

OLE-Automatisierungshandbuch

OLE-Automatisierungshandbuch

Rechtlicher Hinweis

Copyright © 2008 Symantec Corporation. Alle Rechte vorbehalten.

Symantec, die Marken Symantec-Logoare oder eingetragene Marken der Symantec Corporation oder seiner verbundenen Unternehmen in den USA und anderen Ländern. Andere Bezeichnungen können Marken anderer Rechteinhaber sein.

Dieses Symantec-Produkt kann Drittanbietersoftware enthalten, für das Symantec gehalten ist, die Drittanbieter zu nennen ("Nicht-Symantec-Programme "). Einige Nicht-Symantec-Programme sind unter Open Source- oder kostenlosen Software-Lizenzen verfügbar. Die Lizenzvereinbarung, welche die Software begleitet, ändert die Rechte oder Verpflichtungen, die Sie unter den Open Source- oder kostenlosen Software-Lizenzen haben, nicht. Lesen Sie bitte den Nicht-Symantec-Hinweis-Anhang zu dieser Dokumentation oder die TPIP Readme-Datei, die dieses Symantec-Produkt begleiten, um weitere Informationen über Nicht-Symantec-Programme zu erhalten.

Das in diesem Dokument beschriebene Produkt wird zu Lizenzbedingungen ausgeliefert, die seine Verwendung, Vervielfältigung, Verteilung und Dekompilierung bzw. Reverse Engineering einschränken. Ohne vorherige schriftliche Genehmigung der Symantec Corporation oder ihrer Lizenzgeber (falls zutreffend) darf kein Teil dieses Dokuments vervielfältigt werden, unabhängig davon, in welcher Form oder mit welchen Mitteln dies geschieht.

DIE DOKUMENTATION WIRD OHNE MÄNGELGEWÄHR BEREITGESTELLT. ALLE AUSDRÜCKLICHEN UND STILLSCHWEIGENDEN VORAUSSETZUNGEN, DARSTELLUNGEN UND GEWÄHRLEISTUNGEN, EINSCHLIESSLICH DER STILLSCHWEIGENDEN GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT, EIGNUNG FÜR EINEN BESTIMMTEN ZWECK ODER NICHT-BEEINTRÄCHTIGUNG, SIND AUSGESCHLOSSEN, AUSSER IN DEM UMFANG, IN DEM SOLCHE HAFTUNGSAUSSCHLÜSSE ALS NICHT RECHTSGÜLTIG ANGESEHEN WERDEN. SYMANTEC CORPORATION IST NICHT VERANTWORTLICH FÜR INDIREKTE ODER FOLGESCHÄDEN IM ZUSAMMENHANG MIT DER BEREITSTELLUNG ODER VERWENDUNG DIESER DOKUMENTATION. DIE INFORMATIONEN IN DIESER DOKUMENTATION KÖNNEN SICH OHNE VORHERIGE ANKÜNDIGUNG ÄNDERN.

Die lizenzierte Software und Dokumentation gelten als „kommerzielle Computersoftware“ gemäß FAR Abs. 12.212 und unterliegen, je nach Anwendbarkeit, eingeschränkten Rechten gemäß der Definition in FAR Abs. 52.227-19 „Commercial Computer Software - Restricted Rights“ und gemäß DFARS Abs. 227.7202, „Rights in Commercial Computer Software or Commercial Computer Licensed Software Documentation“ sowie etwaigen Nachfolbestimmungen. Jegliche Verwendung, Modifizierung, Reproduktion, Vorführung, Demonstration oder Offenlegung der unter Lizenz bereitgestellten Software und Dokumentation durch die Regierung der USA erfolgt nur in Übereinstimmung mit dieser Vereinbarung.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

<http://www.symantec.de>

Hergestellt in Irland

10 9 8 7 6 5 4 3 2 1

Technischer Support

Der technische Support von Symantec behält Support-Center global bei. Die primäre Rolle des technischen Supports ist, auf bestimmte Anfragen über Produktmerkmale und Funktionalität zu reagieren. Der technische Support erstellt auch den Inhalt für die Online-Supportdatenbank. Der technische Support arbeitet mit den anderen Funktionsbereichen innerhalb von Symantec zusammen, um Ihre Fragen rechtzeitig zu beantworten. Beispielsweise arbeitet der technische Support mit den Abteilungen Product Engineering und Symantec Security Response zusammen, um Meldungsdienste und Aktualisierungen von Virendefinitionen bereitzustellen.

Symantec bietet folgende Wartungsangebote:

- Eine Vielzahl von Support-Optionen, mit denen Sie den entsprechenden Servicebedarf für Unternehmen aller Größen flexibel auswählen können.
- Telefon- und Web-Support, der schnelle Antworten und aktuelle Informationen zur Verfügung stellt
- Upgrade-Sicherstellung, die automatischen Software-Upgrade-Schutz bietet
- Globaler Support rund um die Uhr (24 Stunden pro Tag, 7 Tage in der Woche)
- Erweiterte Funktionen, einschließlich Account-Verwaltungsdienste

Informationen zu Wartungsprogrammen von Symantec finden Sie auf der Symantec-Website unter der folgenden Adresse:

www.symantec.com/techsupp/

Kontaktieren des technischen Supports

Kunden mit einer aktuellen Wartungsvereinbarung können unter folgender URL auf Informationen des technischen Supports zugreifen:

www.symantec.com/techsupp/

Bevor Sie sich an den technischen Support wenden, stellen Sie sicher, dass die in Ihrer Produktdokumentation aufgeführten Systemanforderungen erfüllt sind. Sie sollten auch am Computer sein, an dem das Problem auftrat, falls das Replizieren des Problems notwendig ist.

Wenn Sie den technischen Support kontaktieren, halten Sie bitte die folgenden Informationen bereit:

- Produktversion
- Hardwareinformationen
- Verfügbarer Speicher, Festplattenspeicherplatz und NIC-Informationen

- Betriebssystem
- Version und Patch
- Netzwerktopologie
- Router-, Gateway- und IP-Adressinformationen
- Problembeschreibung:
 - Fehlermeldungen und Protokolldateien
 - Fehlerbehebung, die vor der Kontaktaufnahme mit Symantec durchgeführt wurde
 - Aktuelle Änderungen an Softwarekonfiguration und Netzwerk

Lizenzierung und Registrierung

Wenn Ihr Symantec-Produkt eine Registrierung oder einen Lizenzschlüssel erfordert, rufen Sie die Webseite des technischen Supports unter folgender URL auf:

www.symantec.com/techsupp/

Kundenservice

Informationen zum Kundenservice sind unter folgender URL verfügbar:

www.symantec.com/techsupp/

Der Kundenservice steht zur Verfügung, um Hilfestellung bei folgenden Problemen zu geben:

- Fragen bezüglich der Produktlizenzierung oder der Zuweisung von Seriennummern (Serialisierung)
- Aktualisierungen der Produktregistrierung wie z.B. Adress- oder Namensänderungen
- Allgemeine Produktinformationen (Funktionen, Verfügbarkeit in Bezug auf die Sprache, lokale Händler)
- Neueste Informationen zu Produktaktualisierungen und -upgrades
- Informationen zu Upgrade-Sicherstellung und Wartungsverträgen
- Informationen über die Symantec Buying Programs
- Ratschläge zu den technischen Support-Optionen von Symantec
- Nicht technische Presales-Fragen
- Probleme mit CD-ROMs oder Handbüchern

Wartungsvereinbarungsressourcen

Wenn Sie Symantec bezüglich einer vorhandenen Wartungsvereinbarung kontaktieren möchten, wenden Sie sich wie folgt an das Verwaltungsteam für Wartungsvereinbarungen für Ihre Region:

Asien-Pazifik und Japan	contractsadmin@symantec.com
Europa, Naher Osten und Afrika	semea@symantec.com
Nordamerika und Lateinamerika	supportsolutions@symantec.com

Zusätzliche Unternehmensdienste

Symantec bietet umfassende Dienste an, damit Sie Ihre Investition in Symantec-Produkte maximieren und Wissen, Fachkenntnisse sowie globale Einblicke entwickeln können, wodurch Sie Ihre Geschäftsrisiken proaktiv verwalten können.

Zu den verfügbaren Unternehmensdiensten gehören:

Symantec Early Warning Solutions	Diese Lösungen bieten eine Frühwarnung vor Cyber-Angriffen, eine umfassende Bedrohungsanalyse und Gegenmaßnahmen zur Vorbeugung bzw. Vermeidung von Angriffen.
Managed Security Services (Verwaltete Sicherheitsdienste)	Diese Dienste erleichtern das Verwalten und Überwachen von Sicherheitsgeräten und -ereignissen, wodurch eine schnelle Reaktion auf reale Bedrohungen erfolgen kann.
Consulting Services (Beratungsdienste)	Die Symantec Consulting Services bieten auf der Website technisches Knowhow von Symantec und seinen vertrauenswürdigen Partnern. Die Symantec Consulting Services bieten eine Vielzahl von benutzerdefinierbaren Optionspaketen, die Bewertung, Entwurf, Implementierung, Überwachung und Verwaltungsfunktionen enthalten. Jedes Paket stellt den Aufbau und den Erhalt der Integrität und der Verfügbarkeit Ihrer IT-Ressourcen in den Mittelpunkt.
Educational Services (Schulungen)	Die Educational Services bieten eine Vielzahl technischer Schulungen, Sicherheitsschulungen, Sicherheitszertifizierungen und Hilfestellungen zur Problemerkennung.

Weitere Informationen zu den Unternehmensdiensten finden Sie auf der Website unter folgender URL:

www.symantec.com

Wählen Sie Ihr Land oder Ihre Sprache vom Site-Index aus.

Inhalt

Technischer Support	4
Kapitel 1	Verwenden der OLE-Automatisierung mit Symantec pcAnywhere
	9
	Informationen zur OLE-Automatisierung
	9
	Informationen zum pcAnywhere Automatisierungs-Server
	10
	Welche Aufgaben mit dem pcAnywhere Automatisierungs-Server ausgeführt werden können
	10
	Bevor Sie beginnen
	11
	Remote-Modul automatisch registrieren
	11
	Manuelles Registrieren des Remote-Moduls
	12
	Zugriff auf den pcAnywhere Automatisierungs-Server
	12
	Zugriff auf den pcAnywhere Automatisierungs-Server mit Visual Basic
	12
	Zugriff auf den pcAnywhere Automatisierungs-Server mit Visual C++
	14
	Host- und Remote-OLE-Objekte starten
	16
	Weiterführende Informationen
	16
Kapitel 2	Objektdefinitionen für Visual Basic
	17
	Informationen zu Visual Basic-Objekten
	17
	CRemoteDataManager-Methoden
	18
	CRemoteData-Eigenschaften
	21
	Eigenschaften von Verbindungstypen
	24
	Wählparameter
	25
	Eigenschaften des COM-Geräts
	26
	Eigenschaften für das NetBIOS-Gerät
	29
	Eigenschaften für ISDN über CAPI 2.0-Gerät
	30
	CRemoteDataEx-Objekt
	31
	Visual Basic-Beispielcode für Remote-Funktionen
	32
	CHostDataManager-Methoden
	34
	CHostData-Eigenschaften
	37
	Eigenschaften von Verbindungstypen
	39
	AssignConnection(ConnectionType)-Methode
	42
	UnassignConnection(ConnectionType)-Methode
	42

	Wählparameter	43
	Eigenschaften des COM-Geräts	44
	Eigenschaften für das NetBIOS-Gerät	47
	Eigenschaften für ISDN über CAPI 2.0-Gerät	48
	CHostDataEx-Objekt	48
	Visual Basic-Beispielcode für Hostfunktionen	56
	Awrem32-Funktionen	58
Kapitel 3	Objektdefinitionen für Visual C++	61
	Informationen zu Visual C++ Objekten	61
	CRemoteDataManager-Methoden	62
	CRemoteData-Objekt	66
	Get- und Set-Methoden	66
	Detailmethoden für Remote-Objekte	69
	Methoden für Remote-Objekte	71
	CRemoteDataEx-Objekt	73
	Visual C++ Beispielcode für Remote-Funktionen	73
	CHostDataManager-Methoden	74
	CHostData-Objekt	78
	Get- und Set-Methoden	79
	Detailmethoden für Hostobjekte	80
	Methoden für Hostobjekte	84
	CHostDataEx-Objekt	86
	Visual C++ Beispielcode für Hostfunktionen	88
	Awrem32-Funktionen	89
Index		93

Verwenden der OLE-Automatisierung mit Symantec pcAnywhere

In diesem Kapitel werden folgende Themen behandelt:

- [Informationen zur OLE-Automatisierung](#)
- [Informationen zum pcAnywhere Automatisierungs-Server](#)
- [Welche Aufgaben mit dem pcAnywhere Automatisierungs-Server ausgeführt werden können](#)
- [Bevor Sie beginnen](#)
- [Weiterführende Informationen](#)

Informationen zur OLE-Automatisierung

Mithilfe der OLE-Automatisierung können Sie externe Anwendungen oder sonstige Entwicklungs-Tools (z. B. Skripts oder Makros) erstellen, die Funktionen einer Anwendung steuern und automatisieren.

