in uso mediante Visual C++ ClassWizard.

Le classi di gestione dati disponibili nelle librerie dei tipi forniscono le funzionalità necessarie per ottenere un'interfaccia al Server di automazione di pcAnywhere ed eseguire operazioni di alto livello sul tipo di oggetto associato dell'interfaccia.

Utilizzare l'oggetto di gestione dati per eseguire una delle operazioni seguenti:

- determinazione o modifica della directory corrente
- enumerazione dell'elenco di file di oggetti di dati nella directory corrente
- creazione, recupero o eliminazione di un file oggetto denominato.

Una volta creato o recuperato, l'oggetto utilizza la classe di oggetto di dati associata per esaminare o modificare qualsiasi proprietà esposta. La maggior parte di queste proprietà sono esposte utilizzando due metodi che iniziano con i comandi Get o Set. Ad esempio, si utilizza il metodo GetPhoneNumber per esaminare la proprietà del numero di telefono dell'oggetto corrente e il metodo SetPhoneNumber per impostarlo.

Importazione e visualizzazione di classi

Il Server di automazione di pcAnywhere utilizza le seguenti librerie dei tipi:

Winawsvr.tlb:	fornisce le informazioni necessarie per la connessione al Server di automazione di pcAnywhere e l'accesso alle relative interfacce.
Awrem32.tlb:	fornisce le informazioni necessarie per il controllo delle connessioni di pcAnywhere.

Le procedure seguenti spiegano come importare nell'applicazione MFC le definizioni di classi dalle librerie dei tipi del Server di automazione di pcAnywhere e come visualizzare poi le classi aggiunte all'applicazione.

Per importare classi

- 1 In Visual C++, creare un'applicazione MFC.
- 2 Nel menu Visualizza, fare clic su **Class Wizard**.
- 3 Nella finestra di dialogo Class Wizard, fare clic su **Add Class**, quindi su **From a type library**.
- 4 Fare doppio clic su **winawsvr.tlb**.
- 5 Nella finestra di dialogo Confirm Classes, fare clic su **OK** per importare tutte le definizioni di classe.

- 6 Nella finestra di dialogo Class Wizard, fare clic su **Add Class**, quindi su **From a type library**.
- 7 Fare doppio clic su **awrem32.tlb**.
- 8 Nella finestra di dialogo Confirm Classes, fare clic su **OK** per importare tutte le definizioni di classe.
- 9 Nella finestra di dialogo Class Wizard, fare clic su **OK** per completare il processo di importazione.

Le classi vengono aggiunte all'applicazione. In questo modo sarà possibile gestire gli oggetti e le connessioni.

L'importazione delle definizioni di classi dalle librerie dei tipi consente di aggiungere anche i file di supporto all'applicazione. Questi file contengono le definizioni di classi e il codice sorgente di implementazione per il Server di automazione di pcAnywhere.

Vedere ["Visualizzazione delle definizioni di classi e dei file di implementazione"](#) a pagina 15.

Per visualizzare le classi aggiunte

- 1 In Visual C++, aprire l'applicazione MFC.
- 2 Nella finestra Workspace, fare clic sulla scheda **ClassView**.

Visualizzazione delle definizioni di classi e dei file di implementazione

Quando si importano le librerie dei tipi del Server di automazione di pcAnywhere nell'applicazione, vengono aggiunti i seguenti file:

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

Questi file contengono le definizioni di classi e il codice sorgente di implementazione per il Server di automazione di pcAnywhere. Non è necessario modificare questi file; tuttavia, ciascun file di origine dell'applicazione contenente le chiamate ai metodi del Server di automazione di pcAnywhere deve includere Winawsvr.h.

Per visualizzare le definizioni di classi e i file di implementazione

- 1 In Visual C++, aprire l'applicazione MFC.
- 2 Nella finestra dell'area di lavoro, fare clic sulla scheda **FileView**.

Avvio di oggetti OLE host e remoti

Symantec pcAnywhere richiede che tutti gli oggetti host e remoti vengano memorizzati nella directory di dati predefinita. Prima di avviare un oggetto host o remoto creato mediante la tecnologia OLE, verificare che si trovi nella directory di dati predefinita di pcAnywhere. La directory predefinita è la cartella All Users in Windows XP e la cartella ProgramData in Vista.

Dove ottenere ulteriori informazioni

Per ulteriori informazioni sull'automazione OLE, consultare i riferimenti seguenti:

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

Definizioni degli oggetti Visual Basic

Il capitolo contiene i seguenti argomenti:

- [Informazioni sugli oggetti Visual Basic](#)
- [Metodi di CRemoteDataManager](#)
- [Proprietà CRemoteData](#)
- [oggetto CRemoteDataEx](#)
- [Metodi di CHostDataManager](#)
- [Proprietà CHostData](#)
- [oggetto CHostDataEx](#)
- [funzioni Awrem32](#)

Informazioni sugli oggetti Visual Basic

Per supportare Automazione OLE, il Server di automazione di pcAnywhere fornisce i componenti seguenti:

Winawsvr	fornisce le informazioni necessarie per la connessione al Server di automazione di pcAnywhere e l'accesso alle relative interfacce.
Awrem32	fornisce le informazioni necessarie per il controllo delle connessioni di pcAnywhere.

Nella [Tabella 2-1](#) vengono descritti gli oggetti che costituiscono il componente Winawsvr.

Tabella 2-1 Oggetti Winawsvr

Oggetto	Descrizione	Riferimenti
CRemoteDataManager	Fornisce i metodi di creazione, apertura, modifica, salvataggio ed eliminazione di oggetti CRemoteData.	Vedere " Metodi di CRemoteDataManager " a pagina 18.
CRemoteData	Definisce i parametri per l'accesso e il controllo della funzionalità del remoto di pcAnywhere.	Vedere " Proprietà CRemoteData " a pagina 21. Vedere " oggetto CRemoteDataEx " a pagina 31.
CHostDataManager	Fornisce i metodi di creazione, apertura, modifica, salvataggio ed eliminazione di oggetti CHostData.	Vedere " Metodi di CHostDataManager " a pagina 34.
CHostData	Definisce i parametri per l'accesso e il controllo della funzionalità dell'host di pcAnywhere.	Vedere " Proprietà CHostData " a pagina 37. Vedere " oggetto CHostDataEx " a pagina 48.

Awrem32 comprende un solo oggetto, costituito da otto interfacce per supportare le sessioni di controllo remoto e trasferimento di file.

Vedere "[funzioni Awrem32](#)" a pagina 58.

Alcune funzioni non sono più supportate, ma le definizioni di oggetti vengono comunque fornite per consentire l'utilizzo con le versioni precedenti di pcAnywhere.

Per le funzioni che richiedono le password, i valori delle password possono essere impostati ma non richiamati. Ciò è per gli scopi di sicurezza.

Metodi di CRemoteDataManager

I metodi di CRemoteDataManager forniscono i parametri e i valori restituiti per l'accesso e il controllo degli oggetti CRemoteData.

Tabella 2-2 Metodi di CRemoteDataManager

Metodo CRemoteDataManager	Parametro o valore restituito	Descrizione
CurrentDirectory()	Valore restituito = Stringa	Restituisce il percorso completo della directory corrente in cui sono memorizzati gli oggetti remoti di pcAnywhere

Metodo CRemoteDataManager	Parametro o valore restituito	Descrizione
ChangeDirectory(NewDirectory)	Parametro = NewDirectory	Cambia la directory corrente in cui sono memorizzati gli oggetti remoti di pcAnywhere
FindFirst(Pattern)	Parametro = Criterio come stringa	Trova il primo file di oggetto remoto di pcAnywhere (*.chf) nella directory corrente; filtri per nome file in base al criterio specificato (un asterisco [*] trova tutti i file nella directory corrente)
FindFirst(Name string)	Parametro = Nome come stringa	Buffer di ritorno per il percorso completo del file di oggetto remoto (*.chf) corrispondente al criterio specificato
FindNext(Name)	Parametro = Nome come stringa	Buffer di ritorno per il percorso completo del file di oggetto remoto corrispondente al criterio specificato nella chiamata originale a FindFirst()
FindNext(Name)	Valore restituito = booleano	TRUE se viene trovato un altro file di oggetto remoto corrispondente al criterio specificato nella chiamata a FindFirst(). Il percorso completo del file corrispondente viene memorizzato in Name
RetrieveObject(Name)	Parametro = Nome come stringa	Il nome completamente qualificato del file di oggetto remoto da caricare
RetrieveObject(AccessMode)	Parametro = AccessMode come intero	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione
RetrieveObject>Password)	Parametro = Password come stringa	La password per l'oggetto. Può essere un valore NULL

Metodo CRemoteDataManager	Parametro o valore restituito	Descrizione
RetrieveObjectEx(Name)	Parametro = Nome come stringa	Il nome completamente qualificato del file di oggetto remoto da caricare
RetrieveObjectEx(AccessMode)	Parametro = AccessMode come intero	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione
RetrieveObjectEx(Password)	Parametro = Password come stringa	La password per l'oggetto. Può essere un valore NULL
RetrieveObjectEx	Valore restituito = Oggetto	L'oggetto CRemoteDataEx ottenuto dal file specificato
CreateObject(Name)	Parametro = Nome come stringa	Crea un oggetto CRemoteData e restituisce un puntatore LPDISPATCH che punta ad esso. Il parametro è il nome completamente qualificato del file dell'oggetto remoto per il nuovo oggetto
CreateObject	Valore restituito = Oggetto	CRemoteData
CreateObjectEx(Name)	Parametro = Nome come stringa	Crea un oggetto CRemoteDataEx e restituisce un puntatore LPDISPATCH che punta ad esso. Il parametro è il nome completamente qualificato del file dell'oggetto remoto per il nuovo oggetto
CreateObjectEx	Valore restituito = Oggetto	CRemoteDataEx
DeleteObject(Name)	Parametro = Nome come stringa	Il nome completamente qualificato del file di oggetto remoto dell'oggetto da eliminare

Metodo CRemoteDataManager	Parametro o valore restituito	Descrizione
DeleteObject>Password)	Parametro = Password come stringa	La password per l'oggetto
DeleteObject	Valore restituito = booleano	TRUE se l'oggetto viene eliminato

Proprietà CRemoteData

Le proprietà e i parametri disponibili per l'oggetto CRemoteData vengono visualizzati nel modo seguente.

In [Tabella 2-3](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-3 Proprietà e parametri CRemoteData

Proprietà	Parametro	Descrizione
<CRemoteData>.ComputerName(String)	Stringa	Imposta il nome o l'indirizzo IP del computer host.
<CRemoteData>.PhoneNumber(String)	Stringa	Imposta il numero di telefono del computer host.
<CRemoteData>.UseDialingProperties(Bool)	Booleano	Imposta le proprietà di composizione del sistema.
<CRemoteData>.RedialCount(Integer)	Intero	Imposta il numero di tentativi di composizione prima dell'annullamento della chiamata.
<CRemoteData>.RedialDelay(Integer)	Intero	Imposta il numero di secondi di attesa tra i tentativi di composizione.
<CRemoteData>.AutoLoginName(String)	Stringa	Imposta il nome dell'utente per l'accesso automatico. Per ulteriori informazioni sull'utilizzo degli account di dominio: Vedere " oggetto CRemoteDataEx " a pagina 31.

Proprietà	Parametro	Descrizione
<CRemoteData>.AutoLoginPassword(String)	Stringa	Imposta la password per gli accessi automatici nell'oggetto remoto. Per motivi di sicurezza, il Server di automazione di pcAnywhere non consente di leggere il valore della password. Il valore della password non viene restituito.
<CRemoteData>.Password(String)	Stringa	Imposta la password nell'oggetto remoto da utilizzare con le impostazioni per ExecuteProtection, ReadProtection e WriteProtection. Per motivi di sicurezza, il Server di automazione di pcAnywhere non consente di leggere il valore della password. Il valore della password non viene restituito.
<CRemoteData>.ExecuteProtection(Bool)	Booleano	Imposta il requisito di una password per l'esecuzione dell'oggetto. Impostato mediante password.
<CRemoteData>.ReadProtection(Bool)	Booleano	Imposta il requisito di una password per la visualizzazione delle proprietà dell'oggetto remoto. Impostato mediante password.
<CRemoteData>.WriteProtection(Bool)	Booleano	Imposta il requisito di una password per il salvataggio delle modifiche all'oggetto remoto. Impostato mediante password.
<CRemoteData>.LogSession(Bool)	Booleano	Attiva e disattiva la registrazione della sessione.
<CRemoteData>.RecordFile(String)	Stringa	Imposta il nome e il percorso completamente qualificati della posizione del file che registra la sessione attiva.
<CRemoteData>.RecordSession(Bool)	Booleano	Attiva e disattiva la registrazione automatica della sessione.
<CRemoteData>.ReadObject(String)	Stringa	Imposta la password per l'oggetto. Utilizzare tale proprietà per aggiornare la copia locale dei dati dell'oggetto remoto.
<CRemoteData>.WriteObject(String)	Stringa	Imposta la password per l'oggetto. Utilizzare tale proprietà per creare l'oggetto remoto o per scrivere le modifiche ad esso apportate.

Le proprietà e i valori restituiti per CRemoteData vengono mostrati nella seguente tabella.

In [Tabella 2-4](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-4 Proprietà e valori restituiti per CRemoteData

Proprietà	Valore restituito	Descrizione
Stringa = <CRemoteData>.ComputerName	Stringa	Restituisce il nome o l'indirizzo IP del computer host.
Stringa = <CRemoteData>.PhoneNumber	Stringa	Restituisce il numero di telefono del computer host
Booleano = <CRemoteData>.UseDialingProperties	Booleano	Restituisce le proprietà di composizione del sistema impostate nell'oggetto remoto
Intero = <CRemoteData>.RedialCount	Intero	Restituisce il numero di tentativi di composizione impostato nell'oggetto remoto
Intero = <CRemoteData>.RedialDelay	Intero	Restituisce il numero di secondi tra i tentativi di composizione
Stringa = <CRemoteData>.AutoLoginName	Stringa	Restituisce il nome di accesso utilizzato per gli accessi automatici
Booleano = <CRemoteData>.ExecuteProtection	Booleano	Restituisce il valore dell'impostazione per ExecuteProtection
Booleano = <CRemoteData>.ReadProtection	Booleano	Restituisce il valore dell'impostazione per ReadProtection
Booleano = <CRemoteData>.WriteProtection	Booleano	Restituisce il valore dell'impostazione per WriteProtection
Booleano = <CRemoteData>.LogSession	Booleano	Restituisce TRUE se è attivata la registrazione della sessione
Stringa = <CRemoteData>.RecordFile	Stringa	Restituisce il nome e il percorso completamente qualificati del file di registrazione della sessione
Booleano = <CRemoteData>.RecordSession	Booleano	Restituisce il valore dell'impostazione di registrazione della sessione

Proprietà dei tipi di connessione

La seguente tabella definisce le proprietà e i parametri dei tipi di connessione.

In [Tabella 2-5](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-5 Proprietà e parametri dei tipi di connessione

Proprietà	Parametro	Descrizione
<CRemoteData>.ConnectionType(String)	Stringa	Imposta il tipo di connessione del remoto. Il valore trasmesso deve essere un tipo di connessione valido definito dalle funzioni FirstConnectionType() e NextConnectionType(). Di seguito sono riportati esempi di tipi di connessione validi: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS. ■ ISDN tramite CAPI 2.0 ■ Nome modem (come visualizzato sul computer).

La seguente tabella definisce le proprietà e i valori restituiti dei tipi di connessione.

In [Tabella 2-6](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-6 Proprietà e valori restituiti dei tipi di connessione

Proprietà	Valore restituito	Descrizione
Stringa = <CRemoteData>.ConnectionTypes	Stringa	Restituisce il tipo di connessione dell'oggetto remoto.

Proprietà	Valore restituito	Descrizione
Intero = <CRemoteData>.ConnectionType	Intero	Restituisce il numero di tipi di connessione disponibili. Di seguito sono riportati esempi di tipi di connessione validi: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS. ■ ISDN tramite CAPI 2.0 ■ Nome modem (come visualizzato sul computer).
Stringa = <CRemoteData>.FirstConnectionType	Stringa	Restituisce il primo tipo di connessione disponibile.
Stringa = <CRemoteData>.NextConnectionType	Stringa	Restituisce il successivo tipo di connessione disponibile. La chiamata viene eseguita in sequenza per il numero di tipi di connessione impostato in <CRemoteData>.ConnectionType per enumerare tutti i tipi di connessione.
Booleano = <CRemoteData>.FindConnectionType(ConnectionType)	Booleano	Restituisce TRUE se il tipo di connessione specificato viene trovato nell'elenco dei tipi di connessione disponibili.

Proprietà di composizione

La seguente tabella definisce le proprietà e i parametri per impostare le proprietà di composizione per le connessioni tramite modem.

In [Tabella 2-7](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-7 Proprietà e parametri per le proprietà di composizione

Proprietà	Parametro	Descrizione
<CRemoteData>.AreaCode(String)	Stringa	Imposta le proprietà di composizione dell'indicativo di località per le connessioni modem.

Proprietà	Parametro	Descrizione
<CRemoteData>.CountryCode(String)	Stringa	Imposta le proprietà di composizione dell'indicativo di paese per le connessioni modem.

La seguente tabella definisce le proprietà e i valori restituiti per le proprietà di composizione del modem.

In [Tabella 2-8](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-8 Proprietà e valori restituiti per le proprietà di composizione

Proprietà	Valore restituito	Descrizione
Stringa = <CRemoteData>.AreaCode	Stringa	Restituisce le proprietà di composizione dell'indicativo di località.
Stringa = <CRemoteData>.CountryCode	Stringa	Restituisce l'indicativo di paese delle proprietà di composizione.
Intero = <CRemoteData>.CountryCodes	Intero	Restituisce il numero di indicativi di paese disponibili.
Stringa = <CRemoteData>.FirstCountryCode	Stringa	Restituisce il primo indicativo di paese disponibile elencato nel sistema operativo.
Stringa = <CRemoteData>.NextCountryCode	Stringa	Restituisce il successivo indicativo di paese disponibile. La chiamata viene eseguita in sequenza per il numero di indicativi di paese impostato in <CRemoteData.CountryCodes> per enumerare tutti i codici di paese.

Proprietà per le periferiche COM

Nella seguente tabella vengono descritti le proprietà e i parametri che consentono di personalizzare le impostazioni della porta per le connessioni modem e le altre connessioni basate su COM.

In [Tabella 2-9](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-9 Proprietà e parametri di periferiche COM

Proprietà	Parametro	Descrizione
<CRemoteData>.ComParity(String)	Stringa	Imposta la parità di comunicazione. I valori validi sono: ■ <Nessuna> ■ Dispari ■ Pari ■ Contrassegno ■ Spazio.
<CRemoteData>.ComFlowControl(String)	Stringa	Imposta il controllo di flusso dei tipi di connessione basata su COM. I valori validi sono: ■ <Nessuna> ■ XONXOFF ■ RTS/CTS ■ Entrambi
<CRemoteData>.ComStartedBy(String)	Stringa	Imposta il valore di avvio dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI). ■ Ricezione di 2 <CR> ■ Risposta modem.
<CRemoteData>.ComEndedBy(String)	Stringa	Imposta il valore di fine dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI).

Proprietà	Parametro	Descrizione
<CRemoteData>.ComSpeed(Long)	Lungo	Contiene l'impostazione di velocità massima della porta COM. I valori validi sono: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

La seguente tabella descrive i proprietà delle periferiche COM e i valori restituiti.

In [Tabella 2-10](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-10 Proprietà e valori restituiti delle periferiche COM

Proprietà	Valore restituito	Descrizione
Stringa = <CRemoteData>.ComParity	Stringa	Restituisce uno dei valori seguenti per la parità di comunicazione: ■ <Nessuna> ■ Dispari ■ Pari ■ Contrassegno ■ Spazio.
Stringa = <CRemoteData>.ComFlowControl	Stringa	Restituisce l'impostazione del flusso di comunicazione dell'oggetto remoto. I valori validi sono: ■ <Nessuna> ■ XONXOFF ■ RTS/CTS ■ Entrambi

Proprietà	Valore restituito	Descrizione
Stringa = <CRemoteData>.ComStartedBy	Stringa	Restituisce il controllo dell'avvio di comunicazione dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI). ■ Ricezione di 2 <CR> ■ Risposta modem.
Stringa = <CRemoteData>.ComEndedBy	Stringa	Restituisce il controllo della fine comunicazione dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI).
Long = <CRemoteData>.ComSpeed	Lungo	Restituisce l'impostazione corrente della velocità di comunicazione dell'oggetto remoto. I valori validi sono: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Proprietà per le periferiche NetBIOS

La seguente tabella definisce la proprietà e il parametro per una periferica di rete di NetBIOS.

In [Tabella 2-11](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-11 Proprietà e parametro per NetBIOS

Proprietà	Parametro	Descrizione
<CRemoteData>.LanaNumber(Integer)	Intero	Imposta il numero LANA (LAN Adapter) per le connessioni NetBIOS.

La seguente tabella definisce il valore restituito e la proprietà NetBIOS.

In [Tabella 2-12](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-12 Proprietà e valore restituito per NetBIOS

Proprietà	Valore restituito	Descrizione
Intero = <CRemoteData>.LanaNumber	Intero	Restituisce l'impostazione corrente del numero LANA (LAN Adapter) per le connessioni NetBIOS.

Proprietà per le periferiche ISDN mediante CAPI 2.0

La seguente tabella definisce le proprietà e i parametri per le connessioni europee ISDN.

In [Tabella 2-13](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-13 Proprietà e parametri per ISDN mediante CAPI 2.0

Proprietà	Parametro	Descrizione
<CRemoteData>.CapiChannelBonding(Bool)	Booleano	Attiva o disattiva il collegamento a canale per le periferiche ISDN CAPI.
<CRemoteData>.CapiExtensions(String)	Stringa	Imposta eventuali estensioni CAPI aggiuntive necessarie per le comunicazioni.

La seguente tabella definisce le proprietà e i valori restituiti per le connessioni europee ISDN.

In [Tabella 2-14](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-14 Proprietà e valori restituiti per ISDN mediante CAPI 2.0

Proprietà	Valore restituito	Descrizione
Booleano= <CRemoteData>.CapiChannelBonding	Booleano	Restituisce l'impostazione corrente del collegamento a canale per le periferiche ISDN CAPI nell'oggetto remoto.
Stringa = <CRemoteData>.CapiExtensions	Stringa	Restituisce l'elenco corrente delle estensioni CAPI dell'oggetto remoto.

oggetto CRemoteDataEx

L'oggetto CRemoteDataEx contiene le stesse funzionalità dell'oggetto CRemoteData, più alcune funzionalità aggiuntive.

La seguente tabella descrive le proprietà e i parametri.

In [Tabella 2-15](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-15 Parametri CRemoteDataEx

Proprietà	Parametro	Descrizione
<CRemoteData>.PrivateKey(String)	Stringa	Imposta il nome del contenitore della chiave privata da utilizzare.
<CRemoteData>.CertificateName(String)	Stringa	Imposta il nome comune della chiave privata da utilizzare.
<CRemoteData>.EncryptionLevel(Byte)	Byte	Imposta il livello di crittografia. I valori validi sono: ■ -1: Nessuna ■ 0: pcAnywhere ■ 1: Simmetrica ■ 2: Chiave pubblica

Proprietà	Parametro	Descrizione
<CRemoteData>.DenyLowerEncrypt(Bool)	Booleano	Definisce se il computer remoto consente una connessione a un computer host che utilizza un livello di crittografia inferiore.
<CRemoteData>.AutoDomain(String)	Stringa	Imposta il nome di dominio per gli accessi automatici. Questa opzione viene utilizzata con le autenticazioni NT e Windows.

La seguente tabella descrive le proprietà e i valori restituiti per l'oggetto CRemoteDataEx.

In [Tabella 2-16](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-16 Proprietà e valori restituiti per CRemoteDataEx

Proprietà	Valore restituito	Descrizione
Stringa = <CRemoteData>.PrivateKey	Stringa	Restituisce il nome del contenitore della chiave privata attualmente attivo.
Stringa = <CRemoteData>.CertificateName	Stringa	Restituisce il nome comune del contenitore della chiave privata attivo.
Byte = <CRemoteData>.EncryptionLevel	Byte	Restituisce una delle seguenti impostazioni di crittografia: ■ -1: Nessuna ■ 0: pcAnywhere ■ 1: Simmetrica ■ 2: Chiave pubblica
Booleano = <CRemoteData>.DenyLowerEncrypt	Booleano	Restituisce il valore dell'impostazione per negare il livello di crittografia inferiore.
Stringa = <CRemoteData>.AutoDomain	Stringa	Restituisce l'impostazione del nome di dominio per gli accessi automatici. Questa opzione viene utilizzata con le autenticazioni NT e Windows.