Die OLE-Automatisierung besteht aus folgenden Komponenten:

OLE-Automatisierung

Eine Anwendungs- oder Softwarekomponente, deren Funktionen durch andere Anwendungen oder Entwicklungs-Tools aufgerufen oder gesteuert werden können

Der pcAnywhere Automatisierungs-Server ist ein Beispiel für einen OLE-Automatisierungs-Server.

OLE-Automatisierungs-Controller Eine Anwendung oder ein Entwicklungs-Tool, die bzw. das die vom OLE-Automatisierungs-Server angebotenen Komponenten aufruft und steuert.

Sie können jede Programmiersprache verwenden, die die OLE-Automatisierung unterstützt. Microsoft Visual Basic und Microsoft Visual C++ sind die am häufigsten verwendeten Programmiersprachen.

Eine externe Anwendung stellt für den Zugriff auf einen OLE-Automatisierungs-Server eine Verbindung mit dem Server her und fordert anschließend Zugang zu einer oder mehreren öffentlichen Schnittstellen dieses Servers. Eine Schnittstelle ist ein Portal, das den Zugang zu einer oder mehreren verknüpften Methoden oder Eigenschaften gewährt. Nachdem eine Anwendung die Verbindung mit einem Server hergestellt hat, kann sie alle Schnittstellenmethoden so aufrufen, als ob sie Teil der externen Anwendung sind.

Informationen zum pcAnywhere Automatisierungs-Server

Mithilfe des pcAnywhere Automatisierungs-Servers können externe Anwendungen Remote- und Hostdateien von pcAnywhere verwalten, um Fernsteuerungs- und Dateiübertragungsaufgaben zu automatisieren. Der pcAnywhere Automatisierungs-Server dient als programmierbarer Ersatz für die Benutzeroberfläche von Symantec pcAnywhere und unterstützt einen Großteil des Standardverhaltens dieser Benutzeroberfläche.

Beispielsweise wird beim Erstellen eines Host-Objekts in pcAnywhere standardmäßig das erste verfügbare TAPI-Modem zugewiesen. Analog dazu ist das erste verfügbare TAPI-Modem bereits zugewiesen, wenn ein Host-Objekt mithilfe des pcAnywhere Automatisierungs-Servers erstellt wird und dann die Liste der zugeordneten Verbindungen durchläuft.

Welche Aufgaben mit dem pcAnywhere Automatisierungs-Server ausgeführt werden können

Mit dem pcAnywhere Automatisierungs-Server können Sie eine Vielzahl von Verwaltungs- und Produktivitätsaufgaben automatisieren.

Mit dem pcAnywhere Automatisierungs-Server können Sie:

- Software-Updates automatisch auf mehreren Computern im Netzwerk installieren

- Automatische Dateiübertragungen zwischen Computern für Überwachungs- oder Archivierungszwecke planen
- Automatisches Hinzufügen oder Entfernen eines Namens aus der Liste der zugelassenen Anrufer auf jedem pcAnywhere-Host in Ihrem Netzwerk durchführen

Dieses Handbuch enthält mehrere in Visual Basic und Visual C++ geschriebene Beispiele, die verdeutlichen, wie die Verbindung mit dem pcAnywhere Automatisierungs-Server hergestellt und wie dieser Server verwendet wird.

Siehe "[Informationen zu Visual Basic-Objekten](#)" auf Seite 17.

Siehe "[Informationen zu Visual C++ Objekten](#)" auf Seite 61.

Bevor Sie beginnen

Sofern eine Verbindung mit dem pcAnywhere Automatisierungs-Server und seinen Schnittstellen vorliegt, werden auch als GUIDs (Globally Unique Identifiers) bezeichnete ID-Parameter an die API-Funktionen der Automatisierungsbibliothek übergeben. Dem pcAnywhere Automatisierungs-Server und jeder Schnittstelle wird eine separate GUID zugewiesen. Diese GUIDs müssen in der Systemregistrierung vorhanden sein, um eine externe Anwendung mit dem pcAnywhere Automatisierungs-Server und dessen Schnittstellen zu verbinden.

Wenn Sie die externe Anwendung auf einem Computer mit Symantec pcAnywhere ausführen, können die GUID-Einträge automatisch registriert werden. Andernfalls müssen Sie die GUID-Einträge manuell registrieren.

Siehe "[Remote-Modul automatisch registrieren](#)" auf Seite 11.

Siehe "[Manuelles Registrieren des Remote-Moduls](#)" auf Seite 12.

Remote-Modul automatisch registrieren

Bevor mithilfe des OLE-Clients zum ersten Mal eine Verbindung mit einem anderen Computer hergestellt werden kann, muss das Remote-Modul registriert werden. Dies kann automatisch erfolgen, indem Sie mit Symantec pcAnywhere ein Remote-Objekt starten.

So registrieren Sie das Remote-Modul automatisch

- 1 Führen Sie einen der folgenden Schritte aus, um Symantec pcAnywhere zu öffnen:
 - Doppelklicken Sie auf dem Desktop auf das Symantec pcAnywhere-Programmsymbol.

- Klicken Sie in der Windows-Taskleiste auf "Start">"Programme">"Symantec pcAnywhere".
- 2 Klicken Sie in pcAnywhere-Manager auf "Remotes".
- 3 Doppelklicken Sie auf ein Remote-Verbindungsobjekt.

Dadurch wird das Remote-Modul registriert. Sie müssen die Verbindung nicht herstellen.

Manuelles Registrieren des Remote-Moduls

Wenn pcAnywhere nicht auf dem Computer installiert ist, auf dem die externe Anwendung ausgeführt wird, müssen Sie die GUIDs manuell registrieren, indem Sie die ausführbare Datei (Winawsvr.exe) für den pcAnywhere Automatisierungs-Server ausführen. Sie müssen die ausführbare Datei einmal ausführen, um die GUIDs zur Registrierung hinzuzufügen. Die Datei Winawsvr.exe befindet sich im Installationsverzeichnis.

Zugriff auf den pcAnywhere Automatisierungs-Server

Der Zugriff auf den pcAnywhere Automatisierungs-Server ist mit jeder Programmiersprache möglich, die die OLE-Automatisierung unterstützt. Visual Basic und Visual C++ sind die zwei bekanntesten Programmiersprachen, die die OLE-Automatisierung unterstützen.

Die Prinzipien der Codierung dieser beiden Plattformen ähneln sich, wobei in der Visual Basic-Umgebung zahlreiche Low-Level-Prozesse vom Visual Basic-Laufzeitsystem im Hintergrund ausgeführt werden.

Zugriff auf den pcAnywhere Automatisierungs-Server mit Visual Basic

Die Programmiersprache Visual Basic unterstützt die Interaktion mit OLE-Automatisierungs-Servern, wie beispielsweise dem pcAnywhere Automatisierungs-Server. Sie erstellen ein Standard-Exe-Projekt und geben dann für jede Methode Code ein, um auf den pcAnywhere Automatisierungs-Server zuzugreifen. Visual Basic erweitert die in den Quelldateien enthaltenen Aufrufe für die High-Level-Methode und erweitert sie intern in die entsprechenden Low-Level-Aufrufe für die OLE-Automatisierungsmethode.

So greifen Sie mit Visual Basic auf den pcAnywhere Automatisierungs-Server zu

- 1** Fügen Sie ein Objektvariablenpaar für jedes pcAnywhere-Objekt hinzu, auf das Sie zugreifen möchten.

Definieren Sie beispielsweise beim Arbeiten mit Remote-Objekten RemoteDataManager und RemoteDataObject als Objekt.

- 2** Stellen Sie mit Hilfe von RemoteDataManager eine Verbindung mit dem Daten-Manager des Remote-Objekts her.

Rufen Sie beispielsweise die Methode "CreateObject" mit WINAWSVR.REMOTEDATAMANAGER als Parameter auf.

Visual Basic ermittelt mit Hilfe des Textparameters die ID des Managers in der Registrierung und gibt die Schnittstelle an diesen Manager zurück.

- 3** Sobald es ein gültiges Daten-Manager-Objekt gibt, können Sie damit Folgendes ausführen:

- Das aktuelle Verzeichnis ermitteln
- In ein anderes Verzeichnis wechseln
- Die verknüpften Datenobjektdateien im aktuellen Verzeichnis aufzählen
- Eine Datenobjektdatei erstellen, laden oder löschen

- 4** Sobald ein Datenobjekt erstellt oder geladen ist, können Sie die Eigenschaften des Objekts abrufen bzw. festlegen.

Die Visual Basic-Syntax verwendet nicht den Namen einer Eigenschaft, um zwischen dem Abrufen und Festlegen des Werts dieser Eigenschaft zu unterscheiden. Stattdessen bestimmt die Position der Eigenschaft in Relation zum Zuweisungsoperator, ob es sich bei dem zugrunde liegenden Methodenaufruf um "Get" oder "Set" handelt.

Die folgenden Beispiele zeigen die Verwendung von "Get" und "Set":

- Um die Telefonnummer eines Objekts abzurufen, müssen Sie den Eigenschaftsnamen rechts vom Zuweisungsoperator platzieren.
Beispiel: `s = RemoteData.PhoneNumber()`, wobei "s" eine String-Variable ist.
- Sie müssen den Eigenschaftsnamen links vom Zuweisungsoperator platzieren, wenn Sie die Telefonnummer festlegen möchten.
Beispielsweise `RemoteData.PhoneNumber = "555-1212"`

Zugriff auf den pcAnywhere Automatisierungs-Server mit Visual C++

Der pcAnywhere Automatisierungs-Server verwendet Typlibibliotheken, um Automatisierungs-Clients, die in Visual C++ geschrieben sind, Informationen zu den Schnittstellen und Methoden bereitzustellen. Diese Typlibibliotheken verwenden Microsoft Foundation Classes (MFC), die mit dem Visual C++ Klassen-Assistenten in Ihre Anwendung importiert werden können.

Die Daten-Manager-Klassen der Typlibibliotheken stellen die Funktionalität zur Verfügung, die benötigt wird, um eine Schnittstelle zum pcAnywhere Automatisierungs-Server herzustellen und High-Level-Operationen für den zugehörigen Objekttyp der Schnittstelle auszuführen.

Führen Sie mithilfe des Daten-Manager-Objekts folgende Aufgaben aus:

- Das aktuelle Verzeichnis ermitteln oder ändern
- Die Liste der Datenobjektdateien im aktuellen Verzeichnis durchlaufen
- Ein benanntes Objekt erstellen, laden oder löschen

Ein erstelltes oder geladenes Objekt verwendet die verknüpfte Datenobjektklasse zur Überprüfung oder Änderung seiner verfügbaren Eigenschaften. Die meisten dieser Eigenschaften werden über ein Methodenpaar bereitgestellt, das mit dem Wort "Get" oder "Set" beginnt. Beispiel: Ein Benutzer ruft die Methode "GetPhoneNumber" auf, um die aktuelle Telefonnummerneigenschaft des Objekts zu überprüfen, und er ruft "SetPhoneNumber" auf, um die Telefonnummer anzugeben.

Klassen importieren und anzeigen

Der pcAnywhere Automatisierungs-Server verwendet die folgenden Typlibibliotheken:

Winawsvr.tlb	Liefert die Informationen, die zum Herstellen einer Verbindung mit dem pcAnywhere Automatisierungs-Server und für den Zugriff auf dessen Schnittstellen erforderlich sind.
Awrem32.tlb	Liefert die Informationen, die zum Steuern von pcAnywhere-Verbindungen erforderlich sind.

Die folgenden Anweisungen erläutern, wie Sie die Klassendefinitionen von den Typlibibliotheken des pcAnywhere Automatisierungs-Servers in Ihre MFC-Anwendung importieren und dann die Klassen anzeigen, die zur Anwendung hinzugefügt wurden.

So importieren Sie Klassen

- 1 Erstellen Sie in Visual C++ eine MFC-Anwendung.
- 2 Klicken Sie im Menü "Ansicht" auf "Klassen-Assistent".
- 3 Klicken Sie im Dialogfeld "Klassen-Assistent" auf "Klasse hinzufügen" und klicken Sie dann auf "Aus einer Typbibliothek".
- 4 Doppelklicken Sie auf "winawsvr.tlb".
- 5 Klicken Sie im Dialogfeld "Klassen bestätigen" auf "OK", um alle Klassendefinitionen zu importieren.
- 6 Klicken Sie im Dialogfeld "Klassen-Assistent" auf "Klasse hinzufügen" und klicken Sie dann auf "Aus einer Typbibliothek".
- 7 Doppelklicken Sie auf "awrem32.tlb".
- 8 Klicken Sie im Dialogfeld "Klassen bestätigen" auf "OK", um alle Klassendefinitionen zu importieren.
- 9 Klicken Sie im Dialogfeld "Klassen-Assistent" auf "OK", um den Importvorgang abzuschließen.

Die Klassen werden zur Anwendung hinzugefügt. Mithilfe dieser Klassen können Sie Objekte manipulieren und Verbindungen verwalten.