Codice di esempio di Visual Basic per le funzionalità remote

Il codice di esempio di Visual Basic riportato di seguito consente di recuperare un oggetto di dati remoto e di modificarne le proprietà.

```
Private Sub Command1_Click()  
Dim RemoteDataManager as Object  
Dim RemoteData as Object  
Dim s as string  
  
'Creazione dell'oggetto CRemoteDataManager  
Set RemoteDataManager = CreateObject(WINAWSVR.REMOTEDATAMANAGER)  
  
'Visualizzazione e cambiamento della directory corrente  
s = RemoteDataManager.CurrentDirectory()  
MsgBox ( s )  
RemoteDataManager.ChangeDirectory ("C:\dev\bin.w32\data")  
s = RemoteDataManager.CurrentDirectory()  
MsgBox ( s )  
  
'Recupero dell'oggetto di dati remoto  
Set RemoteData = RemoteDataManager.RetrieveObjectEx("pod.CHF",  
2, 0)  
'Visualizzazione di alcune proprietà  
s = RemoteData.AreaCode()  
MsgBox (s)  
s = RemoteData.PhoneNumber()  
MsgBox (s)  
  
'Impostazione di alcune proprietà  
RemoteData.AreaCode = "212"  
RemoteData.PhoneNumber = "555-5555"  
  
'Scrittura dell'oggetto sul disco  
RemoteData.WriteObject (0)  
End Sub
```

Nell'esempio riportato di seguito vengono utilizzati i metodi FindFirst e FindNext per visualizzare un file remoto presente in una directory.

```
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
```

Crea un oggetto remoto. Impostare il tipo di connessione su TCP/IP e il nome del computer "Host1" e quindi avviarlo nel modo seguente:

```
Private Sub Command6_Click()
Dim RemoteDataManager as Object
Dim RemoteData as Object
Dim s as string

Set RemoteDataManager =
CreateObject ("WINAHSV.R. 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
```

Metodi di CHostDataManager

I metodi di CHostDataManager forniscono i parametri e i valori restituiti per l'accesso e il controllo degli oggetti CHostData.

Metodo CHostDataManager	Parametro o valore restituito	Descrizione
CurrentDirectory()	Valore restituito = Stringa	Il percorso completo della directory di dati pcAnywhere corrente in cui sono memorizzati gli oggetti host di pcAnywhere

Metodo CHostDataManager	Parametro o valore restituito	Descrizione
FindFirst(Pattern)	Parametro = Criterio come stringa	Criterio per il nome file utilizzato per filtrare i file di oggetti. Un asterisco [*] consente di trovare tutti i file presenti nella directory corrente.
FindFirst(Name string)	Parametro = Nome come stringa	Buffer di ritorno per il percorso completo del file di oggetto host corrispondente al criterio specificato
FindFirst	Valore restituito = booleano	TRUE se viene trovato un file di oggetto host corrispondente al criterio specificato. Il percorso completo del file corrispondente viene memorizzato in Name
FindNext(Name)	Parametro = Nome come stringa	Buffer di ritorno per il percorso completo del file di oggetto host corrispondente al criterio specificato nella chiamata originale a FindFirst()
FindNext	Valore restituito = booleano	TRUE se viene trovato un altro file di oggetto host corrispondente al criterio specificato nella chiamata a FindFirst(). Il percorso completo del file corrispondente viene memorizzato in Name
RetrieveObject(Name)	Parametro = Nome come stringa	Il nome completamente qualificato del file dell'oggetto host da caricare
RetrieveObject(AccessMode)	Parametro = AccessMode come intero	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password. Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione
RetrieveObject>Password)	Parametro = Password come stringa	La password per l'oggetto. Può essere un valore NULL
RetrieveObject	Valore restituito = Oggetto	L'oggetto CHostData ottenuto dal file specificato

Metodo CHostDataManager	Parametro o valore restituito	Descrizione
RetrieveObjectEx(Name)	Parametro = Nome come stringa	Il nome completamente qualificato del file dell'oggetto host da caricare
RetrieveObjectEx(AccessMode)	Parametro = AccessMode come intero	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione
RetrieveObjectEx>Password)	Parametro = Password come stringa	La password per l'oggetto. Può essere un valore NULL
RetrieveObjectEx	Valore restituito = Oggetto	L'oggetto CHostDataEx ottenuto dal file specificato
CreateObject(Name)	Parametro = Nome come stringa	Crea un oggetto CHostData e restituisce un puntatore LPDISPATCH che punta ad esso. Il parametro è il nome completamente qualificato del file di oggetti host per il nuovo oggetto
CreateObject	Valore restituito = Oggetto	CHostData
CreateObjectEx(Name)	Parametro = Nome come stringa	Crea un oggetto CHostDataEx e restituisce un puntatore LPDISPATCH che punta ad esso. Il parametro è il nome completamente qualificato del file di oggetti host per il nuovo oggetto
CreateObjectEx	Valore restituito = Oggetto	CHostDataEx
DeleteObject(Name)	Parametro = Nome come stringa	Il nome completamente qualificato del file di oggetto host dell'oggetto da eliminare

Metodo CHostDataManager	Parametro o valore restituito	Descrizione
DeleteObject(Password)	Parametro = Password come stringa	La password per l'oggetto
DeleteObject	Valore restituito = booleano	TRUE se l'oggetto viene eliminato
Launch(Name)	Parametro = Nome come stringa	Avvia un file di oggetto host, che apre la finestra del terminale host di pcAnywhere. Il parametro è il nome completamente qualificato del file di oggetti host da avviare
Avvio	Valore restituito = booleano	TRUE se l'oggetto viene avviato correttamente

Proprietà CHostData

La seguente tabella descrive le proprietà e i parametri disponibili per l'oggetto CHostData.

In [Tabella 2-17](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-17 Proprietà e parametri CHostData

Proprietà	Parametro	Descrizione
<CHostData>.PhoneNumber(String)	Stringa	Imposta il numero di telefono del computer host.
<CHostData>.UseDialingProperties(Bool)	Booleano	Imposta le proprietà di composizione del sistema.
<CHostData>.RedialCount(Integer)	Intero	Imposta il numero di tentativi di composizione prima dell'annullamento della chiamata.
<CHostData>.RedialDelay(Integer)	Intero	Imposta il numero di secondi di attesa tra i tentativi di composizione.

Proprietà	Parametro	Descrizione
<CHostData>.Password(String)	Stringa	Imposta la password nell'oggetto host da utilizzare con le impostazioni per ExecuteProtection, ReadProtection e WriteProtection. Per motivi di sicurezza, il Server di automazione di pcAnywhere non consente di leggere il valore della password. Il valore della password non viene restituito.
<CHostData>.ExecuteProtection(Bool)	Booleano	Imposta il requisito di una password per l'esecuzione dell'oggetto. Impostato mediante password.
<CHostData>.ReadProtection(Bool)	Booleano	Imposta il requisito di una password per la visualizzazione delle proprietà dell'oggetto host. Impostato mediante password.
<CHostData>.WriteProtection(Bool)	Booleano	Imposta il requisito di una password per il salvataggio delle modifiche apportate all'oggetto host. Impostato mediante password.
<CHostData>.LogSession(Bool)	Booleano	Attiva e disattiva la registrazione della sessione.
<CHostData>.RecordFile(String)	Stringa	Imposta il nome e il percorso completamente qualificati della posizione del file che registra la sessione attiva.
<CHostData>.RecordSession(Bool)	Booleano	Attiva e disattiva la registrazione automatica della sessione.
<CHostData>.ReadObject(String)	Stringa	Imposta la password per l'oggetto. Questa proprietà può essere utilizzata per aggiornare la copia locale dei dati dell'oggetto host.
<CHostData>.WriteObject(String)	Stringa	Imposta la password per l'oggetto. Utilizzare questa proprietà per creare l'oggetto host o per scrivere le modifiche apportate ad esso.

La seguente tabella descrive le proprietà e i valori restituiti per CHostData.

In [Tabella 2-18](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-18 Proprietà e valori restituiti per CHostData

Proprietà	Valore restituito	Descrizione
Stringa = <CHostData>.PhoneNumber	Stringa	Restituisce il numero di telefono del computer host
Booleano = <CHostData>.UseDialingProperties	Booleano	Restituisce l'uso delle proprietà di composizione del sistema impostate nell'oggetto host
Intero = <CHostData>.RedialCount	Intero	Restituisce il numero di tentativi di composizione impostato nell'oggetto host
Intero = <CHostData>.RedialDelay	Intero	Restituisce il numero di secondi tra i tentativi di composizione
Booleano = <CHostData>.ExecuteProtection	Booleano	Restituisce il valore dell'impostazione per ExecuteProtection
Booleano = <CHostData>.ReadProtection	Booleano	Restituisce il valore dell'impostazione per ReadProtection
Booleano = <CHostData>.WriteProtection	Booleano	Restituisce il valore dell'impostazione per Write Protection
Booleano = <CHostData>.LogSession	Booleano	Restituisce TRUE se è attivata la registrazione della sessione
Stringa = <CHostData>.RecordFile	Stringa	Restituisce il nome e il percorso completamente qualificati del file di registrazione della sessione
Booleano = <CHostData>.RecordSession	Booleano	Restituisce il valore dell'impostazione di registrazione della sessione

Proprietà dei tipi di connessione

La seguente tabella descrive le proprietà e i parametri dei tipi di connessione.

In [Tabella 2-19](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-19 Proprietà e parametri dei tipi di connessione

Proprietà	Parametro	Descrizione
<CHostData>.ConnectionType(String)	Stringa	Imposta il tipo di connessione dell'host. Il valore trasmesso deve essere un tipo di connessione valido definito dalle funzioni FirstConnectionType() e NextConnectionType(). Di seguito sono riportati esempi di tipi di connessione validi: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN tramite CAPI 2.0 ■ Nome modem (come visualizzato sul computer).
<CHostData>.AssignConnection(String)	Stringa	Imposta il tipo di connessione definito nella stringa sullo stato attivo.
<CHostData>.UnassignConnection(String)	Stringa	Imposta il tipo di connessione definito nella stringa sullo stato inattivo.

La seguente tabella descrive le proprietà e i valori restituiti dei tipi di connessione.

In [Tabella 2-20](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-20 Proprietà e valori restituiti dei tipi di connessione

Proprietà	Valore restituito	Descrizione
Stringa = <CHostData>.ConnectionTypes	Stringa	Restituisce il tipo di connessione dell'oggetto host.

Proprietà	Valore restituito	Descrizione
Intero = <CHostData>.ConnectionType	Intero	Restituisce il numero di tipi di connessione disponibili. Di seguito sono riportati esempi di tipi di connessione validi: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN tramite CAPI 2.0 ■ Nome modem (come visualizzato sul computer).
Stringa = <CHostData>.FirstConnectionType	Stringa	Restituisce il primo tipo di connessione disponibile.
Stringa = <CHostData>.NextConnectionType	Stringa	Restituisce il successivo tipo di connessione disponibile. La chiamata viene eseguita in sequenza per il numero di tipi di connessione impostato in <CHostData>.ConnectionTypes per enumerare tutti i tipi di connessione.
Booleano = <CHostData>.FindConnectionType(ConnectionType)	Booleano	Restituisce TRUE se il tipo di connessione specificato viene trovato nell'elenco dei tipi di connessione disponibili.
Intero = <CHostData>.MaxAssignedConnections	Intero	Restituisce il numero massimo di tipi di connessione che possono essere attivi su questo host.
Stringa = <CHostData>.FirstAssignedConnection	Stringa	Restituisce il primo tipo di connessione attivo assegnato.
Stringa = <CHostData>.NextAssignedConnection	Stringa	Restituisce il successivo tipo di connessione attivo assegnato.
Booleano = <CHostData>.FindAssignedConnection(ConnectionType)	Booleano	Restituisce TRUE se il tipo di connessione trasmesso viene trovato nell'elenco dei tipi di connessione attivi.

Metodo AssignConnection(ConnectionType)

Il metodo AssignConnection(ConnectionType) inserisce il tipo di connessione richiesto nell'elenco di tipi di connessione assegnati dell'oggetto host e lo imposta come tipo corrente per l'elaborazione delle successive chiamate a metodi specifici per le periferiche.

Se il tipo di connessione richiesto è già presente nell'elenco delle connessioni assegnate, l'elenco non viene modificato. Solo il tipo di connessione corrente viene cambiato nel tipo richiesto. Di norma, il metodo AssignConnection viene richiamato sullo stesso oggetto più volte durante un'operazione di recupero e impostazione di valori specifici di una connessione.

La funzione AssignConnection restituisce TRUE se il tipo di connessione trasmesso esiste sul computer e viene assegnato correttamente oppure risulta già assegnato. Viene restituito FALSE se il tipo di connessione richiesto non esiste sul computer oppure se il numero corrente di connessioni assegnate è già il massimo consentito.

Un oggetto host di pcAnywhere può supportare fino a due tipi di connessione assegnati. Se si tenta di superare questo limite, il metodo AssignConnection restituisce FALSE.

La seguente tabella definisce il parametro di AssignConnection:

Parametro	Descrizione
ConnectionType	Il nome di un tipo di periferica di connessione da assegnare, come stringa

La seguente tabella definisce il valore restituito di AssignConnection:

Valore restituito	Descrizione
Booleano	TRUE se il tipo di periferica è disponibile e il numero massimo consentito di connessioni assegnate non è già stato raggiunto.

Metodo UnassignConnection(ConnectionType)

Il metodo UnassignConnection(Connection Type) restituisce TRUE se il tipo di connessione trasmesso viene rimosso con esito positivo dall'elenco dei tipi di connessione assegnati.

La seguente tabella definisce il parametro di UnassignConnection:

Parametro	Descrizione
ConnectionType come stringa	Il nome di un tipo di periferica di connessione di cui annullare l'assegnazione.

La seguente tabella definisce il valore restituito di UnassignConnection:

Valore restituito	Descrizione
Booleano	TRUE se l'assegnazione del tipo di periferica viene annullata con esito positivo.

Proprietà di composizione

La seguente tabella definisce le proprietà e i valori restituiti per impostare le proprietà di composizione per le connessioni tramite modem.

In [Tabella 2-21](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-21 Proprietà e parametri per le proprietà di composizione del modem

Proprietà	Parametro	Descrizione
<CHostData>.AreaCode(String)	Stringa	Imposta l'indicativo di località per le connessioni modem.
<CHostData>.CountryCode(String)	Stringa	Imposta l'indicativo di paese per le connessioni modem.

La seguente tabella definisce le proprietà e i valori restituiti per le proprietà di composizione del modem.

In [Tabella 2-22](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-22 Proprietà e valori restituiti per le proprietà di composizione del modem

Proprietà	Valore restituito	Descrizione
Stringa = <CHostData>.AreaCode	Stringa	Restituisce l'indicativo di località.
Stringa = <CHostData>.CountryCode	Stringa	Restituisce l'indicativo di paese.
Intero = <CHostData>.CountryCodes	Intero	Restituisce il numero di indicativi di paese disponibili.

Proprietà	Valore restituito	Descrizione
Stringa= <CHostData>.FirstCountryCode	Stringa	Restituisce il primo indicativo di paese disponibile elencato nel sistema operativo.
Stringa = <CHostData>.NextCountryCode	Stringa	Restituisce il successivo indicativo di paese disponibile. La chiamata viene eseguita in sequenza per il numero di indicativi di paese impostato in <CHostData>.CountryCodes per enumerare tutti gli indicativi di paese.

Proprietà per le periferiche COM

Nella seguente tabella vengono descritti le proprietà e i parametri che consentono di personalizzare le impostazioni della porta per le connessioni modem e le altre connessioni basate su COM.

In [Tabella 2-23](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-23 Proprietà e parametri di periferiche COM

Proprietà	Parametro	Descrizione
<CHostData>.ComParity(String)	Stringa	Imposta la parità di comunicazione. I valori validi sono: ■ <Nessuna> ■ Dispari ■ Pari ■ Contrassegno ■ Spazio.
<CHostData>.ComFlowControl(String)	Stringa	Imposta il controllo di flusso dei tipi di connessione basata su COM. I valori validi sono: ■ <Nessuna> ■ XONXOFF ■ RTS/CTS ■ Entrambi

Proprietà	Parametro	Descrizione
<CHostData>.ComStartedBy(String)	Stringa	Imposta il valore di avvio dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI). ■ Ricezione di 2 <CR> ■ Risposta modem.
<CHostData>.ComEndedBy(String)	Stringa	Imposta il valore di fine dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI).
<CHostData>.ComSpeed(Long)	Lungo	Imposta il valore della velocità massima della porta COM. I valori validi sono: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

La seguente tabella descrive i proprietà delle periferiche COM e i valori restituiti.
In [Tabella 2-24](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-24 Proprietà e valori restituiti delle periferiche COM

Proprietà	Valore restituito	Descrizione
Stringa = <CHostData>.ComParity	Stringa	Restituisce uno dei seguenti valori come parità di comunicazione: ■ <Nessuna> ■ Dispari ■ Pari ■ Contrassegno ■ Spazio.
Stringa = <CHostData>.ComFlowControl	Stringa	Restituisce l'impostazione del flusso di comunicazione dell'oggetto host. I valori validi sono: ■ <Nessuna> ■ XONXOFF ■ RTS/CTS ■ Entrambi
Stringa = <CHostData>.ComStartedBy	Stringa	Restituisce il controllo dell'avvio di comunicazione dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI). ■ Ricezione di 2 <CR> ■ Risposta modem.
Stringa = <CHostData>.ComEndedBy	Stringa	Restituisce il controllo della fine comunicazione dei tipi di connessione basata su COM. I valori validi sono: ■ Sempre connesso ■ Rileva portante (DCD) ■ Pronto per inviare (CTS) ■ Set dati pronto (DSR) ■ Indicatore squillo (RI).

Proprietà	Valore restituito	Descrizione
Lungo = <CHostData>.ComSpeed	Lungo	Restituisce l'impostazione corrente della velocità di comunicazione dell'oggetto host. I valori validi sono: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Proprietà per le periferiche NetBIOS

La seguente tabella definisce la proprietà e il parametro le periferiche di rete di NetBIOS.

In [Tabella 2-25](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-25 Proprietà e parametro per NetBIOS

Proprietà	Parametro	Descrizione
<CHostData>.LanaNumber(Integer)	Intero	Imposta il numero LANA (LAN Adapter) per le connessioni NetBIOS.

La seguente tabella definisce la proprietà e il valore restituito le periferiche di rete di NetBIOS.

In [Tabella 2-26](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-26 Valori restituiti per NetBIOS

Proprietà	Valore restituito	Descrizione
Intero = <CHostData>.LanaNumber	Intero	Restituisce l'impostazione corrente del numero LANA (LAN Adapter) per le connessioni NetBIOS.

Proprietà per le periferiche ISDN mediante CAPI 2.0

La seguente tabella definisce le proprietà e i parametri per le connessioni europee ISDN.

In [Tabella 2-27](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-27 Proprietà e parametri per ISDN mediante CAPI 2.0

Proprietà	Parametro	Descrizione
<CHostData>.CapiChannelBonding(Bool)	Booleano	Attiva o disattiva il collegamento a canale per le periferiche ISDN CAPI.
<CHostData>.CapiExtensions(String)	Stringa	Imposta eventuali estensioni CAPI aggiuntive necessarie per le comunicazioni.

Nella [Tabella 2-28](#) vengono definiti le proprietà e i valori restituiti per le connessioni ISDN europee.

Tabella 2-28 Proprietà e valori restituiti per ISDN mediante CAPI 2.0

Proprietà	Valore restituito	Descrizione
Booleano= <CHostData>.CapiChannelBonding	Booleano	Restituisce l'impostazione corrente del collegamento a canale per le periferiche ISDN CAPI nell'oggetto host.
Stringa = <CHostData>.CapiExtensions	Stringa	Restituisce l'elenco corrente delle estensioni CAPI dell'oggetto host.

oggetto CHostDataEx

L'oggetto CHostDataEx contiene le stesse funzionalità dell'oggetto CHostData, più alcune funzionalità aggiuntive.

La seguente tabella descrive le proprietà e i parametri.

In [Tabella 2-29](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-29 Proprietà e parametri per CHostDataEx

Proprietà	Parametro	Descrizione
<CHostData>.CryptPrivateKey(String)	Stringa	Imposta il nome del contenitore della chiave privata da utilizzare.
<CHostData>.CryptCommonName(String)	Stringa	Imposta il nome comune della chiave privata da utilizzare.
<CHostData>.CryptReqLevel(Byte)	Byte	Imposta il livello di crittografia. I valori validi sono: ■ -1: Nessuna ■ 0: pcAnywhere ■ 1: Simmetrica ■ 2: Chiave pubblica
<CHostData>.CryptRefuseLower(Bool)	Booleano	Definisce se il computer host consente connessioni da un computer remoto che utilizza un livello di crittografia inferiore.
<CHostData>.CallersPath(String)	Stringa	Imposta il percorso completamente qualificato dei file dei chiamanti.
<CHostData>.ConfirmConnect(Bool)	Booleano	Definisce se all'utente host viene chiesto di confermare le connessioni.
<CHostData>.ConfirmTimeout(Byte)	Byte	Imposta il numero di secondi prima che scada la richiesta di conferma del limite di tempo della connessione.
<CHostData>.ConfirmDeny(Bool)	Booleano	Definisce se la connessione deve essere terminata in caso di scadenza della richiesta di conferma del limite di tempo della connessione.
<CHostData>.PwCaseSensitive(Bool)	Booleano	Forza l'uso di password che distinguono tra lettere maiuscole e minuscole.
<CHostData>.PwAttempts(Byte)	Byte	Imposta il numero di tentativi di accesso non riusciti consecutivi consentiti prima che venga chiusa la connessione.

Proprietà	Parametro	Descrizione
<CHostData>.PwTimeout(Byte)	Byte	Imposta il numero di minuti a disposizione dell'utente per completare l'accesso prima che venga interrotta la connessione.
<CHostData>.ActiveKbds(Byte)	Byte	Definisce il mouse e la tastiera attivi durante la connessione. I valori validi sono: ■ 0: Host e remoto ■ 1: Host ■ 2: Remoto
<CHostData>.InactiveTimeout(Byte)	Byte	Imposta il numero di minuti in cui la tastiera e il mouse possono rimanere inattivi prima che venga chiusa la connessione.
<CHostData>.LockSystemWhileWait(Bool)	Booleano	Imposta l'opzione di blocco del computer host all'avvio.
<CHostData>.MinimizeOnLaunch(Bool)	Booleano	Imposta l'opzione di avvio dell'host ridotto a icona.
<CHostData>.RunAsService(Bool)	Booleano	Attiva l'esecuzione dell'host come servizio.
<CHostData>.ConnLostWait(Byte)	Byte	Imposta il numero di minuti di attesa prima di consentire la connessione di un altro chiamante.
<CHostData>.ConnLostHostOpts(Bool)	Booleano	Definisce se attendere un'altra connessione o annullare l'host in caso di chiusura anomala della sessione. I valori validi sono: ■ FALSE: attende un'altra connessione. ■ TRUE: annulla l'host.
<CHostData>.EnableConnLostSecurity(Bool)	Booleano	Attiva o disattiva le opzioni di sicurezza per la chiusura della sessione chiuse in modo anomalo.

Proprietà	Parametro	Descrizione
<CHostData>.AuthenticationType(Byte)	Byte	Imposta il tipo di autenticazione. I valori validi sono: ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS (servizi Active Directory) ■ 7: LDAP di Microsoft ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: SecurID RSA L'autenticazione pcAnywhere viene utilizzata per impostazione predefinita se non viene impostata alcuna autenticazione o se il valore impostato non è valido, ad esempio il tipo di autenticazione non è disponibile.
<CHostData>.ConnLostSecurity(Byte)	Byte	Imposta le opzioni di sicurezza per la gestione della fine di una sessione terminata in modo anomalo. I valori validi sono: ■ 1: Scollega utente ■ 2: Riavvia computer host ■ 3: Blocca computer
<CHostData>.CallbkDelay(Byte)	Byte	Imposta il numero di secondi di attesa prima che il modem host richiami il remoto.
<CHostData>.EndSessHostOpts(Bool)	Booleano	Definisce se l'host attende un'altra connessione o viene annullato dopo la fine di una sessione terminata in modo normale. I valori validi sono: ■ FALSE: attende la connessione successiva ■ TRUE: annulla l'host

Proprietà	Parametro	Descrizione
<CHostData>.EnableEndSessSecurity(Bool)	Booleano	Attiva o disattiva le opzioni di sicurezza per la fine di una sessione terminata in modo normale.
<CHostData>.EndSessSecurity(Byte)	Byte	Imposta le opzioni di sicurezza per la fine di una sessione terminata in modo normale. I valori validi sono: ■ 1: Scollega utente ■ 2: Riavvia computer host ■ 3: Blocca computer
<CHostData>.BlankHost(Bool)	Booleano	Imposta l'azzeramento dello schermo nel computer host. Alcune schede video non supportano questa opzione.
<CHostData>.AllowRemoteMouse(Bool)	Booleano	Disattiva l'uso del mouse nel computer remoto durante una sessione.
<CHostData>.RebootOnDisconnect(Bool)	Booleano	Forza il riavvio del computer dopo qualsiasi chiusura della sessione se il valore è TRUE.
<CHostData>.PasswordAfterDisc(Bool)	Booleano	Scollega l'utente dopo la chiusura della sessione.
<CHostData>.LogFailures(Bool)	Booleano	Registra i tentativi di immissione di password non riusciti.
<CHostData>.AllowDriveSecurity(Bool)	Booleano	Attiva le opzioni di sicurezza dell'unità. Questa impostazione è valida solo nei file system NTFS.
<CHostData>.UseDirectoryServices(Bool)	Booleano	Consente l'uso di servizi di directory per l'autenticazione.
<CHostData>.DirectoryServiceEntry(String)	Stringa	Imposta i valori relativi ai servizi di directory.