Beim Importieren der Klassendefinitionen aus den Typbibliotheken werden auch Unterstützungsdateien zur Anwendung hinzugefügt. Diese Dateien enthalten die Klassendefinitionen und den Implementierungsquellcode für den pcAnywhere Automatisierungs-Server.

Siehe ["Klassendefinitionen und Implementierungsdateien anzeigen"](#) auf Seite 15.

So zeigen Sie die hinzugefügten Klassen an

- 1 Öffnen Sie in Visual C++ die MFC-Anwendung.
- 2 Klicken Sie im Fenster "Arbeitsbereich" auf die Registerkarte "Klassen".

Klassendefinitionen und Implementierungsdateien anzeigen

Beim Importieren der Typbibliotheken des pcAnywhere Automatisierungs-Servers in Ihre Anwendung werden die folgenden Dateien hinzugefügt:

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

Diese Dateien enthalten die Klassendefinitionen und den Implementierungsquellcode für den pcAnywhere Automatisierungs-Server. Sie müssen diese Dateien nicht bearbeiten. Jede Anwendungsquelldatei, die Aufrufe für die Methoden des pcAnywhere Automatisierungs-Servers enthält, muss jedoch "Winawsvr.h" enthalten.

So zeigen Sie die Klassendefinitionen und Implementierungsdateien an

- 1 Öffnen Sie in Visual C++ die MFC-Anwendung.
- 2 Klicken Sie im Fenster "Arbeitsbereich" auf die Registerkarte "Dateien".

Host- und Remote-OLE-Objekte starten

Für Symantec pcAnywhere müssen alle Host- und Remote-Objekte im Standarddatenverzeichnis gespeichert werden. Vor dem Starten eines Host- oder Remote-Objekts, das Sie mit OLE erstellt haben, müssen Sie sicherstellen, dass das Objekt im Standarddatenverzeichnis von pcAnywhere gespeichert ist. Das Standardverzeichnis ist entweder der Ordner "All Users" in Windows XP oder der Ordner "ProgramData" in Vista.

Weiterführende Informationen

Weitere Informationen zur OLE-Automatisierung finden Sie in folgendem Referenzmaterial:

- 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.

Objektdefinitionen für Visual Basic

In diesem Kapitel werden folgende Themen behandelt:

- [Informationen zu Visual Basic-Objekten](#)
- [CRemoteDataManager-Methoden](#)
- [CRemoteData-Eigenschaften](#)
- [CRemoteDataEx-Objekt](#)
- [CHostDataManager-Methoden](#)
- [CHostData-Eigenschaften](#)
- [CHostDataEx-Objekt](#)
- [Awrem32-Funktionen](#)

Informationen zu Visual Basic-Objekten

Der pcAnywhere Automatisierungs-Server enthält die folgenden Komponenten zur Unterstützung der OLE-Automatisierung:

Winawsvr	Liefert die Informationen, die zum Herstellen einer Verbindung mit dem pcAnywhere Automatisierungs-Server und für den Zugriff auf dessen Schnittstellen erforderlich sind.
Awrem32	Liefert die Informationen, die zum Steuern von pcAnywhere-Verbindungen erforderlich sind.

In [Tabelle 2-1](#) werden die Objekte für Winawsvr beschrieben.

Tabelle 2-1 Winawsvr-Objekte

Objekt	Beschreibung	Referenz
CRemoteDataManager	Stellt die Methoden zum Erstellen, Öffnen, Bearbeiten, Speichern und Löschen von CRemoteData-Objekten zur Verfügung.	Siehe " CRemoteDataManager-Methoden " auf Seite 18.
CRemoteData	Definiert die Parameter zum Abrufen und Steuern der pcAnywhere-Remote-Funktionen.	Siehe " CRemoteData-Eigenschaften " auf Seite 21. Siehe " CRemoteDataEx-Objekt " auf Seite 31.
CHostDataManager	Stellt die Methoden zum Erstellen, Öffnen, Bearbeiten, Speichern und Löschen von CHostData-Objekten zur Verfügung.	Siehe " CHostDataManager-Methoden " auf Seite 34.
CHostData	Definiert die Parameter zum Abrufen und Steuern der pcAnywhere-Host-Funktionen.	Siehe " CHostData-Eigenschaften " auf Seite 37. Siehe " CHostDataEx-Objekt " auf Seite 48.

Awrem32 weist ein Objekt auf, das aus acht Schnittstellen zur Unterstützung von Fernsteuerungs- und Dateiübertragungssitzungen besteht.

Siehe "[Awrem32-Funktionen](#)" auf Seite 58.

Einige Funktionen werden nicht mehr unterstützt. Es werden jedoch Objekt-Definitionen für frühere Versionen zur Verfügung gestellt. Objektdefinitionen sind jedoch für die Verwendung mit früheren Versionen von pcAnywhere vorhanden.

Für Funktionen, die Kennwörter erfordern, können Kennwortwerte zwar festgelegt, aber nicht abgerufen werden. Diese Einschränkung erfolgt aus Sicherheitsgründen.

CRemoteDataManager-Methoden

Die CRemoteDataManager-Methoden liefern die Parameter und Rückgabewerte zum Abrufen und Steuern von CRemoteData-Objekten.

Tabelle 2-2 CRemoteDataManager-Methoden

CRemoteDataManager-Methode	Parameter oder Rückgabewert	Beschreibung
CurrentDirectory()	Rückgabewert = Zeichenkette	Gibt den vollständigen Pfadnamen des aktuellen Verzeichnisses zurück, in dem pcAnywhere-Remote-Objekte gespeichert sind
ChangeDirectory(NewDirectory)	Parameter = NewDirectory	Ändert das aktuelle Verzeichnis, in dem pcAnywhere-Remote-Objekte gespeichert sind
FindFirst(Pattern)	Parameter = Pattern als Zeichenkette	Findet die erste pcAnywhere-Remote-Objektdatei (*.chf) im aktuellen Verzeichnis. Das Suchmuster für Dateinamen filtert anhand des vorgegebenen Suchmusters für Dateinamen (ein Sternchen [*] findet alle Dateien im aktuellen Verzeichnis)
FindFirst(Name string)	Parameter = Name als Zeichenkette	Rückgabepuffer für den vollständigen Pfadnamen der Remote-Objektdatei (*.chf), die mit dem angegebenen Muster übereinstimmt
FindNext(Name)	Parameter = Name als Zeichenkette	Rückgabepuffer für den vollständigen Pfadnamen der Remote-Objektdatei, die mit dem Muster übereinstimmt, das im ursprünglichen Aufruf für FindFirst() angegeben wurde.
FindNext(Name)	Rückgabewert = Boolean	TRUE, wenn eine andere Remote-Objektdatei gefunden wird, die mit dem Muster übereinstimmt, das im Aufruf für FindFirst() angegeben wurde. Der vollständige Pfadname der übereinstimmenden Datei wird in "Name" gespeichert
RetrieveObject(Name)	Parameter = Name als Zeichenkette	Der vollständige Name der zu ladenden Remote-Objektdatei

CRemoteDataManager-Methode	Parameter oder Rückgabewert	Beschreibung
RetrieveObject(AccessMode)	Parameter = AccessMode als Ganzzahl	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
RetrieveObject>Password)	Parameter = Password als Zeichenkette	Objektkennwort. Darf NULL sein
RetrieveObjectEx(Name)	Parameter = Name als Zeichenkette	Der vollständige Name der zu ladenden Remote-Objektdatei
RetrieveObjectEx(AccessMode)	Parameter = AccessMode als Ganzzahl	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
RetrieveObjectEx>Password)	Parameter = Password als Zeichenkette	Objektkennwort. Darf NULL sein
RetrieveObjectEx	Rückgabewert = Objekt	CRemoteDataEx-Objekt aus der angegebenen Datei
CreateObject(Name)	Parameter = Name als Zeichenkette	Erstellt ein CRemoteData-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück. Der Parameter ist der vollständige Name der Remote-Objektdatei für das neue Objekt
CreateObject	Rückgabewert = Objekt	CRemoteData

CRemoteDataManager-Methode	Parameter oder Rückgabewert	Beschreibung
CreateObjectEx(Name)	Parameter = Name als Zeichenkette	Erstellt ein CRemoteDataEx-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück. Der Parameter ist der vollständige Name der Remote-Objektdatei für das neue Objekt
CreateObjectEx	Rückgabewert = Objekt	CRemoteDataEx
DeleteObject(Name)	Parameter = Name als Zeichenkette	Der vollständige Name der Remote-Objektdatei des zu löschenden Objekts.
DeleteObject>Password)	Parameter = Password als Zeichenkette	Objektkennwort.
DeleteObject	Rückgabewert = Boolean	TRUE, wenn das Objekt gelöscht wurde.

CRemoteData-Eigenschaften

Die Eigenschaften und Parameter, die für das CRemoteData-Objekt verfügbar sind, werden wie folgt angezeigt.

Ersetzen Sie in [Tabelle 2-3](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-3 CRemoteData-Eigenschaften und -Parameter

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.ComputerName(String)	String	Legt den Computernamen oder die IP-Adresse des Hostcomputers fest.
<CRemoteData>.PhoneNumber(String)	String	Legt die Telefonnummer des Hostcomputers fest.
<CRemoteData>.UseDialingProperties(Bool)	Bool	Legt die Wählparameter des Systems fest.
<CRemoteData>.RedialCount(Integer)	Integer	Legt fest, nach wie vielen Wählvorgängen der Anruf abgebrochen wird.
<CRemoteData>.RedialDelay(Integer)	Integer	Legt fest, wie viele Sekunden zwischen den Wahlwiederholungen gewartet werden soll.

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.AutoLoginName(String)	String	Legt den Namen des Benutzers für das automatische Login fest. Für weitere Informationen über das Verwenden von Domänen-Logins: Siehe "CRemoteDataEx-Objekt" auf Seite 31.
<CRemoteData>.AutoLoginPassword(String)	String	Legt das Kennwort für automatische Logins im Remote-Objekt fest. Aus Sicherheitsgründen bietet der pcAnywhere Automatisierungs-Server nicht die Möglichkeit, den Kennwortwert zu lesen. Es wird kein Kennwortwert zurückgegeben.
<CRemoteData>.Password(String)	String	Legt das Kennwort für das Remote-Objekt zur Verwendung mit den Parametern ExecuteProtection, ReadProtection und WriteProtection fest. Aus Sicherheitsgründen bietet der pcAnywhere Automatisierungs-Server nicht die Möglichkeit, den Kennwortwert zu lesen. Es wird kein Kennwortwert zurückgegeben.
<CRemoteData>.ExecuteProtection(Bool)	Bool	Legt fest, ob ein Kennwort zum Ausführen des Objekts erforderlich ist. Wird von Password festgelegt.
<CRemoteData>.ReadProtection(Bool)	Bool	Legt fest, ob ein Kennwort zum Anzeigen der Eigenschaften des Remote-Objekts erforderlich ist. Wird von Password festgelegt.
<CRemoteData>.WriteProtection(Bool)	Bool	Legt fest, ob ein Kennwort zum Speichern der Änderungen am Remote-Objekt erforderlich ist. Wird von Password festgelegt.
<CRemoteData>.LogSession(Bool)	Bool	Aktiviert und deaktiviert die Protokollierung der Sitzung.
<CRemoteData>.RecordFile(String)	String	Legt den vollständigen Pfad und Namen für den Speicherort der Datei fest, in der die aktive Sitzung aufgezeichnet wird.
<CRemoteData>.RecordSession(Bool)	Bool	Aktiviert und deaktiviert die automatische Aufzeichnung der Sitzung.

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.ReadObject(String)	String	Legt das Kennwort des Objekts fest. Verwenden Sie diese Eigenschaft, um die lokale Kopie des Remote-Objekts zu aktualisieren.
<CRemoteData>.WriteObject(String)	String	Legt das Kennwort des Objekts fest. Verwenden Sie diese Eigenschaft, um das Remote-Objekt zu erstellen oder Änderungen am Remote-Objekt zu speichern.

Die Eigenschaften und Rückgabewerte für CRemoteData werden in der folgenden Tabelle angezeigt.

Ersetzen Sie in [Tabelle 2-4](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-4 CRemoteData-Eigenschaften und -Rückgabewerte

Eigenschaft	Rückgabewert	Beschreibung
String = <CRemoteData>.ComputerName	String	Gibt den Computernamen oder die IP-Adresse des Host-Computers zurück.
String = <CRemoteData>.PhoneNumber	String	Gibt die Telefonnummer des Host-Computers zurück
Bool = <CRemoteData>.UseDialingProperties	Bool	Gibt die Wählparameter des Systems zurück, die im Remote-Objekt festgelegt sind.
Integer = <CRemoteData>.RedialCount	Integer	Gibt die Anzahl der Wahlwiederholungen zurück, die im Remote-Objekt festgelegt ist.
Integer = <CRemoteData>.RedialDelay	Integer	Gibt die Anzahl der Sekunden zwischen den Wahlwiederholungen zurück.
String = <CRemoteData>.AutoLoginName	String	Gibt den Loginnamen zurück, der für automatische Logins verwendet wird.
Bool = <CRemoteData>.ExecuteProtection	Bool	Gibt den Wert der ExecuteProtection-Einstellung zurück.
Bool = <CRemoteData>.ReadProtection	Bool	Gibt den Wert der ReadProtection-Einstellung zurück.
Bool = <CRemoteData>.WriteProtection	Bool	Gibt den Wert der WriteProtection-Einstellung zurück.
Bool = <CRemoteData>.LogSession	Bool	Gibt TRUE zurück, wenn die Protokollierung der Sitzung aktiviert ist.