La seguente tabella descrive le proprietà e i valori restituiti per l'oggetto CHostDataEx.

In [Tabella 2-30](#), sostituire le informazioni tra parentesi angolari con i valori effettivi.

Tabella 2-30 Proprietà e valori restituiti per CHostDataEx

Proprietà	Valore restituito	Descrizione
Stringa = <CHostDataEx>.CryptPrivateKey	Stringa	Restituisce il contenitore della chiave privata attualmente attivo.
Stringa = <CHostDataEx>.CryptCommonName	Stringa	Restituisce il nome comune del contenitore della chiave privata attivo.
Byte = <CHostDataEx>.CryptReqLevel	Byte	Restituisce una delle seguenti impostazioni di crittografia: ■ -1: Nessuna ■ 0: pcAnywhere ■ 1: Simmetrica ■ 2: Chiave pubblica
Booleano = <CHostDataEx>.CryptRefuseLower	Booleano	Restituisce il valore dell'impostazione per negare il livello di crittografia inferiore.
Stringa = <CHostDataEx>.CallersPath	Stringa	Restituisce il percorso attualmente attivo dei file dei chiamanti.
Booleano = <CHostDataEx>.ConfirmConnect	Booleano	Restituisce il valore dell'impostazione per la conferma della connessione.
Byte = <CHostDataEx>.ConfirmTimeout	Byte	Restituisce il valore dell'impostazione per negare il livello di crittografia inferiore.
Booleano = <CHostDataEx>.ConfirmDeny	Booleano	Restituisce il valore dell'impostazione per la disconnessione in caso di timeout.
Booleano = <CHostDataEx>.PwCaseSensitive	Booleano	Restituisce il valore dell'impostazione per la creazione di password che distinguono tra lettere maiuscole e minuscole.
Byte = <CHostDataEx>.PwAttempts	Byte	Restituisce il valore dell'impostazione di limitazione dei tentativi di accesso per ogni chiamata.
Byte = <CHostDataEx>.PwTimeout	Byte	Restituisce il valore dell'impostazione relativa al limite di tempo per il completamento dell'accesso (in minuti).

Proprietà	Valore restituito	Descrizione
Byte = <CHostDataEx>.ActiveKbds	Byte	Restituisce le impostazioni relative alla tastiera e al mouse attivi. I valori validi sono: ■ 0: Host e remoto ■ 1: Host ■ 2: Remoto
Byte = <CHostDataEx>.InactiveTimeout	Byte	Restituisce il numero di minuti di attesa prima della disconnessione se scade il tempo limite di inattività.
Booleano = <CHostDataEx>.LockSystemWhileWait	Booleano	Restituisce l'impostazione relativa al blocco del computer host all'avvio.
Booleano = <CHostDataEx>.MinimizeOnLaunch	Booleano	Restituisce l'impostazione relativa all'avvio dell'host ridotto a icona.
Booleano = <CHostDataEx>.RunAsService	Booleano	Restituisce l'impostazione relativa all'esecuzione dell'host come servizio.
Byte = <CHostDataEx>.ConnLostWait	Byte	Restituisce il numero di minuti di attesa prima di consentire un'altra connessione.
Booleano = <CHostDataEx>.ConnLostHostOpts	Booleano	Restituisce un valore che indica se l'host attende un'altra connessione o viene annullato in caso di chiusura anomala della sessione. I valori validi sono: ■ FALSE: attende un'altra connessione ■ TRUE: annulla l'host
Booleano = <CHostDataEx>.ConnLostWait	Booleano	Restituisce il valore dell'opzione di sicurezza impostata per la gestione di una chiusura anomala della sessione.

Proprietà	Valore restituito	Descrizione
Byte = <CHostDataEx>.AuthenticationType	Byte	Restituisce il riferimento numerico del tipo di autenticazione. I valori validi sono: ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS (servizi Active Directory) ■ 7: LDAP di Microsoft ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: SecurID RSA L'autenticazione pcAnywhere viene utilizzata per impostazione predefinita se non viene impostata alcuna autenticazione o se il valore impostato non è valido, ad esempio il tipo di autenticazione non è disponibile.
Byte = <CHostDataEx>.ConnLostSecurity	Byte	Restituisce la rappresentazione numerica del livello di sicurezza impostato per la gestione di una chiusura anomala della sessione. I valori validi sono: ■ 1: Scollega utente ■ 2: Riavvia computer host ■ 3: Blocca computer
Byte = <CHostDataEx>.CallbkDelay	Byte	Restituisce il numero di secondi di attesa prima che il modem host richiami il remoto.
Booleano = <CHostDataEx>.EndSessHostOpts	Booleano	Restituisce un valore che indica se l'host attende un'altra connessione o viene annullato dopo la fine di una sessione terminata in modo normale. I valori validi sono: ■ FALSE: attende la connessione successiva ■ TRUE: annulla l'host

Proprietà	Valore restituito	Descrizione
Booleano = <CHostDataEx>.EnableEndSessSecurity	Booleano	Restituisce un valore che indica se le opzioni di sicurezza relative alla fine della sessione sono attivate per una chiusura normale della sessione.
Byte = <CHostDataEx>.EndSessSecurity	Byte	Restituisce l'opzione di sicurezza impostata per una chiusura normale della sessione. I valori validi sono: ■ 1: Scollega utente ■ 2: Riavvia computer host ■ 3: Blocca computer
Booleano = <CHostDataEx>.BlankHost	Booleano	Restituisce l'opzione di azzeramento dello schermo impostata.
Booleano = <CHostDataEx>.AllowRemoteMouse	Booleano	Restituisce un valore che indica se l'utente remoto dispone del controllo della tastiera e del mouse durante una sessione.
Booleano = <CHostDataEx>.RebootOnDisconnect	Booleano	Restituisce un valore che indica se il computer host viene riavviato dopo la chiusura della sessione.
Booleano = <CHostDataEx>.PasswordAfterDisc	Booleano	Restituisce un valore che indica se l'utente host viene scollegato dopo la chiusura della sessione.
Booleano = <CHostDataEx>.LogFailures	Booleano	Restituisce un valore che indica se l'opzione relativa alla registrazione degli errori di password è impostata.
Booleano = <CHostDataEx>.AllowDriveSecurity	Booleano	Restituisce un valore che indica se è attivata l'opzione di sicurezza dell'unità.
Booleano = <CHostDataEx>.UseDirectoryServices	Booleano	Restituisce un valore che indica se è attivata l'opzione relativa ai servizi di directory.
Stringa = <CHostDataEx>.DirectoryServiceEntry	Stringa	Restituisce le impostazioni relative ai servizi di directory.

Codice di esempio di Visual Basic per le funzionalità dell'host

Il codice di esempio di Visual Basic riportato di seguito consente di recuperare un oggetto di dati host e di modificarne le proprietà.

```
Private Sub Command1_Click()  
Dim HostDataManager as Object  
Dim HostData as Object  
Dim s as string  
  
'Creazione dell'oggetto CHostDataManager  
Set HostDataManager = CreateObject(WINAWSVR.BEHOSTDATAMANAGER)  
  
'Visualizzazione e cambiamento della directory corrente  
s = HostDataManager.CurrentDirectory()  
MsgBox (s)  
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")  
s = HostDataManager.CurrentDirectory()  
MsgBox (s)  
'Recupero dell'oggetto di dati remoto  
Set HostData = HostDataManager.RetrieveObject("pod.BHF", 2, 0)  
'Visualizzazione di alcune proprietà  
s = HostData.AreaCode()  
MsgBox (s)  
s = HostData.PhoneNumber()  
MsgBox (s)  
'Impostazione di alcune proprietà  
RemoteData.HostData = "212"  
RemoteData.HostData = "555-5555"  
  
'Scrittura dell'oggetto sul disco  
HostData.WriteObject (0)  
End Sub
```

Nell'esempio riportato di seguito vengono utilizzati i metodi FindFirst e FindNext per visualizzare un file host presente in una directory.

```
Private Sub Command5_Click()  
  
Dim HostDataManager as Object  
Dim HostData as Object  
Dim s as string  
  
Set HostDataManager = CreateObject("WINAWSVR.BEHOSTDATAMANAGER")  
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")  
HostDataManager.FindFirst "*", s  
MsgBox (s)  
HostDataManager.FindNext s  
MsgBox (s)
```

End Sub

funzioni Awrem32

Le funzioni di Awrem32 forniscono i parametri e i valori restituiti per la gestione delle connessioni tra computer host e computer remoti.

Funzione Awrem32	Parametro o valore restituito	Descrizione
awConnect(FileName)	Parametro = Nome come stringa	Il nome completamente qualificato del file .chf contenente le informazioni relative al computer host
awConnect	Valore restituito = booleano	TRUE se il comando viene eseguito
awDisconnect()	Valore restituito = booleano	Una volta eseguita la chiamata alla funzione, il programma deve eliminare l'oggetto (in C++: delete IAwrem32X*, in VB: set ObjectName = Nothing;).
FileXferFromHost(HostFile)	Parametro = HostFile come stringa	Contiene il nome e il percorso completamente qualificati del file da copiare dal computer host.
FileXferFromHost(RemoteFile)	Parametro = RemoteFile come stringa	Contiene il nome file e il percorso di destinazione completamente qualificati. Non è necessario che le stringhe HostFile e RemoteFile siano identiche
FileXferFromHost	Valore restituito = booleano	TRUE se il comando viene eseguito
FileXferToHost(HostFile)	Parametro = HostFile come stringa	Copia un file dal computer remoto nel computer host. I parametri possono contenere caratteri jolly Il parametro contiene il nome file e il percorso di destinazione completamente qualificati

Funzione Awrem32	Parametro o valore restituito	Descrizione
FileXferToHost(RemoteFile)	Parametro = RemoteFile come stringa	Contiene il nome e il percorso completamente qualificati del file da copiare dal computer remoto. Non è necessario che le stringhe HostFile e RemoteFile siano identiche
CreateFolderOnHost(FolderName)	Parametro = FolderName come stringa	Contiene l'unità e il percorso del computer host in cui creare la cartella.
CreateFolderOnHost	Valore restituito = booleano	TRUE se il comando viene eseguito
ExecuteHostFile(FileName)	Parametro = FileName come stringa	Esegue un file esistente nel computer host. Questa funzione consente di eseguire solo file batch, eseguibili e di comandi. Non vengono eseguiti file associati a eseguibili. Ad esempio, se si esegue un file .doc, Microsoft Word non viene aperto. Il parametro contiene il percorso completamente qualificato del file nel computer host
ExecuteHostFile	Valore restituito = booleano	TRUE se il comando viene eseguito
GetError()	Valore restituito = Stringa	Restituisce l'ultimo errore generato in Awrem32
ConnectionStatus()	Valore restituito = Short	Restituisce lo stato corrente della connessione al computer host I valori possibili sono: ■ -1 = Connessione perduta ■ 0 = Nessuna connessione ■ 1 = Sessione connessa

Definizioni degli oggetti Visual C++

Il capitolo contiene i seguenti argomenti:

- [Informazioni sugli oggetti Visual C++](#)
- [Metodi di CRemoteDataManager](#)
- [oggetto CRemoteData](#)
- [oggetto CRemoteDataEx](#)
- [Metodi di CHostDataManager](#)
- [oggetto CHostData](#)
- [oggetto CHostDataEx](#)
- [funzioni Awrem32](#)

Informazioni sugli oggetti Visual C++

Per supportare Automazione OLE, il Server di automazione di pcAnywhere fornisce i componenti seguenti:

Winawsvr	fornisce le informazioni necessarie per la connessione al Server di automazione di pcAnywhere e l'accesso alle relative interfacce.
Awrem32	fornisce le informazioni necessarie per il controllo delle connessioni di pcAnywhere.

Gli oggetti descritti nella [Tabella 3-1](#) costituiscono il componente Winawsvr.

Tabella 3-1 Oggetti Winawsvr

Oggetto	Descrizione	Riferimenti
CRemoteDataManager	Fornisce i metodi di creazione, apertura, modifica, salvataggio ed eliminazione di oggetti CRemoteData.	Vedere " Metodi di CRemoteDataManager " a pagina 62.
CRemoteData	Definisce i parametri per l'accesso e il controllo della funzionalità del remoto di pcAnywhere.	Vedere " oggetto CRemoteData " a pagina 66. Vedere " oggetto CRemoteDataEx " a pagina 73.
CHostDataManager	Fornisce i metodi di creazione, apertura, modifica, salvataggio ed eliminazione di oggetti CHostData.	Vedere " Metodi di CHostDataManager " a pagina 74.
CHostData	Definisce i parametri per l'accesso e il controllo della funzionalità dell'host di pcAnywhere.	Vedere " oggetto CRemoteData " a pagina 66. Vedere " oggetto CRemoteDataEx " a pagina 73.