Eigenschaft	Rückgabewert	Beschreibung
String = <CRemoteData>.RecordFile	String	Gibt den vollständigen Pfad und Namen der Datei zum Aufzeichnen der Sitzung zurück.
Bool = <CRemoteData>.RecordSession	Bool	Gibt den Wert der Einstellung zum Aufzeichnen der Sitzung zurück.

Eigenschaften von Verbindungstypen

In der folgenden Tabelle werden die Eigenschaften und Parameter der Verbindungstypen definiert.

Ersetzen Sie in [Tabelle 2-5](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-5 Eigenschaften und Parameter von Verbindungstypen

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.ConnectionType(String)	String	Legt den Verbindungstyp des Remote-Objekts fest. Der übergebene Wert muss ein gültiger Verbindungstyp sein, der durch die Funktionen FirstConnectionType() und NextConnectionType() definiert wird. Es folgt eine Aufstellung der gültigen Verbindungstypen: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN über CAPI 2.0 ■ Modemname (gemäß der Anzeige auf dem Computer)

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte der Verbindungstypen definiert.

Ersetzen Sie in [Tabelle 2-6](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-6 Eigenschaften und Rückgabewerte von Verbindungstypen

Eigenschaft	Rückgabewert	Beschreibung
String = <CRemoteData>.ConnectionTypes	String	Gibt den Verbindungstyp des Remote-Objekts zurück.
Integer = <CRemoteData>.ConnectionType	Integer	Gibt die Zahl der verfügbaren Verbindungstypen zurück. Es folgt eine Aufstellung der gültigen Verbindungstypen: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN über CAPI 2.0 ■ Modemname (gemäß der Anzeige auf dem Computer)
String = <CRemoteData>.FirstConnectionType	String	Gibt den ersten verfügbaren Verbindungstyp zurück.
String = <CRemoteData>.NextConnectionType	String	Gibt den nächsten verfügbaren Verbindungstyp zurück. Dies wird sequenziell für die Anzahl der Verbindungstypen abgerufen, die in <CRemoteData>.ConnectionType festgelegt ist, um alle Verbindungstypen aufzulisten.
Bool = <CRemoteData>.FindConnectionType(ConnectionType)	Bool	Gibt TRUE zurück, wenn der angegebene Verbindungstyp in der Liste der verfügbaren Verbindungstypen gefunden wird.

Wählparameter

In der folgenden Tabelle werden die Eigenschaften und Parameter zum Festlegen der Wählparameter für Modemverbindungen definiert.

Ersetzen Sie in [Tabelle 2-7](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-7 Eigenschaften und Parameter für die Wählparameter

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.AreaCode(String)	String	Legt die Ortskennzahl-Wählparameter für Modemverbindungen fest.
<CRemoteData>.CountryCode(String)	String	Legt die Landeskennzahl-Wählparameter für Modemverbindungen fest.

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte für Modemwählparameter definiert.

Ersetzen Sie in [Tabelle 2-8](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-8 Eigenschaften und Rückgabewerte für Wählparameter

Eigenschaft	Rückgabewert	Beschreibung
String = <CRemoteData>.AreaCode	String	Gibt die Ortskennzahl-Wählparameter zurück.
String = <CRemoteData>.CountryCode	String	Gibt die Landeskennzahl-Wählparameter zurück.
Integer = <CRemoteData>.CountryCodes	Integer	Gibt die Zahl der verfügbaren Landeskennzahlen zurück.
String = <CRemoteData>.FirstCountryCode	String	Gibt die erste verfügbare Landeskennzahl zurück, die im Betriebssystem angegeben ist.
String = <CRemoteData>.NextCountryCode	String	Gibt die nächste verfügbare Landeskennzahl zurück. Dies wird sequenziell für die Anzahl der Landeskennzahlen abgerufen, die in <CRemoteData.CountryCodes> festgelegt ist, um alle Landeskennzahlen aufzulisten.

Eigenschaften des COM-Geräts

In der folgenden Tabelle werden die Eigenschaften und Parameter beschrieben, mit denen Sie die Anschlusseinstellungen für Modem- und andere COM-basierte Verbindungen anpassen können.

Ersetzen Sie in [Tabelle 2-9](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-9 Eigenschaften und Parameter des COM-Geräts

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.ComParity(String)	String	Legt die Parität für die Kommunikation fest. Folgende Werte sind zulässig: ■ <Keine> ■ Odd ■ Even ■ Mark ■ Space
<CRemoteData>.ComFlowControl(String)	String	Legt die Flusssteuerung von COM-basierten Verbindungstypen fest. Folgende Werte sind zulässig: ■ <Keine> ■ XONXOFF ■ RTS/CTS ■ Both
<CRemoteData>.ComStartedBy(String)	String	Legt die Starteinstellung von COM-basierten Verbindungstypen fest. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI) ■ <CR> empfangen ■ Modemantwort
<CRemoteData>.ComEndedBy(String)	String	Legt die Endeinstellung von COM-basierten Verbindungstypen fest. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI)

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.ComSpeed(Long)	Long	Enthält die maximale Einstellung für die COM-Geschwindigkeit. Folgende Werte sind zulässig: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte des COM-Geräts beschrieben.

Ersetzen Sie in [Tabelle 2-10](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-10 Eigenschaften und Rückgabewerte des COM-Geräts

Eigenschaft	Rückgabewert	Beschreibung
String = <CRemoteData>.ComParity	String	Gibt einen der folgenden Werte für die Parität der Kommunikation zurück: ■ <Keine> ■ Odd ■ Even ■ Mark ■ Space
String = <CRemoteData>.ComFlowControl	String	Gibt die Flusssteuerungseinstellung des Remote-Objekts zurück. Folgende Werte sind zulässig: ■ <Keine> ■ XONXOFF ■ RTS/CTS ■ Both

Eigenschaft	Rückgabewert	Beschreibung
String = <CRemoteData>.ComStartedBy	String	Gibt die Startsteuerung von COM-basierten Verbindungstypen zurück. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI) ■ <CR> empfangen ■ Modemantwort
String = <CRemoteData>.ComEndedBy	String	Gibt die Endsteuerung von COM-basierten Verbindungstypen zurück. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI)
Long = <CRemoteData>.ComSpeed	Long	Gibt die aktuelle Einstellung für die Kommunikationsgeschwindigkeit des Remote-Objekts zurück. Folgende Werte sind zulässig: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Eigenschaften für das NetBIOS-Gerät

In der folgenden Tabelle werden die Eigenschaften und Parameter für ein NetBIOS-Netzwerkgerät definiert.

Ersetzen Sie in [Tabelle 2-11](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-11 NetBIOS-Eigenschaften und -Parameter

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.LanaNumber(Integer)	Integer	Legt die LANA-Nummer (LANA = LAN Adapter) für NetBIOS-Verbindungen fest.

In der folgenden Tabelle wird die Eigenschaft und der Rückgabewert für NetBIOS definiert.

Ersetzen Sie in [Tabelle 2-12](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-12 NetBIOS-Eigenschaft und -Rückgabewert

Eigenschaft	Rückgabewert	Beschreibung
Integer = <CRemoteData>.LanaNumber	Integer	Gibt die aktuelle Einstellung der LANA-Nummer (LANA = LAN Adapter) für NetBIOS-Verbindungen zurück.

Eigenschaften für ISDN über CAPI 2.0-Gerät

In der folgenden Tabelle werden die Eigenschaften und Parameter für europäische ISDN-Verbindungen definiert.

Ersetzen Sie in [Tabelle 2-13](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-13 Eigenschaften und Parameter für ISDN über CAPI 2.0

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.CapiChannelBonding(Bool)	Bool	Aktiviert oder deaktiviert die Kanalbündelung für ISDN-CAPI-Geräte.
<CRemoteData>.CapiExtensions(String)	String	Legt zusätzliche CAPI-Erweiterungen fest, die für die Kommunikation erforderlich sind.

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte für europäische ISDN-Verbindungen definiert.

Ersetzen Sie in [Tabelle 2-14](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-14 Eigenschaften und Rückgabewerte für ISDN über CAPI 2.0

Eigenschaft	Rückgabewert	Beschreibung
Bool= <CRemoteData>.CapiChannelBonding	Bool	Gibt die aktuelle Einstellung für die ISDN-CAPI-Kanalbündelung des Remote-Objekts zurück.
String = <CRemoteData>.CapiExtensions	String	Gibt die aktuelle Liste der CAPI-Erweiterungen des Remote-Objekts zurück.

CRemoteDataEx-Objekt

Das CRemoteDataEx-Objekt beinhaltet die gesamte Funktionalität des CRemoteData-Objekts sowie zusätzliche Funktionen.

Die folgende Tabelle beschreibt die Eigenschaften und Parameter.

Ersetzen Sie in [Tabelle 2-15](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-15 CRemoteDataEx-Parameter

Eigenschaft	Parameter	Beschreibung
<CRemoteData>.PrivateKey(String)	String	Legt den Namen des zu verwendenden Containers für private Schlüssel fest.
<CRemoteData>.CertificateName(String)	String	Legt den gemeinsamen Namen des zu verwendenden privaten Schlüssels fest.
<CRemoteData>.EncryptionLevel(Byte)	Byte	Legt die Verschlüsselungsstufe fest. Folgende Werte sind zulässig: ■ -1: Keine ■ 0: pcAnywhere ■ 1: Symmetrisch ■ 2: Öffentlicher Schlüssel
<CRemoteData>.DenyLowerEncrypt(Bool)	Bool	Definiert, ob der Remote-Computer eine Verbindung mit einem Hostcomputer zulässt, der eine niedrigere Verschlüsselungsstufe verwendet.
<CRemoteData>.AutoDomain(String)	String	Legt den Domännennamen für automatische Anmeldungen fest. Diese Option wird für die Windows NT- und Windows-Authentifizierung verwendet.

Die folgende Tabelle beschreibt die Eigenschaften und Rückgabewerte für das CRemoteDataEx-Objekt.

Ersetzen Sie in [Tabelle 2-16](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-16 CRemoteDataEx-Eigenschaften und -Rückgabewerte

Eigenschaft	Rückgabewert	Beschreibung
String = <CRemoteData>.PrivateKey	String	Gibt den Namen des derzeit aktiven Containers für private Schlüssel zurück.
String = <CRemoteData>.CertificateName	String	Gibt den gemeinsamen Namen des aktiven Containers für private Schlüssel zurück.
Byte = <CRemoteData>.EncryptionLevel	Byte	Gibt eine der folgenden Verschlüsselungseinstellungen zurück: ■ -1: Keine ■ 0: pcAnywhere ■ 1: Symmetrisch ■ 2: Öffentlicher Schlüssel
Bool = <CRemoteData>.DenyLowerEncrypt	Bool	Gibt den Wert der Einstellung "Kein Zugriff bei niedrigerer Verschlüsselungsstufe" zurück.
String = <CRemoteData>.AutoDomain	String	Gibt die Domänennameneinstellung für automatische Anmeldungen zurück. Diese Option wird für die Windows NT- und Windows-Authentifizierung verwendet.

Visual Basic-Beispielcode für Remote-Funktionen

Der folgende Visual Basic-Beispielcode lädt ein Remote-Datenobjekt und ändert dessen Eigenschaften:

```
Private Sub Command1_Click()  
    Dim RemoteDataManager as Object  
    Dim RemoteData as Object  
    Dim s as string  
  
    'Erstellen Sie das CRemoteDataManager-Objekt  
    Set RemoteDataManager = CreateObject(WINAWSVR.REMOTEDATAMANAGER)
```

```
'Zeigen Sie das aktuelle Verzeichnis an und ändern Sie es
s = RemoteDataManager.CurrentDirectory()
MsgBox ( s )
RemoteDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
s = RemoteDataManager.CurrentDirectory()
MsgBox ( s )

'Laden Sie das Hostdatenobjekt
Set RemoteData = RemoteDataManager.RetrieveObjectEx("pod.CHF",
2, 0)

'Zeigen Sie einige Eigenschaften an
s = RemoteData.AreaCode()
MsgBox (s)
s = RemoteData.PhoneNumber()
MsgBox (s)

'Legen Sie einige Eigenschaften fest
RemoteData.AreaCode = "212"
RemoteData.PhoneNumber = "555-5555"

'Schreiben Sie das Objekt auf den Datenträger
RemoteData.WriteObject (0)
End Sub
```

Mit den Methoden "FindFirst" und "FindNext" die Remote-Datei in einem Verzeichnis anzeigen:

```
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
```

Erstellen Sie ein Remote-Objekt. Legen Sie für den Verbindungstyp "TCP/IP" und für den Computernamen ""Host1"" fest. Rufen Sie dann das Objekt wie folgt auf:

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

Set RemoteDataManager =
CreateObject ("WINAWSVR.REMOTEDATAMANAGER")
MsgBox (RemoteDataManager.CurrentDirectory())

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

CHostDataManager-Methoden

Die CHostDataManager-Methoden liefern die Parameter und Rückgabewerte zum Abrufen und Steuern von CHostData-Objekten.

CHostDataManager-Methode	Parameter oder Rückgabewert	Beschreibung
CurrentDirectory()	Rückgabewert = Zeichenkette	Der vollständige Pfadname des aktuellen pcAnywhere-Datenverzeichnisses, in dem pcAnywhere-Hostobjekte gespeichert werden

CHostDataManager-Methode	Parameter oder Rückgabewert	Beschreibung
FindFirst(Pattern)	Parameter = Pattern als Zeichenkette	Suchmuster für Dateinamen, mit dem Objektdateien gefiltert werden (ein Sternchen [*] findet alle Dateien im aktuellen Verzeichnis).
FindFirst(Name string)	Parameter = Name als Zeichenkette	Rückgabepuffer für den vollständigen Pfadnamen der Hostobjektdatei, die mit dem angegebenen Muster übereinstimmt.
FindFirst	Rückgabewert = Boolean	TRUE, wenn eine Hostobjektdatei gefunden wird, die mit dem angegebenen Muster übereinstimmt. Der vollständige Pfadname der übereinstimmenden Datei wird in "Name" gespeichert
FindNext(Name)	Parameter = Name als Zeichenkette	Rückgabepuffer für den vollständigen Pfadnamen der Hostobjektdatei, die mit dem Muster übereinstimmt, das im ursprünglichen Aufruf für FindFirst() angegeben wurde.
FindNext	Rückgabewert = Boolean	TRUE, wenn eine andere Hostobjektdatei gefunden wird, die mit dem Muster übereinstimmt, das im Aufruf für FindFirst() angegeben wurde. Der vollständige Pfadname der übereinstimmenden Datei wird in "Name" gespeichert
RetrieveObject(Name)	Parameter = Name als Zeichenkette	Der vollständige Name der zu ladenden Hostobjektdatei
RetrieveObject(AccessMode)	Parameter = AccessMode als Ganzzahl	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz. Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
RetrieveObject>Password)	Parameter = Password als Zeichenkette	Objektkennwort. Darf NULL sein

CHostDataManager-Methode	Parameter oder Rückgabewert	Beschreibung
RetrieveObject	Rückgabewert = Objekt	CHostData-Objekt aus der angegebenen Datei.
RetrieveObjectEx(Name)	Parameter = Name als Zeichenkette	Der vollständige Name der zu ladenden Hostobjektdatei
RetrieveObjectEx(AccessMode)	Parameter = AccessMode als Ganzzahl	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
RetrieveObjectEx>Password)	Parameter = Password als Zeichenkette	Objektkennwort. Darf NULL sein
RetrieveObjectEx	Rückgabewert = Objekt	CHostDataEx-Objekt aus der angegebenen Datei
CreateObject(Name)	Parameter = Name als Zeichenkette	Erstellt ein CHostData-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück. Der Parameter ist der vollständige Name der Hostobjektdatei für das neue Objekt
CreateObject	Rückgabewert = Objekt	CHostData
CreateObjectEx(Name)	Parameter = Name als Zeichenkette	Erstellt ein CHostDataEx-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück. Der Parameter ist der vollständige Name der Hostobjektdatei für das neue Objekt
CreateObjectEx	Rückgabewert = Objekt	CHostDataEx
DeleteObject(Name)	Parameter = Name als Zeichenkette	Der vollständige Name der Hostobjektdatei des zu löschenden Objekts

CHostDataManager-Methode	Parameter oder Rückgabewert	Beschreibung
DeleteObject>Password)	Parameter = Password als Zeichenkette	Objektkennwort.
DeleteObject	Rückgabewert = Boolean	TRUE, wenn das Objekt gelöscht wurde.
Launch(Name)	Parameter = Name als Zeichenkette	Startet eine Hostobjektdatei, die das pcAnywhere-Host-Terminal-Fenster öffnet. Der Parameter ist die vollständige Hostobjektdatei des zu startenden Objekts
Launch	Rückgabewert = Boolean	TRUE, wenn das Objekt erfolgreich gestartet wird.

CHostData-Eigenschaften

Die folgende Tabelle beschreibt die Eigenschaften und Parameter, die für das CHostData-Objekt verfügbar sind.

Ersetzen Sie in [Tabelle 2-17](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-17 CHostData-Eigenschaften und -Parameter

Eigenschaft	Parameter	Beschreibung
<CHostData>.PhoneNumber(String)	String	Legt die Telefonnummer des Hostcomputers fest.
<CHostData>.UseDialingProperties(Bool)	Bool	Legt die Wählparameter des Systems fest.
<CHostData>.RedialCount(Integer)	Integer	Legt fest, nach wie vielen Wählvorgängen der Anruf abgebrochen wird.
<CHostData>.RedialDelay(Integer)	Integer	Legt fest, wie viele Sekunden zwischen den Wahlwiederholungen gewartet werden soll.

Eigenschaft	Parameter	Beschreibung
<CHostData>.Password(String)	String	Legt das Kennwort für das Hostobjekt zur Verwendung mit den Parametern ExecuteProtection, ReadProtection und WriteProtection fest. Aus Sicherheitsgründen bietet der pcAnywhere Automatisierungs-Server nicht die Möglichkeit, den Kennwortwert zu lesen. Es wird kein Kennwortwert zurückgegeben.
<CHostData>.ExecuteProtection(Bool)	Bool	Legt fest, ob ein Kennwort zum Ausführen des Objekts erforderlich ist. Wird von Password festgelegt.
<CHostData>.ReadProtection(Bool)	Bool	Legt fest, ob ein Kennwort zum Anzeigen der Eigenschaften des Hostobjekts erforderlich ist. Wird von Password festgelegt.
<CHostData>.WriteProtection(Bool)	Bool	Legt fest, ob ein Kennwort zum Speichern der Änderungen am Hostobjekt erforderlich ist. Wird von Password festgelegt.
<CHostData>.LogSession(Bool)	Bool	Aktiviert und deaktiviert die Protokollierung der Sitzung.
<CHostData>.RecordFile(String)	String	Legt den vollständigen Pfad und Namen für den Speicherort der Datei fest, in der die aktive Sitzung aufgezeichnet wird.
<CHostData>.RecordSession(Bool)	Bool	Aktiviert und deaktiviert die automatische Aufzeichnung der Sitzung.
<CHostData>.ReadObject(String)	String	Legt das Kennwort für das Objekt fest. Mit dieser Eigenschaft können Sie die lokale Kopie des Hostobjekts aktualisieren.
<CHostData>.WriteObject(String)	String	Legt das Kennwort für das Objekt fest. Verwenden Sie die Eigenschaft, um das Hostobjekt zu erstellen oder Änderungen am Hostobjekt zu speichern.

Die folgende Tabelle beschreibt die CHostData-Eigenschaften und -Rückgabewerte. Ersetzen Sie in [Tabelle 2-18](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-18 CHostData-Eigenschaften und -Rückgabewerte

Eigenschaft	Rückgabewert	Beschreibung
String = <CHostData>.PhoneNumber	String	Gibt die Telefonnummer des Hostcomputers zurück
Bool = <CHostData>.UseDialingProperties	Bool	Gibt die Verwendung der Wählparameter des Systems zurück, die im Hostobjekt festgelegt sind
Integer = <CHostData>.RedialCount	Integer	Gibt die Anzahl der Wahlwiederholungen zurück, die im Hostobjekt festgelegt ist
Integer = <CHostData>.RedialDelay	Integer	Gibt die Anzahl der Sekunden zwischen den Wahlwiederholungen zurück.
Bool = <CHostData>.ExecuteProtection	Bool	Gibt den Wert der ExecuteProtection-Einstellung zurück.
Bool = <CHostData>.ReadProtection	Bool	Gibt den Wert der ReadProtection-Einstellung zurück.
Bool = <CHostData>.WriteProtection	Bool	Gibt den Wert der WriteProtection-Einstellung zurück.
Bool = <CHostData>.LogSession	Bool	Gibt TRUE zurück, wenn die Protokollierung der Sitzung aktiviert ist.
String = <CHostData>.RecordFile	String	Gibt den vollständigen Pfad und Namen der Datei zum Aufzeichnen der Sitzung zurück.
Bool = <CHostData>.RecordSession	Bool	Gibt den Wert der Einstellung zum Aufzeichnen der Sitzung zurück.

Eigenschaften von Verbindungstypen

In der folgenden Tabelle werden die Eigenschaften und Parameter der Verbindungstypen beschrieben.

Ersetzen Sie in [Tabelle 2-19](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-19 Eigenschaften und Parameter von Verbindungstypen

Eigenschaft	Parameter	Beschreibung
<CHostData>.ConnectionType(String)	String	Legt den Verbindungstyp des Host-Objekts fest. Der übergebene Wert muss ein gültiger Verbindungstyp sein, der durch die Funktionen FirstConnectionType() und NextConnectionType() definiert wird. Es folgt eine Aufstellung der gültigen Verbindungstypen: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN über CAPI 2.0 ■ Modemname (gemäß der Anzeige auf dem Computer)
<CHostData>.AssignConnection(String)	String	Legt den aktiven Status für den Verbindungstyp fest, der im String definiert ist.
<CHostData>.UnassignConnection(String)	String	Legt den inaktiven Status für den Verbindungstyp fest, der im String definiert ist.

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte der Verbindungstypen beschrieben.

Ersetzen Sie in [Tabelle 2-20](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-20 Eigenschaften und Rückgabewerte von Verbindungstypen

Eigenschaft	Rückgabewert	Beschreibung
String = <CHostData>.ConnectionTypes	String	Gibt den Verbindungstyp des Host-Objekts zurück.

Eigenschaft	Rückgabewert	Beschreibung
Integer = <CHostData>.ConnectionType	Integer	Gibt die Zahl der verfügbaren Verbindungstypen zurück. Es folgt eine Aufstellung der gültigen Verbindungstypen: ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ ISDN über CAPI 2.0 ■ Modemname (gemäß der Anzeige auf dem Computer)
String = <CHostData>.FirstConnectionType	String	Gibt den ersten verfügbaren Verbindungstyp zurück.
String = <CHostData>.NextConnectionType	String	Gibt den nächsten verfügbaren Verbindungstyp zurück. Dies wird sequenziell für die Anzahl der Verbindungstypen abgerufen, die in <CHostData>.ConnectionTypes festgelegt ist, um alle Verbindungstypen aufzulisten.
Bool = <CHostData>.FindConnectionType(ConnectionType)	Bool	Gibt TRUE zurück, wenn der angegebene Verbindungstyp in der Liste der verfügbaren Verbindungstypen gefunden wird.
Integer = <CHostData>.MaxAssignedConnections	Integer	Gibt die maximal zulässige Anzahl an Verbindungstypen zurück, die auf diesem Host aktiv sein können.
String = <CHostData>.FirstAssignedConnection	String	Gibt den ersten zugeordneten aktiven Verbindungstyp zurück.
String = <CHostData>.NextAssignedConnection	String	Gibt den nächsten zugeordneten aktiven Verbindungstyp zurück.
Bool = <CHostData>.FindAssignedConnection(ConnectionType)	Bool	Gibt TRUE zurück, wenn der übergebene Verbindungstyp einem der aktiven Verbindungstypen entspricht.

AssignConnection(ConnectionType)-Methode

Die AssignConnection(Connection Type)-Methode fügt den angeforderten Verbindungstyp in die Liste mit den zugeordneten Verbindungstypen des Hostobjekts ein und macht diesen Verbindungstyp bei der Verarbeitung nachfolgender gerätespezifischer Methodenaufrufe zum aktuellen Verbindungstyp.

Falls der angeforderte Verbindungstyp bereits in der Liste der zugeordneten Verbindungen vorhanden ist, wird diese Liste nicht geändert. Nur der aktuelle Verbindungstyp wird an den angeordneten Verbindungstyp angepasst. Es ist nicht ungewöhnlich, die Methode "AssignConnection" während des Abrufens und Definierens verbindungsspezifischer Werte mehrmals auf demselben Objekt aufzurufen.

"AssignConnection" gibt TRUE zurück, wenn es den übergebenen Verbindungstyp auf dem Computer gibt und er entweder erfolgreich zugewiesen wird oder bereits zugewiesen ist. Die Methode gibt FALSE zurück, wenn es entweder den angeforderten Verbindungstyp auf diesem Computer nicht gibt oder der aktuelle Zähler für die zugeordnete Verbindung bereits seine maximal zulässige Höchstgrenze erreicht hat.

Ein pcAnywhere-Host-Objekt kann bis zu zwei zugeordnete Verbindungstypen unterstützen. Die Methode "AssignConnection" gibt FALSE zurück, wenn sie registriert, dass versucht wurde, diese Grenze zu überschreiten.