Awrem32 comprende un solo oggetto, costituito da otto interfacce per supportare le sessioni di controllo remoto e trasferimento di file.

Vedere "[funzioni Awrem32](#)" a pagina 88.

Alcune funzioni non sono più supportate, ma le definizioni di oggetti vengono comunque fornite per consentire l'utilizzo con le versioni precedenti.

Per le funzioni che richiedono le password, i valori delle password possono essere impostati ma non richiamati. Ciò è per gli scopi di sicurezza.

Metodi di CRemoteDataManager

I metodi di CRemoteDataManager forniscono i parametri e i valori restituiti per l'accesso e il controllo degli oggetti CRemoteData.

Metodo	Parametro o valore restituito	Descrizione
BSTR CurrentDirectory();	Parametro = BSTR	Il percorso completo della directory di dati pcAnywhere corrente in cui sono memorizzati gli oggetti remoti di pcAnywhere

Metodo	Parametro o valore restituito	Descrizione
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Parametro = LPCTSTR lpszNewDirectory Valore restituito = BOOL	Cambia la directory corrente in cui sono memorizzati gli oggetti remoti di pcAnywhere. Il parametro è il nome di una directory esistente Il valore restituito è TRUE se l'esito è positivo
BOOL FindFirst(LPCTSTR lpszPattern, BSTR FAR*);	Parametro = LPCTSTR lpszPattern	Criterio per il nome file utilizzato per filtrare i file di oggetti. Un asterisco [*] consente di trovare tutti i file presenti nella directory corrente.
BOOL FindFirst(pbstrFullQualName);	Parametro = BSTR FAR * pbstrFullQualName	Buffer di ritorno per il percorso completo del file di oggetti remoti corrispondente al criterio specificato
BOOL FindFirst	Valore restituito = BOOL	TRUE se viene trovato un file di oggetti remoti corrispondente al criterio specificato. Il percorso completo del file corrispondente viene memorizzato in pbstrFullQualName
BOOL FindNext(BSTR FAR* pbstrFullQualName);	Parametro = BSTR FAR * pbstrFullQualName Valore restituito = BOOL	Una volta chiamata la funzione FindFirst() per ottenere il nome di un file di oggetti remoti nella directory corrente, è possibile utilizzare FindNext() per individuare il file successivo, se esiste, che corrisponde al criterio specificato Il valore restituito è TRUE se viene trovato un altro file di oggetti remoti corrispondente al criterio specificato nella chiamata a FindFirst(). Il percorso completo del file corrispondente viene memorizzato in pbstrFullQualName
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName	Recupera un oggetto CRemoteData in base al nome file Il parametro è il nome completamente qualificato del file di oggetti remoti da caricare

Metodo	Parametro o valore restituito	Descrizione
LPDISPATCH RetrieveObject(short wAccessMode);	Parametro = short wAccessMode	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password. Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword	La password per l'oggetto. Può essere un valore NULL
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName	Recupera un oggetto CRemoteDataEx in base al nome file Il parametro è il nome completamente qualificato del file di oggetti remoti da caricare
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parametro = short wAccessMode	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password. Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword	La password per l'oggetto. Può essere un valore NULL
LPDISPATCH RetrieveObjectEx	Valore restituito = LPDISPATCH	Puntatore a un oggetto dispatch OLE. Si tratta di un oggetto CRemoteDataEx. Vedere Codice di esempio di VisualC++ per le funzionalità del remoto.

Metodo	Parametro o valore restituito	Descrizione
LPDISPATCH CreateObject(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName Valore restituito = LPDISPATCH	Crea un oggetto CRemoteData e restituisce un puntatore LPDISPATCH che punta a esso. Il parametro è il nome completamente qualificato del file di oggetti remoti per il nuovo oggetto Il valore restituito è un puntatore a un oggetto dispatch OLE. Si tratta di un oggetto CRemoteData. Vedere Codice di esempio di VisualC++ per le funzionalità del remoto.
LPDISPATCH CreateObjectEx(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName Valore restituito = LPDISPATCH	Crea un oggetto CRemoteDataEx e restituisce un puntatore LPDISPATCH che punta ad esso. Il parametro è il nome completamente qualificato del file di oggetti remoti per il nuovo oggetto Il valore restituito è un puntatore a un oggetto dispatch OLE. Si tratta di un oggetto CRemoteDataEx. Vedere Codice di esempio di VisualC++ per le funzionalità del remoto.
BOOL DeleteObject(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName	Elimina un file di oggetti remoti. Il parametro è il nome completamente qualificato del file di oggetti remoti da eliminare
BOOL DeleteObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword	La password per l'oggetto. Può essere un valore NULL
BOOL DeleteObject	BOOL	TRUE se l'oggetto viene eliminato

Metodo	Parametro o valore restituito	Descrizione
BOOL Launch(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName Valore restituito = BOOL	Avvia un file di oggetti remoti, che apre la finestra del terminale remoto di pcAnywhere. Il parametro è il nome completamente qualificato del file di oggetti remoti da avviare Il valore restituito è TRUE se l'oggetto viene avviato correttamente

oggetto CRemoteData

Utilizzare questo oggetto per modificare i dati dell'oggetto remoto.

Metodi Get e Set

I metodi seguenti vengono utilizzati per ottenere e impostare le proprietà dell'oggetto CRemoteData.

Il nome del computer è il nome dell'host di pcAnywhere da richiamare quando si avvia l'oggetto remoto.

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

Il numero di telefono è il numero da comporre per stabilire una connessione modem con un computer host di pcAnywhere.

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

Indica se devono essere utilizzate le proprietà di composizione TAPI (informazioni sulla località) (TRUE) o se è necessario utilizzare la stringa del numero di telefono esattamente come appare (FALSE).

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

Se si devono utilizzare le proprietà di composizione, questo è l'indicativo di località del numero da chiamare.

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

Se si devono utilizzare le proprietà di composizione, questo è il indicativo di paese del numero da chiamare.

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

Indica il numero di tentativi di composizione del numero da effettuare in caso di chiamata non riuscita.

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

Indica il tempo di attesa (in secondi) tra i tentativi di composizione.

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

Indica il nome di accesso da inviare all'host quando si stabilisce una connessione. Se non si specifica questo valore, durante la connessione viene chiesto all'utente di specificare un nome di accesso.

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

Indica la password di accesso da inviare all'host quando si stabilisce una connessione. Se non si specifica questo valore, durante la connessione viene chiesto all'utente di specificare la password.

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

Indica la password per l'oggetto.

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

Indica che è possibile avviare l'oggetto soltanto se si utilizza la password (TRUE).

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

Indica che è possibile visualizzare l'oggetto soltanto se si specifica la password corretta (TRUE).

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

Indica che è possibile scrivere l'oggetto soltanto se si specifica la password corretta (TRUE).

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

Consente di stabilire se le sessioni che utilizzano questo oggetto devono essere registrate.

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

Consente di stabilire se le sessioni che utilizzano questo oggetto devono essere registrate dall'inizio.

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

Indica il nome del file di registrazione per le sessioni che utilizzano questo oggetto.

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

Imposta i tipi di connessione.

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

I tipi di connessione disponibili sono:

- COM1
- COM2
- COM3
- COM4
- ISDN tramite CAPI 2.0
- NetBIOS
- SPX
- TAPI PREDEFINITA.
- TCP/IP

È anche possibile utilizzare il nome di una periferica TAPI come tipo di connessione. TAPI PREDEFINITA utilizza la prima periferica TAPI rilevata nel sistema. Per usare una periferica TAPI specifica, utilizzare

FirstConnectionType()/NextConnectionType() per ricercare le periferiche disponibili.

Metodi relativi ai dettagli degli oggetti remoti

Se un oggetto remoto viene assegnato a un tipo di connessione, i dettagli relativi alla periferica vengono impostati su valori predefiniti validi.

Per i tipi di connessione seguenti sono disponibili opzioni di configurazione avanzate che possono essere impostate nell'applicazione:

- COM, periferiche
- Periferiche NetBIOS
- Periferiche ISDN tramite CAPI 2.0 (solo ISDN europea).

Dettagli delle periferiche COM

Imposta il livello di parità della comunicazione.

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

I valori di parità delle comunicazioni sono i seguenti:

- Nessuna
- Dispari
- Pari
- Contrassegno
- Spazio.

Imposta il livello di controllo del flusso.

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

I valori di controllo del flusso sono i seguenti:

- <Nessuna>
- XONXOFF
- RTS/CTS
- ENTRAMBI.

Imposta il valore di avvio della connessione.

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

I valori di avvio della connessione sono i seguenti:

- Sempre connesso
- Rileva portante (DCD)
- Pronto per inviare (CTS)
- Set dati pronto (DSR)
- Indicatore squillo (RI).
- Ricezione di 2 <CR>'
- Risposta modem.

Imposta i valori di fine della connessione.

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

I valori di fine della connessione sono i seguenti:

- Sempre connesso
- Rileva portante (DCD)
- Pronto per inviare (CTS)
- Set dati pronto (DSR)
- Indicatore squillo (RI).

Imposta la velocità di connessione.

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

I valori della velocità di connessione sono i seguenti:

- 110
- 300
- 600
- 1200
- 2400
- 4800
- 9600

- 19200
- 38400
- 57600
- 115200

Dettagli delle periferiche NetBIOS

Imposta il numero LANA (LAN Adapter) da utilizzare per questa connessione.

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

Dettagli delle periferiche ISDN tramite CAPI 2.0

Attiva il collegamento a canale (due canali ISDN per una connessione) se il valore è TRUE.

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

Imposta eventuali estensioni CAPI aggiuntive necessarie per le comunicazioni.

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

Metodi dell'oggetto remoto

Nella tabella seguente sono riportati i metodi comuni dell'oggetto remoto, che non vengono utilizzati per ottenere e impostare le proprietà:

Metodo	Parametro o valore restituito	Descrizione
short ConnectionTypes();	Valore restituito = Short	Il numero di tipi di connessione disponibili nel computer in uso.

Metodo	Parametro o valore restituito	Descrizione
BSTR FirstConnectionType(); e BSTR NextConnectionType();	Valore restituito = BSTR	Le funzioni FirstConnectionType() e NextConnectionType() vengono utilizzate per passare da un tipo di connessione disponibile all'altro. Queste funzioni restituiscono un BSTR, che è il nome di un tipo di connessione disponibile. È possibile utilizzare questi tipi di connessione restituiti con la funzione SetConnectionType(). Il valore restituito è il nome di un tipo di periferica di connessione supportato
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Parametro = LPCTSTR lpszConnectionTyp Valore restituito = BOOL	Il nome di un tipo di periferica di connessione. Il valore restituito è TRUE se il tipo di connessione trasmesso esiste sul computer
short CountryCodes();	Valore restituito = Short	Restituisce il numero di indicativi di paese disponibili.
BSTR FirstCountryCode(); e BSTR NextCountryCode();	Valore restituito = BSTR	Le funzioni FirstCountryCode() e NextCountryCode() vengono utilizzate per passare da un indicativo di paese disponibile all'altro. Queste funzioni restituiscono un BSTR, che è il nome di un indicativo paese disponibile. È possibile utilizzare questi valori restituiti con la funzione SetCountryCode() Il valore restituito è la prima stringa di indicativo paese o quella successiva
BOOL ReadObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword Valore restituito = BOOL	Legge i dati dell'oggetto dal file di oggetto remoto. Il parametro è la password dell'oggetto Il valore restituito è TRUE se l'oggetto viene letto correttamente

Metodo	Parametro o valore restituito	Descrizione
BOOL WriteObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword Valore restituito = BOOL	Scrive i dati dell'oggetto nel file di oggetto remoto. Il parametro è la password dell'oggetto Il valore restituito è TRUE se l'oggetto viene scritto correttamente

oggetto CRemoteDataEx

L'oggetto CRemoteDataEx contiene le stesse funzionalità dell'oggetto CRemoteData ma dispone dei seguenti metodi Get e Set aggiuntivi:

```
BSTR GetPrivateKey(); //Restituisce le informazioni PrivateKey
void SetPrivateKey(LPCTSTR lpszNewValue);
BSTR GetCertificationName(); //Restituisce il nome di certificazione
void SetCertificationName(LPCTSTR lpszNewValue);
short GetEncryptionLevel(); //Restituisce il valore del livello
                                //di crittografia
void SetEncryptionLevel(short nNewValue);
BOOL GetDenyLowerEncrypt(); //Restituisce il valore DenyLowerEncrypt
void SetDenyLowerEncrypt(BOOL bNewValue);
BSTR GetAutoDomain(); //Restituisce il valore AutoDomain
void SetAutoDomain(LPCTSTR lpszNewValue);
```