In der folgenden Tabelle wird der AssignConnection-Parameter definiert:

Parameter	Beschreibung
ConnectionType as string	Der Name des Gerätetyps, der zugeordnet werden soll.

In der folgenden Tabelle wird der AssignConnection-Rückgabewert definiert:

Rückgabewert	Beschreibung
Boolean	TRUE, wenn dieser Gerätetyp verfügbar ist und die maximal zulässige Anzahl zugeordneter Verbindungen noch nicht erreicht wurde.

UnassignConnection(ConnectionType)-Methode

Die UnassignConnection(Connection Type)-Methode gibt TRUE zurück, wenn der übergebene Verbindungstyp ordnungsgemäß aus der Liste mit den zugeordneten Verbindungstypen gelöscht wird.

In der folgenden Tabelle wird der UnassignConnection-Parameter definiert:

Parameter	Beschreibung
-----------	--------------

ConnectionType as string	Der Name des Gerätetyps, dessen Zuordnung aufgehoben werden soll.
--------------------------	---

In der folgenden Tabelle wird der UnassignConnection-Rückgabewert definiert:

Rückgabewert	Beschreibung
--------------	--------------

Boolean	TRUE, wenn die Zuordnung dieses Gerätetyps ordnungsgemäß aufgehoben wurde.
---------	--

Wählparameter

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte zum Festlegen der Wählparameter für Modemverbindungen definiert.

Ersetzen Sie in [Tabelle 2-21](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-21 Eigenschaften und Parameter für Modemwählparameter

Eigenschaft	Parameter	Beschreibung
<CHostData>.AreaCode(String)	String	Legt die Ortskennzahl für Modemverbindungen fest.
<CHostData>.CountryCode(String)	String	Legt die Landeskennzahl für Modemverbindungen fest.

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte für Modemwählparameter definiert.

Ersetzen Sie in [Tabelle 2-22](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-22 Eigenschaften und Rückgabewerte für Modemwählparameter

Eigenschaft	Rückgabewert	Beschreibung
String = <CHostData>.AreaCode	String	Gibt die Ortskennzahl zurück.
String = <CHostData>.CountryCode	String	Gibt die Landeskennzahl zurück.
Integer = <CHostData>.CountryCodes	Integer	Gibt die Zahl der verfügbaren Landeskennzahlen zurück.

Eigenschaft	Rückgabewert	Beschreibung
String= <CHostData>.FirstCountryCode	String	Gibt die erste verfügbare Landeskennzahl zurück, die im Betriebssystem angegeben ist.
String = <CHostData>.NextCountryCode	String	Gibt die nächste verfügbare Landeskennzahl zurück. Dies wird sequenziell für die Anzahl der Landeskennzahlen abgerufen, die in <CHostData>.CountryCodes festgelegt ist, um alle Landeskennzahlen aufzulisten.

Eigenschaften des COM-Geräts

In der folgenden Tabelle werden die Eigenschaften und Parameter definiert, mit denen Sie die Anschlusseinstellungen für Modem- und andere COM-basierte Verbindungen anpassen können.

Ersetzen Sie in [Tabelle 2-23](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-23 Eigenschaften und Parameter des COM-Geräts

Eigenschaft	Parameter	Beschreibung
<CHostData>.ComParity(String)	String	Legt die Parität für die Kommunikation fest. Folgende Werte sind zulässig: ■ <Keine> ■ Odd ■ Even ■ Mark ■ Space
<CHostData>.ComFlowControl(String)	String	Legt die Flusssteuerung von COM-basierten Verbindungstypen fest. Folgende Werte sind zulässig: ■ <Keine> ■ XONXOFF ■ RTS/CTS ■ Both

Eigenschaft	Parameter	Beschreibung
<CHostData>.ComStartedBy(String)	String	Legt die Starteinstellung von COM-basierten Verbindungstypen fest. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI) ■ <CR> empfangen ■ Modemantwort
<CHostData>.ComEndedBy(String)	String	Legt die Endeinstellung von COM-basierten Verbindungstypen fest. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI)
<CHostData>.ComSpeed(Long)	Long	Legt die maximale Einstellung für die COM-Geschwindigkeit fest. Folgende Werte sind zulässig: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte des COM-Geräts beschrieben.

Ersetzen Sie in [Tabelle 2-24](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-24 Eigenschaften und Rückgabewerte des COM-Geräts

Eigenschaft	Rückgabewert	Beschreibung
String = <CHostData>.ComParity	String	Gibt einen der folgenden Werte für die Parität der Kommunikation zurück: ■ <Keine> ■ Odd ■ Even ■ Mark ■ Space
String = <CHostData>.ComFlowControl	String	Gibt die Flusssteuerungseinstellung des Host-Objekts zurück. Folgende Werte sind zulässig: ■ <Keine> ■ XONXOFF ■ RTS/CTS ■ Both
String = <CHostData>.ComStartedBy	String	Gibt die Startsteuerung von COM-basierten Verbindungstypen zurück. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI) ■ <CR> empfangen ■ Modemantwort
String = <CHostData>.ComEndedBy	String	Gibt die Endsteuerung von COM-basierten Verbindungstypen zurück. Folgende Werte sind zulässig: ■ Immer verbunden ■ Empfangssignalpegel (DCD) ■ Sendebereitschaft (CTS) ■ Betriebsbereitschaft (DSR) ■ Ankommender Anruf (RI)

Eigenschaft	Rückgabewert	Beschreibung
Long = <CHostData>.ComSpeed	Long	Gibt die aktuelle Einstellung für die Kommunikationsgeschwindigkeit des Host-Objekts zurück. Folgende Werte sind zulässig: ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Eigenschaften für das NetBIOS-Gerät

In der folgenden Tabelle werden die Eigenschaften und Parameter für NetBIOS-Netzwerkgeräte definiert.

Ersetzen Sie in [Tabelle 2-25](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-25 NetBIOS-Eigenschaften und -Parameter

Eigenschaft	Parameter	Beschreibung
<CHostData>.LanaNumber(Integer)	Integer	Legt die LANA-Nummer (LANA = LAN Adapter) für NetBIOS-Verbindungen fest.

In der folgenden Tabelle werden die Eigenschaften und Rückgabewerte für NetBIOS-Netzwerkgeräte definiert.

Ersetzen Sie in [Tabelle 2-26](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-26 NetBIOS-Rückgabewerte

Eigenschaft	Rückgabewert	Beschreibung
Integer = <CHostData>.LanaNumber	Integer	Gibt die aktuelle Einstellung der LANA-Nummer (LANA = LAN Adapter) für NetBIOS-Verbindungen zurück.

Eigenschaften für ISDN über CAPI 2.0-Gerät

In der folgenden Tabelle werden die Eigenschaften und Parameter für europäische ISDN-Verbindungen definiert.

Ersetzen Sie in [Tabelle 2-27](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-27 Eigenschaften und Parameter für ISDN über CAPI 2.0

Eigenschaft	Parameter	Beschreibung
<CHostData>.CapiChannelBonding(Bool)	Bool	Aktiviert oder deaktiviert die Kanalbündelung für ISDN-CAPI-Geräte.
<CHostData>.CapiExtensions(String)	String	Legt zusätzliche CAPI-Erweiterungen fest, die für die Kommunikation erforderlich sind.

In [Tabelle 2-28](#) werden die Eigenschaften und Rückgabewerte für europäische ISDN-Verbindungen definiert.

Tabelle 2-28 Eigenschaften und Rückgabewerte für ISDN über CAPI 2.0

Eigenschaft	Rückgabewert	Beschreibung
Bool= <CHostData>.CapiChannelBonding	Bool	Gibt die aktuelle Einstellung für die ISDN-CAPI-Kanalbündelung des Host-Objekts zurück.
String = <CHostData>.CapiExtensions	String	Gibt die aktuelle Liste der CAPI-Erweiterungen des Host-Objekts zurück.

CHostDataEx-Objekt

Das CHostDataEx-Objekt beinhaltet die gesamte Funktionalität des CHostData-Objekts sowie zusätzliche Funktionen.

Die folgende Tabelle beschreibt die Eigenschaften und Parameter.

Ersetzen Sie in [Tabelle 2-29](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-29 CHostDataEx-Eigenschaften und -Parameter

Eigenschaft	Parameter	Beschreibung
<CHostData>.CryptPrivateKey(String)	String	Legt den Namen des zu verwendenden Containers für private Schlüssel fest.

Eigenschaft	Parameter	Beschreibung
<CHostData>.CryptCommonName(String)	String	Legt den gemeinsamen Namen des zu verwendenden privaten Schlüssels fest.
<CHostData>.CryptReqLevel(Byte)	Byte	Legt die Verschlüsselungsstufe fest. Folgende Werte sind zulässig: ■ -1: Keine ■ 0: pcAnywhere ■ 1: Symmetrisch ■ 2: Öffentlicher Schlüssel
<CHostData>.CryptRefuseLower(Byte)	Bool	Definiert, ob der Hostcomputer Verbindungen von einem Remote-Computer zulässt, der eine niedrigere Verschlüsselungsstufe verwendet.
<CHostData>.CallersPath(String)	String	Legt den vollständigen Pfad zu den Anruferdateien fest.
<CHostData>.ConfirmConnect(Byte)	Bool	Definiert, ob der Hostbenutzer zur Bestätigung von Verbindungen aufgefordert wird.
<CHostData>.ConfirmTimeout(Byte)	Byte	Legt fest, nach wie vielen Sekunden die Aufforderung zur Bestätigung des Verbindungszeit-Grenzwertes abläuft.
<CHostData>.ConfirmDeny(Byte)	Bool	Definiert, ob die Verbindung beendet werden soll, wenn die Aufforderung zur Bestätigung des Verbindungszeit-Grenzwertes abläuft.
<CHostData>.PwCaseSensitive(Byte)	Bool	Erzwingt die Verwendung von Kennwörtern, bei denen die Groß- und Kleinschreibung beachtet wird.
<CHostData>.PwAttempts(Byte)	Byte	Legt fest, nach wie vielen aufeinander folgenden fehlgeschlagenen Anmeldeversuchen die Verbindung getrennt wird.
<CHostData>.PwTimeout(Byte)	Byte	Legt die Anzahl der Minuten fest, die dem Benutzer für die Anmeldung zur Verfügung stehen, bevor die Verbindung getrennt wird.

Eigenschaft	Parameter	Beschreibung
<CHostData>.ActiveKbds(Byte)	Byte	Definiert die Maus und die Tastatur, die während der Verbindung aktiv sind. Folgende Werte sind zulässig: ■ 0: Host und Remote ■ 1: Host ■ 2: Remote
<CHostData>.InactiveTimeout(Byte)	Byte	Legt fest, nach wie vielen Minuten der Inaktivität der Tastatur und der Maus die Verbindung getrennt wird.
<CHostData>.LockSystemWhileWait(Bool)	Bool	Legt die Einstellung zum Sperren des Hostcomputers beim Start fest.
<CHostData>.MinimizeOnLaunch(Bool)	Bool	Legt die Option für die minimierte Ausführung beim Start des Hosts fest.
<CHostData>.RunAsService(Bool)	Bool	Ermöglicht die Ausführung des Hosts als Dienst.
<CHostData>.ConnLostWait(Byte)	Byte	Legt fest, nach wie vielen Minuten ein anderer Benutzer eine Verbindung herstellen kann.
<CHostData>.ConnLostHostOpts(Bool)	Bool	Definiert, ob auf eine andere Verbindung gewartet werden soll oder ob der Host beendet werden soll, falls die Sitzung nicht ordnungsgemäß beendet wird. Folgende Werte sind zulässig: ■ FALSE: Warten ■ TRUE: Host beenden
<CHostData>.EnableConnLostSecurity(Bool)	Bool	Aktiviert oder deaktiviert die Sitzungsende-Sicherheitsoptionen für Sitzungen, die nicht ordnungsgemäß beendet werden.

Eigenschaft	Parameter	Beschreibung
<CHostData>.AuthenticationType(Byte)	Byte	Legt den Authentifizierungstyp fest. Folgende Werte sind zulässig: ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS Active Directory-Dienste ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID Standardmäßig wird die pcAnywhere-Authentifizierung verwendet, falls kein Authentifizierungstyp festgelegt wird oder falls die festgelegte Option unzulässig ist (wenn beispielsweise der Authentifizierungstyp nicht verfügbar ist).
<CHostData>.ConnLostSecurity(Byte)	Byte	Legt die Sicherheitsoptionen für die Verwaltung einer nicht ordnungsgemäß beendeten Sitzung fest. Folgende Werte sind zulässig: ■ 1: Benutzer abmelden ■ 2: Hostcomputer neu starten ■ 3: Computer sperren
<CHostData>.CallbkDelay(Byte)	Byte	Legt fest, nach wie vielen Sekunden das Hostmodem den Remote-Host wieder anwählt.
<CHostData>.EndSessHostOpts(Bool)	Bool	Definiert, ob der Host nach einem normalen Sitzungsende auf eine andere Verbindung wartet oder ob er beendet wird. Folgende Werte sind zulässig: ■ FALSE: Auf nächste Verbindung warten ■ TRUE: Host beenden

Eigenschaft	Parameter	Beschreibung
<CHostData>.EnableEndSessSecurity(Bool)	Bool	Aktiviert oder deaktiviert die Sicherheitsoptionen für Sitzungen, die ordnungsgemäß beendet werden.
<CHostData>.EndSessSecurity(Byte)	Byte	Legt die Sicherheitsoptionen für eine ordnungsgemäß beendete Sitzung fest. Folgende Werte sind zulässig: ■ 1: Benutzer abmelden ■ 2: Hostcomputer neu starten ■ 3: Computer sperren
<CHostData>.BlankHost(Bool)	Bool	Legt die Funktion zum Löschen des Bildschirms auf dem Hostcomputer fest. Einige Grafikkarten unterstützen diese Option nicht.
<CHostData>.AllowRemoteMouse(Bool)	Bool	Deaktiviert die Verwendung der Maus auf dem Remote-Computer während einer Sitzung.
<CHostData>.RebootOnDisconnect(Bool)	Bool	Erzwingt einen Neustart des Computers, wenn das Sitzungsende den Wert TRUE aufweist.
<CHostData>.PasswordAfterDisc(Bool)	Bool	Meldet den Benutzer nach dem Ende der Sitzung ab.
<CHostData>.LogFailures(Bool)	Bool	Protokolliert fehlgeschlagene Kennworteingabeversuche.
<CHostData>.AllowDriveSecurity(Bool)	Bool	Aktiviert Optionen für die Laufwerksicherheit. Diese Einstellung ist nur für NTFS-Dateisysteme zulässig.
<CHostData>.UseDirectoryServices(Bool)	Bool	Aktiviert die Verwendung von Verzeichnisdiensten für die Authentifizierung.
<CHostData>.DirectoryServiceEntry(String)	String	Legt die Einstellungen für die Verzeichnisdienste fest.

Die folgende Tabelle beschreibt die Eigenschaften und Rückgabewerte für das CHostDataEx-Objekt.

Ersetzen Sie in [Tabelle 2-30](#) die Informationen in spitzen Klammern durch die tatsächlichen Werte.

Tabelle 2-30 CHostDataEx-Eigenschaften und -Rückgabewerte

Eigenschaft	Rückgabewert	Beschreibung
String = <CHostDataEx>.CryptPrivateKey	String	Gibt den derzeit aktiven Container für private Schlüssel zurück.
String = <CHostDataEx>.CryptCommonName	String	Gibt den gemeinsamen Namen des aktiven Containers für private Schlüssel zurück.
Byte = <CHostDataEx>.CryptReqLevel	Byte	Gibt eine der folgenden Verschlüsselungseinstellungen zurück: ■ -1: Keine ■ 0: pcAnywhere ■ 1: Symmetrisch ■ 2: Öffentlicher Schlüssel
Bool = <CHostDataEx>.CryptRefuseLower	Bool	Gibt den Wert der Einstellung "Kein Zugriff bei niedrigerer Verschlüsselungsstufe" zurück.
String = <CHostDataEx>.CallersPath	String	Gibt den derzeit aktiven Pfad zu den Anruferdateien zurück.
Bool = <CHostDataEx>.ConfirmConnect	Bool	Gibt den Wert der Einstellung zum Bestätigen der Verbindung zurück.
Byte = <CHostDataEx>.ConfirmTimeout	Byte	Gibt den Wert der Einstellung "Kein Zugriff bei niedrigerer Verschlüsselungsstufe" zurück.
Bool = <CHostDataEx>.ConfirmDeny	Bool	Gibt den Wert der Einstellung "Verbindung nach Wartezeit abbrechen" zurück.
Bool = <CHostDataEx>.PwCaseSensitive	Bool	Gibt den Wert der Einstellung "Groß-/Kleinschreibung bei Kennwort" zurück.
Byte = <CHostDataEx>.PwAttempts	Byte	Gibt den Wert der Einstellung "Anmeldeversuche pro Anruf begrenzen" zurück.
Byte = <CHostDataEx>.PwTimeout	Byte	Gibt den Wert der Einstellung "Zeit für Anmeldung begrenzen" (in Minuten) zurück.
Byte = <CHostDataEx>.ActiveKbds	Byte	Gibt die Einstellungen für die aktive Tastatur und die aktive Maus zurück. Folgende Werte sind zulässig: ■ 0: Host und Remote ■ 1: Host ■ 2: Remote

Eigenschaft	Rückgabewert	Beschreibung
Byte = <CHostDataEx>.InactiveTimeout	Byte	Gibt zurück, nach wie vielen Minuten die Verbindung getrennt wird, falls der Grenzwert für die Dauer der Inaktivität überschritten wird.
Bool = <CHostDataEx>.LockSystemWhileWait	Bool	Gibt die Einstellung zum Sperren des Hostcomputers beim Start zurück.
Bool = <CHostDataEx>.MinimizeOnLaunch	Bool	Gibt die Option für die minimierte Ausführung beim Start des Hosts zurück.
Bool = <CHostDataEx>.RunAsService	Bool	Gibt die Einstellung für die Ausführung des Hosts als Dienst zurück.
Byte = <CHostDataEx>.ConnLostWait	Byte	Gibt zurück, nach wie vielen Minuten ein anderer Benutzer eine Verbindung herstellen kann.
Bool = <CHostDataEx>.ConnLostHostOpts	Bool	Gibt zurück, ob auf eine andere Verbindung gewartet werden soll oder ob der Host beendet werden soll, falls die Sitzung nicht ordnungsgemäß beendet wird. Folgende Werte sind zulässig: ■ FALSE: Warten ■ TRUE: Host beenden
Bool = <CHostDataEx>.ConnLostWait	Bool	Gibt den Wert der Sicherheitsoption zurück, die für die Verwaltung einer nicht ordnungsgemäß beendeten Sitzung festgelegt ist.

Eigenschaft	Rückgabewert	Beschreibung
Byte = <CHostDataEx>.AuthenticationType	Byte	Gibt die Referenznummer des Authentifizierungstyps zurück. Folgende Werte sind zulässig: ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS Active Directory-Dienste ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID Standardmäßig wird die pcAnywhere-Authentifizierung verwendet, falls kein Authentifizierungstyp festgelegt wird oder falls die festgelegte Option unzulässig ist (wenn beispielsweise der Authentifizierungstyp nicht verfügbar ist).
Byte = <CHostDataEx>.ConnLostSecurity	Byte	Gibt die numerische Darstellung der Sicherheitsstufe zurück, die für die Verwaltung einer nicht ordnungsgemäß beendeten Sitzung festgelegt ist. Folgende Werte sind zulässig: ■ 1: Benutzer abmelden ■ 2: Hostcomputer neu starten ■ 3: Computer sperren
Byte = <CHostDataEx>.CallbkDelay	Byte	Gibt zurück, nach wie vielen Sekunden das Hostmodem den Remote-Host wieder anwählt.
Bool = <CHostDataEx>.EndSessHostOpts	Bool	Gibt zurück, ob der Host nach einem normalen Sitzungsende auf eine andere Verbindung wartet oder ob er beendet wird. Folgende Werte sind zulässig: ■ FALSE: Auf nächste Verbindung warten ■ TRUE: Host beenden

Eigenschaft	Rückgabewert	Beschreibung
Bool = <CHostDataEx>.EnableEndSessSecurity	Bool	Gibt zurück, ob die Sicherheitsoptionen für das Sitzungsende für eine ordnungsgemäß beendete Sitzung aktiviert werden.
Byte = <CHostDataEx>.EndSessSecurity	Byte	Gibt die Sicherheitsoption zurück, die für eine ordnungsgemäß beendete Sitzung festgelegt ist. Folgende Werte sind zulässig: ■ 1: Benutzer abmelden ■ 2: Hostcomputer neu starten ■ 3: Computer sperren
Bool = <CHostDataEx>.BlankHost	Bool	Gibt die festgelegte Funktion zum Löschen des Bildschirms zurück.
Bool = <CHostDataEx>.AllowRemoteMouse	Bool	Gibt zurück, ob der Remote-Benutzer während einer Sitzung die Kontrolle über Tastatur und Maus hat.
Bool = <CHostDataEx>.RebootOnDisconnect	Bool	Gibt zurück, ob der Hostcomputer nach dem Ende der Sitzung neu gestartet wird.
Bool = <CHostDataEx>.PasswordAfterDisc	Bool	Gibt zurück, ob der Hostbenutzer nach dem Ende der Sitzung abgemeldet wird.
Bool = <CHostDataEx>.LogFailures	Bool	Gibt zurück, ob die Option zum Protokollieren fehlgeschlagener Kennworteingabeveruche festgelegt ist.
Bool = <CHostDataEx>.AllowDriveSecurity	Bool	Gibt zurück, ob die Option für die Laufwerksicherheit aktiviert ist.
Bool = <CHostDataEx>.UseDirectoryServices	Bool	Gibt zurück, ob die Verzeichnisdiensteoption aktiviert ist.
String = <CHostDataEx>.DirectoryServiceEntry	String	Gibt die Einstellungen für die Verzeichnisdienste zurück.

Visual Basic-Beispielcode für Hostfunktionen

Der folgende Visual Basic-Beispielcode lädt ein Host-Datenobjekt und ändert dessen Eigenschaften:

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

```
Dim HostData as Object
Dim s as string

'Erstellen Sie das CHostDataManager-Objekt
Set HostDataManager = CreateObject(WINAVSVR.BEHOSTDATAMANAGER)

'Zeigen Sie das aktuelle Verzeichnis an und ändern Sie es
s = HostDataManager.CurrentDirectory()
MsgBox ( s )
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
s = HostDataManager.CurrentDirectory()
MsgBox ( s )

'Laden Sie das Hostdatenobjekt
Set HostData = HostDataManager.RetrieveObject("pod.BHF", 2, 0)

'Zeigen Sie einige Eigenschaften an
s = HostData.AreaCode()
MsgBox (s)
s = HostData.PhoneNumber()
MsgBox (s)
'Legen Sie einige Eigenschaften fest
RemoteData.HostData = "212"
RemoteData.HostData = "555-5555"

'Schreiben Sie das Objekt auf den Datenträger
HostData.WriteObject (0)
End Sub
```

Mit den Methoden "FindFirst" und "FindNext" die Hostdatei in einem Verzeichnis anzeigen:

```
Private Sub Command5_Click()

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