Codice di esempio di VisualC++ per le funzionalità del remoto

La funzione Visual C++ di esempio riportata di seguito crea un oggetto remoto, imposta il relativo tipo di connessione su TCP/IP, imposta il nome del computer sull'indirizzo TCP/IP trasmesso nella funzione e infine avvia l'oggetto remoto:

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

    CRemoteDataManager remoteDM;

    CRemoteData remoteData;

    // In primo luogo, creare CRemoteDataManager
    remoteDM.CreateDispatch( _T( "WINASVR.RemoteDataManager" ) );
    // In secondo luogo, creare CRemoteData e collegarlo
```

```
remoteData.AttachDispatch( remoteDM.CreateObject("Test", 0) );

// Impostare le proprietà richieste
remoteData.SetConnectionType("TCP/IP");

remoteData.SetComputerName(lpszAddress);

// Salvare i dati dell'oggetto
if (remoteData.WriteObject(0))
{
    // Avviare l'oggetto
    if (remoteData.Launch())
    {
        bReturn = TRUE;
    }
    // Rilasciare l'oggetto remoto.
    remoteData.ReleaseDispatch();

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

    return bReturn;
}
```

Metodi di CHostDataManager

La seguente tabella descrive i metodi CHostDataManager, inclusi i parametri e i valori restituiti per l'accesso e il controllo degli oggetti CHostData:

Metodo	Parametro o valore restituito	Descrizione
BSTR CurrentDirectory();	Valore restituito = BSTR	Restituisce il percorso completo della directory corrente in cui sono memorizzati gli oggetti host di pcAnywhere
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Parametro = LPCTSTR lpszNewDirectory Valore restituito = BOOL	Cambia la directory corrente in cui sono memorizzati gli oggetti host di pcAnywhere Il parametro è il nome di una directory esistente Il valore di ritorno è TRUE se l'esito è positivo

Metodo	Parametro o valore restituito	Descrizione
BOOL FindFirst(LPCTSTR lpszPattern);	Parametro = LPCTSTR lpszPattern Valore restituito = BOOL	Trova il primo file di oggetti host di pcAnywhere (*.bhf) nella directory corrente in base al criterio specificato per i nomi file Il parametro è il criterio per il nome file per filtrare i file di oggetti. Un asterisco [*] consente di trovare tutti i file host presenti nella directory corrente. Il valore restituito è TRUE se un file di oggetti host che corrisponde al criterio specificato viene trovato. Il percorso completo del file corrispondente viene memorizzato in pbstrFullQualName
BOOL FindFirst(BSTR FAR* pbstrFullQualName);	Parametro = BSTR FAR * pbstrFullQualName	Buffer di ritorno per il percorso completo del file di oggetti remoti corrispondente al criterio specificato Il valore restituito è TRUE se un file di oggetti host che corrisponde al criterio specificato viene trovato. Il percorso completo del file corrispondente viene memorizzato in pbstrFullQualName.
BOOL FindNext(BSTR FAR* pbstrFullQualName);	Parametro = BSTR FAR * pbstrFullQualName Valore restituito = BOOL	Una volta richiamata la funzione FindFirst() per ottenere il nome di un file di oggetti host nella directory corrente, è possibile utilizzare FindNext() per individuare il file successivo, se esiste, che corrisponde al criterio specificato Il parametro è il valore del buffer di ritorno per il percorso completo del file di oggetti host corrispondente al criterio specificato nella chiamata originale a FindFirst() Il valore restituito è TRUE se viene trovato un altro file di oggetti host corrispondente al criterio specificato nella chiamata a FindFirst(). Il percorso completo del file corrispondente viene memorizzato in pbstrFullQualName

Metodo	Parametro o valore restituito	Descrizione
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName	Recupera un oggetto CHostData in base al nome file. Il parametro è il nome completamente qualificato del file di oggetti host da caricare
LPDISPATCH RetrieveObject(short wAccessMode);	Parametro = short wAccessMode	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password. Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword	La password per l'oggetto. Può essere un valore NULL
LPDISPATCH RetrieveObject	Valore restituito = LPDISPATCH	Puntatore a un oggetto dispatch OLE. Si tratta di un oggetto CHostData. Vedere Codice di esempio di Visual C++ per le funzionalità dell'host.
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName	Recupera un oggetto CHostDataEx in base al nome file Il parametro è il nome completamente qualificato del file di oggetti host da caricare
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parametro = short wAccessMode	Specifica il modo in cui questo oggetto deve essere utilizzato. Questa impostazione è correlata alla protezione con password. Le opzioni sono le seguenti: ■ 0 = Non specificato ■ 1 = Solo visualizzazione ■ 2 = Visualizzazione e modifica ■ 3 = Esecuzione

Metodo	Parametro o valore restituito	Descrizione
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword	La password per l'oggetto. Può essere un valore NULL
LPDISPATCH RetrieveObjectEx	Valore restituito = LPDISPATCH	Puntatore a un oggetto dispatch OLE. Si tratta di un oggetto CHostDataEx Vedere Codice di esempio di Visual C++ per le funzionalità dell'host.
LPDISPATCH CreateObject(LPCTSTR lpszName);	Parametro = LPCTSTR lpszFQName Valore restituito = LPDISPATCH	Crea un oggetto CHostData e restituisce un puntatore LPDISPATCH che punta ad esso. Il parametro è il nome completamente qualificato del file di oggetti host per il nuovo oggetto Il valore restituito è un puntatore a un oggetto dispatch OLE. Si tratta di un oggetto CHostData Vedere Codice di esempio di Visual C++ per le funzionalità dell'host.
LPDISPATCH CreateObjectEx(LPCTSTR lpszName);	Parametro = LPCTSTR lpszFQName Valore restituito = LPDISPATCH	Crea un oggetto CHostDataEx e restituisce un puntatore LPDISPATCH che punta ad esso. Il parametro è il nome completamente qualificato del file di oggetti host per il nuovo oggetto Il valore restituito è un puntatore a un oggetto dispatch OLE. Si tratta di un oggetto CHostDataEx. Vedere Codice di esempio di Visual C++ per le funzionalità dell'host.
BOOL DeleteObject(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName	Elimina un file di oggetti host Il parametro è il nome completamente qualificato del file di oggetti host da eliminare
BOOL DeleteObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword	La password per l'oggetto. Può essere un valore NULL

Metodo	Parametro o valore restituito	Descrizione
BOOL DeleteObject	Valore restituito = BOOL	TRUE se l'oggetto viene eliminato
BOOL Launch(LPCTSTR lpszFQName);	Parametro = LPCTSTR lpszFQName Valore restituito = BOOL	Avvia un file di oggetti host, che apre la finestra del terminale host di pc Anywhere Il parametro è il nome completamente qualificato del file di oggetti host da avviare Il valore restituito è TRUE se l'oggetto viene avviato correttamente

oggetto CHostData

Utilizzare questo oggetto per modificare i dati dell'oggetto host.

Metodi Get e Set

I metodi seguenti vengono utilizzati per ottenere e impostare le proprietà dell'oggetto CHostData.

Il numero di telefono è il numero da comporre per stabilire una connessione modem con un computer remoto di pcAnywhere.

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

Indica se devono essere utilizzate le proprietà di composizione TAPI (informazioni sulla località) (TRUE) o se è necessario utilizzare la stringa del numero di telefono esattamente come appare (FALSE).

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

Se si devono utilizzare le proprietà di composizione, questo è l'indicativo di località del numero da chiamare.

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

Se si devono utilizzare le proprietà di composizione, questo è il indicativo di paese del numero da chiamare.

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

Indica il numero di tentativi di composizione del numero da effettuare in caso di chiamata non riuscita.

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

Indica il tempo di attesa (in secondi) tra i tentativi di composizione.

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

Consente di stabilire se le sessioni che utilizzano questo oggetto devono essere registrate.

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

Consente di stabilire se le sessioni che utilizzano questo oggetto devono essere registrate dall'inizio.

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

Indica il nome del file di registrazione per le sessioni che utilizzano questo oggetto.

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

Metodi relativi ai dettagli degli oggetti host

Se un oggetto host viene assegnato a un tipo di connessione, i dettagli relativi alle periferiche vengono impostati su valori predefiniti validi.

Per i tipi di connessione seguenti sono disponibili opzioni di configurazione avanzate che possono essere impostate nell'applicazione:

- COM, periferiche
- Periferiche NetBIOS
- Periferiche NASI/NCSI
- Periferiche ISDN tramite CAPI 2.0 (solo ISDN europea).

Dettagli delle periferiche COM

Nel codice riportato di seguito il tipo di connessione richiesto viene inserito nell'elenco dei tipi di connessione assegnati dell'oggetto host e viene impostato come tipo corrente per l'elaborazione delle successive chiamate a metodi specifici per le periferiche:

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

Se il tipo di connessione richiesto è già presente nell'elenco delle connessioni assegnate, l'elenco non viene modificato. Solo il tipo di connessione corrente viene cambiato nel tipo richiesto. Di norma, il metodo `AssignConnection` viene richiamato sullo stesso oggetto più volte durante un'operazione di recupero e impostazione di valori specifici di una connessione.

La funzione `AssignConnection` restituisce `TRUE` se il tipo di connessione trasmesso esiste sul computer e viene assegnato correttamente oppure risulta già assegnato. Viene restituito `FALSE` se il tipo di connessione richiesto non esiste sul computer oppure se il numero corrente di connessioni assegnate è già il massimo consentito.

Un oggetto host di `pcAnywhere` può supportare fino a due tipi di connessione assegnati. Se si tenta di superare questo limite, il metodo `AssignConnection` restituisce `FALSE`.

I tipi di connessione disponibili sono:

- COM1
- COM2
- COM3
- COM4
- SPX
- NetBIOS
- TCP/IP
- ISDN tramite CAPI 2.0
- TAPI PREDEFINITA.

È anche possibile utilizzare il nome di una periferica TAPI come tipo di connessione. `TAPI PREDEFINITA` utilizza la prima periferica TAPI rilevata nel sistema. Per usare una periferica TAPI specifica, utilizzare `FirstConnectionType()` e `NextConnectionType()` per ricercare le periferiche disponibili.

Nel codice riportato di seguito viene annullata l'assegnazione di un tipo di connessione. Una volta eseguita questa operazione, la connessione assegnata

rimanente, se esistente, diventa il tipo di connessione corrente per le successive chiamate a metodi specifici per le periferiche.

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

Imposta i livelli di parità della comunicazione.

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

I valori di parità delle comunicazioni sono i seguenti:

- Nessuna
- Dispari
- Pari
- Contrassegno
- Spazio.

Imposta i livelli di controllo del flusso.

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

I valori di controllo del flusso sono i seguenti:

- <Nessuna>
- XONXOFF
- RTS/CTS
- ENTRAMBI.

Imposta i valori di avvio della connessione.

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

I valori di avvio della connessione sono i seguenti:

- Sempre connesso
- Rileva portante (DCD)
- Pronto per inviare (CTS)
- Set dati pronto (DSR)
- Indicatore squillo (RI).
- Ricezione di 2 <CR>'

- Risposta modem.

Imposta i valori di fine della connessione.

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

I valori di fine della connessione sono i seguenti:

- Sempre connesso
- Rileva portante (DCD)
- Pronto per inviare (CTS)
- Set dati pronto (DSR)
- Indicatore squillo (RI).

Imposta la velocità di connessione.

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

I valori della velocità di connessione sono i seguenti:

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

Dettagli delle periferiche NetBIOS

Imposta il numero LANA (LAN Adapter) da utilizzare per questa connessione.

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

Dettagli delle periferiche NASI/NC SI

Imposta il nome utente per il server NASI.

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

Imposta la password utente per il server NASI.

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

Imposta il nome della sessione NASI.

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

Specifica il server NASI da utilizzare.

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

Specifica il servizio NASI da utilizzare.

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

Specifica la porta NASI da utilizzare.

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

Dettagli delle periferiche ISDN tramite CAPI 2.0

Attiva il collegamento a canale (due canali ISDN per una connessione) se il valore è TRUE.

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

Imposta eventuali estensioni CAPI aggiuntive necessarie per le comunicazioni.

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

Metodi dell'oggetto host

Nella tabella seguente sono riportati i metodi comuni dell'oggetto, che non vengono utilizzati per ottenere e impostare le proprietà:

Metodo	Parametro o valore restituito	Descrizione
short ConnectionTypes();	Valore restituito = Short	Restituisce il numero di tipi di connessione disponibili nel computer in uso.
BSTR FirstConnectionType(); e BSTR NextConnectionType();	Valore restituito = BSTR	Le funzioni FirstConnectionType() e NextConnectionType() vengono utilizzate per passare da un tipo di connessione disponibile all'altro. Queste funzioni restituiscono un BSTR, che è il nome di un tipo di connessione disponibile. È possibile utilizzare questi tipi di connessione restituiti con la funzione SetConnectionType(). Il valore restituito è il nome di un tipo di periferica di connessione supportato
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Parametro = LPCTSTR lpszConnectionTyp Valore restituito = BOOL	Il parametro è il nome di un tipo di periferica di connessione Il valore restituito è TRUE se questo tipo di periferica è disponibile
short MaxAssignedConnections()	Valore restituito = Short	Restituisce il numero massimo di tipi di connessione che possono essere assegnati contemporaneamente (attualmente due).
short AssignedConnections()	Valore restituito = Short	Restituisce il numero di tipi di connessione assegnati.

Metodo	Parametro o valore restituito	Descrizione
BSTR FirstAssignedConnection(); e BSTR NextAssignedConnection();	Valore restituito = BSTR	Le funzioni FirstAssignedConnection() e NextAssignedConnection() vengono utilizzate per passare da una connessione assegnata all'altra. Queste funzioni restituiscono un BSTR, che è il nome di un tipo di connessione assegnato. È possibile utilizzare questi tipi di connessione restituiti con la funzione AssignConnection(). Il valore restituito è il nome di un tipo di periferica di connessione supportato
BOOL FindAssignedConnection (LPCTSTR lpszConnectionType);	Parametro = LPCTSTR lpszConnectionTyp Valore restituito = BOOL	Il parametro è il nome di un tipo di periferica di connessione Il valore restituito è TRUE se questo tipo di periferica è correntemente assegnato
short CountryCodes();	Valore restituito = Short	Restituisce il numero di indicativi di paese disponibili.
BSTR FirstCountryCode(); e BSTR NextCountryCode();	Valore restituito = BSTR	Le funzioni FirstCountryCode() e NextCountryCode() vengono utilizzate per passare da un indicativo di paese disponibile all'altro nell'elenco. Queste funzioni restituiscono un BSTR, che è il nome di un indicativo paese disponibile. È possibile utilizzare questi valori restituiti con la funzione SetCountryCode() Il valore restituito è la prima stringa di indicativo paese o quella successiva
BOOL ReadObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword Valore restituito = BOOL	Legge i dati dell'oggetto dal file di oggetto host. Il parametro è la password dell'oggetto Il valore restituito è TRUE se l'oggetto viene letto correttamente

Metodo	Parametro o valore restituito	Descrizione
BOOL WriteObject(LPCTSTR lpszPassword);	Parametro = LPCTSTR lpszPassword Valore restituito = BOOL	Scrive i dati dell'oggetto nel file di oggetto host. Il parametro è la password dell'oggetto Il valore restituito è TRUE se l'oggetto viene scritto correttamente

oggetto CHostDataEx

L'oggetto CHostDataEx contiene le stesse funzionalità dell'oggetto CHostData, ma dispone dei metodi Get e Set aggiuntivi seguenti:

```
BOOL GetReadProtection();  
void SetReadProtection(BOOL bNewValue);  
BOOL GetWriteProtection();  
void SetWriteProtection(BOOL bNewValue);  
BSTR GetPassword(); //Returns "NON IMPLEMENTATO"  
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); //Imposta 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);
```

Codice di esempio di Visual C++ per le funzionalità dell'host

La funzione Visual C++ di esempio riportata di seguito crea un oggetto host, imposta il relativo tipo di connessione su TCP/IP, imposta il nome del computer sull'indirizzo TCP/IP trasmesso alla funzione e infine avvia l'oggetto host.

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

    CHostDataManager hostDM;
    CHostData hostData;

    // In primo luogo, creare CHostDataManager
    hostDM.CreateDispatch( _T( "WINAWSVR.BeHostDataManager" ) );

    // In secondo luogo, creare CRemoteData e collegarlo
    hostData.AttachDispatch(hostDM.CreateObject("Test", 0) );

    // Impostare le proprietà richieste
    hostData.SetConnectionType("TCP/IP");

    // Salvare i dati dell'oggetto
    if (hostData.WriteObject(0))
    {
        // Avviare l'oggetto
        if (hostData.Launch())
            bReturn = TRUE;
    }

    // Rilasciare l'oggetto host.
    hostData.ReleaseDispatch();

    return (bReturn);
}
```

funzioni Awrem32

La seguente tabella descrive le funzioni Awrem32, inclusi i parametri e i valori restituiti per la gestione delle connessioni fra un host e un computer remoto:

Metodo	Parametro o valore restituito	Descrizione
boolean awConnect(BSTR FileName);	Parametro = Nome come stringa Valore restituito = booleano	Crea la connessione al computer host Il parametro è il nome completamente qualificato del file .chf che contiene le informazioni sul computer host Il valore restituito esegue il comando
boolean awDisconnect();	Valore restituito = booleano	Disconnette il computer host Una volta eseguita la chiamata alla funzione, il programma deve eliminare l'oggetto (in C++: delete IAwrem32*, in VB: set ObjectName = Nothing;)
boolean FileXferFromHost(BSTR HostFile);	Parametro = HostFile come stringa	Copia un file dal computer host nel computer remoto. I parametri possono contenere caratteri jolly Il parametro contiene il nome e il percorso completamente qualificati del file da copiare dal computer host. Non è necessario che le stringhe HostFile e RemoteFile siano identiche
boolean FileXferFromHost(BSTR RemoteFile);	Parametro = RemoteFile come stringa	Copia un file dal computer host nel computer remoto. I parametri possono contenere caratteri jolly Il parametro contiene il nome file e il percorso di destinazione completamente qualificati. Non è necessario che le stringhe HostFile e RemoteFile siano identiche.
boolean FileXferFromHost	Valore restituito = booleano	TRUE se il comando viene eseguito
boolean FileXferToHost(BSTR HostFile);	Parametro = HostFile come stringa	Copia un file dal computer remoto nel computer host. I parametri possono contenere caratteri jolly Il parametro contiene il nome file e il percorso di destinazione completamente qualificati. Non è necessario che le stringhe HostFile e RemoteFile siano identiche

Metodo	Parametro o valore restituito	Descrizione
boolean FileXferToHost(BSTR RemoteFile);	Parametro = RemoteFile come stringa	Copia un file dal computer remoto nel computer host. I parametri possono contenere caratteri jolly Il parametro contiene il nome e il percorso completamente qualificati del file da copiare dal computer remoto. Non è necessario che le stringhe HostFile e RemoteFile siano identiche
boolean FileXferToHost	Valore restituito = booleano	TRUE se il comando viene eseguito
boolean CreateFolderOnHost(BSTR FolderName);	Parametro = FolderName come stringa Valore restituito = booleano	Crea una nuova cartella nel computer host. Questa funzione crea una cartella temporanea nel computer remoto, quindi la copia nel computer host Il parametro contiene l'unità e il percorso del computer host in cui creare la cartella Il valore restituito è TRUE se il comando viene eseguito
boolean ExecuteHostFile(BSTR FileName);	Parametro = FileName come stringa Valore restituito = booleano	Esegue un file esistente nel computer host. Questa funzione consente di eseguire solo file batch, eseguibili e di comandi. Non vengono eseguiti file associati a eseguibili. Ad esempio, se si esegue un file .doc, Microsoft Word non viene aperto. Il parametro contiene il percorso completamente qualificato del file nel computer host Il valore restituito è TRUE se il comando viene eseguito
BSTR GetError();	Valore restituito = Stringa	Restituisce l'ultimo errore generato in Awrem32 come stringa

Metodo	Parametro o valore restituito	Descrizione
short ConnectionStatus();	Valore restituito = Short	Restituisce lo stato corrente della connessione al computer host I valori possibili sono: ■ -1 = Connessione perduta ■ 0 = Nessuna connessione ■ 1 = Sessione connessa

Indice

A

- API, librerie 14
- Automazione OLE.. *Vedere* pcAnywhere, server di automazione
- Automazione, controller
 - Visual Basic, informazioni 12
 - Visual C++, informazioni 13
- Awrem32
 - funzioni 58, 88

C

- C++.. *Vedere* Visual C++
- CAPI, connessioni
 - proprietà host 48, 83
 - proprietà remoto 30, 71
- Cartelle
 - creazione nell'host 59, 90
- CHostData
 - Visual Basic, uso 37
 - Visual C++, uso 78
- CHostDataEx
 - Visual Basic, uso 48
 - Visual C++, uso 86
- CHostDataManager
 - Visual Basic, uso 34
 - Visual C++, uso 74
- Classi, definizioni
 - importazione 14
 - visualizzazione 15
- Codice, esempi
 - Visual C++ 73, 88
- codice, esempi
 - Visual Basic 32, 56
- COM, periferiche
 - dettagli di periferiche host 44, 80
 - dettagli di periferiche remote 26, 69
- Connessione, tipi
 - proprietà remoto 23
- connessione, tipi
 - assegnazione 42
 - assegnazione, annullamento 42

- proprietà host 39, 79, 84
- proprietà remoto 69, 71
- Connessioni
 - avvio 37, 58, 66, 89
 - chiusura 58, 89
 - restituzione dello stato 59, 91
 - restituzione di errori 90
- Connessioni NetBIOS
 - proprietà host 47, 82
 - proprietà remoto 30, 71
- Controllo del flusso 27, 44, 69, 81
- CRemoteData
 - Visual Basic, uso 21
 - Visual C++, uso 66
- CRemoteDataEx
 - Visual Basic, uso 31
 - Visual C++, uso 73
- CRemoteDataManager
 - Visual Basic, uso 18
 - Visual C++, uso 62
- Crittografia, impostazioni
 - host 48, 86
 - remoto 31-32, 73

E

- Errori
 - restituzione 59, 90

F

- File
 - esecuzione nell'host 59, 90
- File, trasferimento
 - dall'host 58, 89
 - nell'host 58, 89

G

- GUID 11

H

Host, oggetti
 avvio 37, 78
 Composizione, proprietà 43
 creazione 77
 dettagli di periferica 79
 directory 34, 74
 eliminazione 36, 77
 metodi 84
 Password 85
 ricerca 35–36, 75, 77

I

Identificatori. *Vedere* GUID
 indicativi di paesi
 host 85
 remoto 72

ISDN

CAPI, connessioni
 proprietà host 48, 83
 proprietà remoto 30, 71

M

modem, connessioni
 host
 Composizione, proprietà 43
 indicativi di paesi 85
 periferiche TAPI 80
 proprietà COM 44, 80
 remoto
 Composizione, proprietà 25
 indicativi di paesi 72
 periferiche TAPI 69
 proprietà COM 26, 69
 Motore remoto
 registrazione automatica 11
 registrazione manuale 12

N

NASI, periferiche 83
 NCSI, periferiche 83

P

Parità 27, 44, 69, 81
 Password
 sugli oggetti host 85
 sugli oggetti remoti 72

pcAnywhere, server di automazione

accesso con Visual Basic 12
 accesso con Visual C++ 14
 esempi di utilizzo 10
 GUID, registrazione 11
 informazioni 10
 tipi, librerie 14

Periferiche. *Vedere* connessione, tipi

periferiche TAPI

host 80
 remoto 69

R

Remoti, oggetti

Composizione, proprietà 25
 creazione 65
 dettagli di periferica 69
 directory 18, 62
 eliminazione 65
 file 19–20, 35, 63–64
 metodi 71
 Password 72

Rete, connessioni

host
 proprietà NetBIOS 47, 82
 remoto
 proprietà NetBIOS 30, 71

S

Server di automazione.. *Vedere* pcAnywhere, server di automazione

Stato

restituzione 59, 91

T

tipi, librerie 14

V

Visual Basic

codice, esempi 32, 56
 funzioni Awrem32 58
 oggetto CHostData 37
 oggetto CHostDataEx 48
 oggetto CHostDataManager 34
 oggetto CRemoteData 21
 oggetto CRemoteDataEx 31
 oggetto CRemoteDataManager 18

- Server di automazione di pcAnywhere,
 - accesso 12

Visual C++

- classi, importazione 14
- Codice, esempi 73, 88
- funzioni Awrem32 88
- inclusione Winawsvr.h 15
- oggetto CHostData 78
- oggetto CHostDataEx 86
- oggetto CHostDataManager 74
- oggetto CRemoteData 66
- oggetto CRemoteDataEx 73
- oggetto CRemoteDataManager 62
- Server di automazione di pcAnywhere,
 - accesso 14

W

Winawsvr

- eseguibile 12
- file di intestazione 15
- oggetti
 - CHostData 37, 78
 - CHostDataEx 48, 86
 - CHostDataManager 34, 74
 - CRemoteData 21, 66
 - CRemoteDataEx 31, 73
 - CRemoteDataManager 18, 62