```

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

End Sub

```

Awrem32-Funktionen

Die Awrem32-Funktionen liefern die Parameter und Rückgabewerte zum Verwalten von Verbindungen zwischen Host- und Remote-Computer.

Awrem32-Funktion	Parameter oder Rückgabewert	Beschreibung
awConnect(FileName)	Parameter = Name als Zeichenkette	Der vollständige Name der .chf-Datei, die Informationen über den Hostcomputer enthält.
awConnect	Rückgabewert = Boolean	TRUE nach Ausführen des Befehls
awDisconnect()	Rückgabewert = Boolean	Nach Aufruf dieser Funktion muss das aufrufende Programm das Objekt löschen (C++ - delete IAwrem32X*, VB - set ObjectName = Nothing;).
FileXferFromHost(HostFile)	Parameter = HostFile als Zeichenkette	Enthält den vollständigen Pfad und Namen der Datei, die vom Hostcomputer kopiert werden soll
FileXferFromHost(RemoteFile)	Parameter = RemoteFile als Zeichenkette	Enthält den vollständigen Zielpfad und Dateinamen. Die HostFile- und RemoteFile-Strings müssen nicht identisch sein
FileXferFromHost	Rückgabewert = Boolean	TRUE nach Ausführen des Befehls

Awrem32-Funktion	Parameter oder Rückgabewert	Beschreibung
FileXferToHost(HostFile)	Parameter = HostFile als Zeichenkette	Kopiert eine Datei vom Remote-Computer auf den Hostcomputer. Die Parameter können Platzhalter enthalten Der Parameter enthält den vollständigen Zielpfad und Dateinamen
FileXferToHost(RemoteFile)	Parameter = RemoteFile als Zeichenkette	Enthält den vollständigen Pfad und Namen der Datei, die vom Remote-Computer kopiert werden soll. Die HostFile- und RemoteFile-Strings müssen nicht identisch sein
CreateFolderOnHost(FolderName)	Parameter = FolderName als Zeichenkette	Enthält das Laufwerk und den Pfad zum Erstellen des Ordners auf dem Hostcomputer
CreateFolderOnHost	Rückgabewert = Boolean	TRUE nach Ausführen des Befehls
ExecuteHostFile(FileName)	Parameter = FileName als Zeichenkette	Führt eine vorhandene Datei auf dem Hostcomputer aus. Diese Funktion führt nur Stapel-, Befehls- und ausführbare Dateien aus. Es werden keine Dateien ausgeführt, die mit ausführbaren Dateien verknüpft sind. Wenn Sie beispielsweise eine .doc-Datei ausführen, wird mit dieser Funktion Microsoft Word nicht geöffnet. Der Parameter enthält den vollständigen Pfad zu der auf dem Hostcomputer gespeicherten Datei
ExecuteHostFile	Rückgabewert = Boolean	TRUE nach Ausführen des Befehls
GetError()	Rückgabewert = Zeichenkette	Gibt den letzten Fehler zurück, der in Awrem32 aufgetreten ist.
ConnectionStatus()	Rückgabewert = Short	Gibt den aktuellen Status der Verbindung mit dem Hostcomputer zurück Die folgenden Werte sind verfügbar: ■ -1 = Verbindung unterbrochen ■ 0 = Keine Verbindung ■ 1 = Verbundene Sitzung

Objektdefinitionen für Visual C++

In diesem Kapitel werden folgende Themen behandelt:

- [Informationen zu Visual C++ Objekten](#)
- [CRemoteDataManager-Methoden](#)
- [CRemoteData-Objekt](#)
- [CRemoteDataEx-Objekt](#)
- [CHostDataManager-Methoden](#)
- [CHostData-Objekt](#)
- [CHostDataEx-Objekt](#)
- [Awrem32-Funktionen](#)

Informationen zu Visual C++ Objekten

Der pcAnywhere Automatisierungs-Server enthält die folgenden Komponenten zur Unterstützung der OLE-Automatisierung:

Winawsvr	Liefert die Informationen, die zum Herstellen einer Verbindung mit dem pcAnywhere Automatisierungs-Server und für den Zugriff auf dessen Schnittstellen erforderlich sind.
Awrem32	Liefert die Informationen, die zum Steuern von pcAnywhere-Verbindungen erforderlich sind.

Winawsvr besteht aus den in [Tabelle 3-1](#) beschriebenen Objekten.

Tabelle 3-1 Winawsvr-Objekte

Objekt	Beschreibung	Referenz
CRemoteDataManager	Stellt die Methoden zum Erstellen, Öffnen, Bearbeiten, Speichern und Löschen von CRemoteData-Objekten zur Verfügung.	Siehe "CRemoteDataManager-Methoden" auf Seite 62.
CRemoteData	Definiert die Parameter zum Abrufen und Steuern der pcAnywhere-Remote-Funktionen.	Siehe "CRemoteData-Objekt" auf Seite 66. Siehe "CRemoteDataEx-Objekt" auf Seite 73.
CHostDataManager	Stellt die Methoden zum Erstellen, Öffnen, Bearbeiten, Speichern und Löschen von CHostData-Objekten zur Verfügung.	Siehe "CHostDataManager-Methoden" auf Seite 74.
CHostData	Definiert die Parameter zum Abrufen und Steuern der pcAnywhere-Host-Funktionen.	Siehe "CRemoteData-Objekt" auf Seite 66. Siehe "CRemoteDataEx-Objekt" auf Seite 73.

Awrem32 weist ein Objekt auf, das aus acht Schnittstellen zur Unterstützung von Fernsteuerungs- und Dateiübertragungssitzungen besteht.

Siehe "Awrem32-Funktionen" auf Seite 89.

Einige Funktionen werden nicht mehr unterstützt. Es werden jedoch Objekt-Definitionen für frühere Versionen zur Verfügung gestellt. Objektdefinitionen sind jedoch für die Verwendung mit früheren Versionen vorhanden.

Für Funktionen, die Kennwörter erfordern, können Kennwortwerte zwar festgelegt, aber nicht abgerufen werden. Diese Einschränkung erfolgt aus Sicherheitsgründen.

CRemoteDataManager-Methoden

Die CRemoteDataManager-Methoden liefern die Parameter und Rückgabewerte zum Abrufen und Steuern von CRemoteData-Objekten.

Methode	Parameter oder Rückgabewert	Beschreibung
BSTR CurrentDirectory();	Parameter = BSTR	Der vollständige Pfadname des aktuellen pcAnywhere-Datenverzeichnisses, in dem pcAnywhere-Remote-Objekte gespeichert werden

Methoden	Parameter oder Rückgabewert	Beschreibung
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Parameter = LPCTSTR lpszNewDirectory Rückgabewert = BOOL	Ändert das aktuelle Verzeichnis, in dem pcAnywhere Remote-Objekte gespeichert sind. Parameter ist der Name eines vorhandenen Verzeichnisses Rückgabewert ist TRUE, wenn erfolgreich
BOOL FindFirst(LPCTSTR lpszPattern, BSTR FAR*);	Parameter = LPCTSTR lpszPattern	Suchmuster für Dateinamen, mit dem Objektdateien gefiltert werden (ein Sternchen [*] findet alle Dateien im aktuellen Verzeichnis).
BOOL FindFirst(pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName	Rückgabepuffer für den vollständigen Pfadnamen der Remote-Objektdatei, die mit dem angegebenen Muster übereinstimmt.
BOOL FindFirst	Rückgabewert = BOOL	TRUE, wenn eine Remote-Objektdatei gefunden wird, die mit dem angegebenen Muster übereinstimmt. Der vollständige Pfadname der übereinstimmenden Datei wird in pbstrFullQualName gespeichert
BOOL FindNext(BSTR FAR* pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName Rückgabewert = BOOL	Nachdem FindFirst() erfolgreich aufgerufen wurde, um den Namen einer Remote-Objektdatei im aktuellen Verzeichnis zu ermitteln, kann FindNext() aufgerufen werden, um die nächste Datei zu ermitteln, die mit dem Muster übereinstimmt Rückgabewert ist TRUE, wenn eine andere Remote-Objektdatei gefunden wird, die mit dem Muster übereinstimmt, das im Aufruf für FindFirst() angegeben wurde. Der vollständige Pfadname der übereinstimmenden Datei wird in pbstrFullQualName gespeichert
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Lädt ein CRemoteData-Objekt anhand des Dateinamens Parameter ist der vollständige Name der zu ladenden Remote-Objektdatei

Methoden	Parameter oder Rückgabewert	Beschreibung
LPDISPATCH RetrieveObject(short wAccessMode);	Parameter = short wAccessMode	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz. Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Objektkennwort. Darf NULL sein
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Lädt ein CRemoteDataEx-Objekt anhand des Dateinamens Parameter ist der vollständige Name der zu ladenden Remote-Objektdatei
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parameter = short wAccessMode	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz. Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Objektkennwort. Darf NULL sein
LPDISPATCH RetrieveObjectEx	Rückgabewert = LPDISPATCH	Zeiger auf ein OLE-Dispatch-Objekt. Das Objekt ist ein CRemoteDataEx-Objekt. Weitere Informationen hierzu finden Sie unter Visual C++ Beispielcode für Remote-Funktionen .

Methoden	Parameter oder Rückgabewert	Beschreibung
LPDISPATCH CreateObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Rückgabewert = LPDISPATCH	Erstellt ein CRemoteData-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück. Parameter ist der vollständige Name der Remote-Objektdatei für das neue Objekt Rückgabewert ist ein Zeiger auf ein OLE-Dispatch-Objekt. Das Objekt ist ein CRemoteData-Objekt. Weitere Informationen hierzu finden Sie unter Visual C++ Beispielcode für Remote-Funktionen .
LPDISPATCH CreateObjectEx(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Rückgabewert = LPDISPATCH	Erstellt ein CRemoteDataEx-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück Parameter ist der vollständige Name der Remote-Objektdatei für das neue Objekt Rückgabewert ist ein Zeiger auf ein OLE-Dispatch-Objekt. Das Objekt ist ein CRemoteDataEx-Objekt. Weitere Informationen hierzu finden Sie unter Visual C++ Beispielcode für Remote-Funktionen .
BOOL DeleteObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Löscht eine Remote-Objektdatei. Parameter ist der vollständige Name der Remote-Objektdatei des zu löschenden Objekts
BOOL DeleteObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Objektkennwort. Darf NULL sein
BOOL DeleteObject	BOOL	TRUE, wenn das Objekt gelöscht wurde.

Methode	Parameter oder Rückgabewert	Beschreibung
BOOL Launch(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Rückgabewert = BOOL	Startet eine Remote-Objektdatei, die das pcAnywhere-Remote-Terminal-Fenster öffnet. Parameter ist die vollständige Remote-Objektdatei des zu startenden Objekts Rückgabewert ist TRUE, wenn das Objekt erfolgreich gestartet wird

CRemoteData-Objekt

Verwenden Sie dieses Objekt für die Bearbeitung von Remote-Objektdateien.

Get- und Set-Methoden

Die folgenden Methoden werden zum Abrufen und Einstellen der Eigenschaften des CRemoteData-Objekts verwendet.

Der Computernamen ist der Name des pcAnywhere-Host-Computers, der beim Starten des Remote-Objekts aufgerufen werden soll.

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

Die Telefonnummer ist die Nummer, die gewählt werden muss, um eine Modemverbindung mit einem pcAnywhere-Host-Computer herzustellen.

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

Gibt an, ob TAPI-Wählparameter verwendet werden sollen (Standortinformation) (TRUE), oder ob der Telefonnummernstring exakt wie angezeigt zu verwenden ist (FALSE).

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

Wenn Wählparameter verwendet werden sollen, ist dies die Ortskennzahl der anzurufenden Nummer.

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

Wenn Wählparameter verwendet werden sollen, ist dies die Landeskennzahl der anzurufenden Nummer.

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

Die Zahl der Wahlwiederholungsversuche für diese Nummer, wenn der Anruf fehlschlägt.

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

Der Zeitraum (in Sekunden), der zwischen Wahlwiederholungsversuchen gewartet werden soll.

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

Der Login-Name, der beim Herstellen einer Verbindung an den Host gesendet werden soll. Ist hier nichts angegeben, so wird der Benutzer beim Verbindungsaufbau zur Eingabe eines Login-Namens aufgefordert.

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

Das Login-Kennwort, das beim Herstellen einer Verbindung an den Host gesendet werden soll. Ist hier nichts angegeben, so wird der Benutzer beim Verbindungsaufbau zur Eingabe eines Kennworts aufgefordert.

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

Das Kennwort für dieses Objekt.

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

Das Objekt kann nur gestartet werden, wenn das Kennwort verwendet wird (TRUE).

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

Das Objekt kann nur angezeigt werden, wenn das korrekte Kennwort eingegeben wird (TRUE).

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

Das Objekt kann nur geschrieben werden, wenn das korrekte Kennwort eingegeben wird (TRUE).

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

Hiermit wird festgelegt, ob Sitzungen, die dieses Objekt verwenden, aufgezeichnet werden.

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

Hiermit wird festgelegt, ob Sitzungen, die dieses Objekt verwenden, von Beginn an aufgezeichnet werden.

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

Der Name der Aufzeichnungsdatei für Sitzungen, die dieses Objekt verwenden.

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

Legt die Verbindungstypen fest.

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

Folgende Verbindungstypen stehen zur Verfügung:

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

Der Name eines TAPI-Geräts kann auch als Verbindungstyp verwendet werden. DEFAULT TAPI verwendet das erste TAPI-Gerät, das im System gefunden wird. Suchen Sie mit FirstConnectionType()/NextConnectionType() nach verfügbaren Geräten, wenn Sie ein bestimmtes TAPI-Gerät verwenden wollen.

Detailmethoden für Remote-Objekte

Wenn einem Remote-Objekt ein Verbindungstyp zugeordnet wird, werden für die Gerätedetails die gültigen Standardwerte eingestellt.

Die folgenden Verbindungstypen weisen erweiterte Konfigurationsoptionen auf, die in der Anwendung festgelegt werden können:

- COM-Geräte
- NetBIOS-Geräte
- ISDN über CAPI 2.0-Geräte (nur europäisches ISDN)

Details für das COM-Gerät

Legt die Paritätsebene für die Kommunikation fest.

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

Folgende Paritätswerte für die Kommunikation sind verfügbar:

- Keine
- Odd
- Even
- Mark
- Space

Legt die Flusssteuerungsebene fest.

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

Folgende Flusssteuerungswerte sind verfügbar:

- <Keine>
- XONXOFF
- RTS/CTS
- BOTH

Legt die Einstellung für den Verbindungsstart fest.

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

Folgende Werte für den Verbindungsstart sind verfügbar:

- Immer verbunden
- Empfangssignalpegel (DCD)
- Sendebereitschaft (CTS)
- Betriebsbereitschaft (DSR)
- Ankommender Anruf (RI)
- Empfangen von 2 <CR>'s
- Modemantwort

Legt die Werte für das Verbindungsende fest.

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

Folgende Werte für das Verbindungsende sind verfügbar:

- Immer verbunden
- Empfangssignalpegel (DCD)
- Sendebereitschaft (CTS)
- Betriebsbereitschaft (DSR)
- Ankommender Anruf (RI)

Legt die Verbindungsgeschwindigkeit fest.

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

Folgende Werte für die Verbindungsgeschwindigkeit sind verfügbar:

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

■ 115200

Details für das NetBIOS-Gerät

Legt die für diese Verbindung zu verwendende LANA-Nummer (LANA = LAN Adapter) fest.

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

Details für ISDN über CAPI 2.0-Gerät

Mit TRUE wird die Kanalbündelung (verwendet zwei ISDN-Kanäle für eine Verbindung) aktiviert.

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

Legt zusätzliche CAPI-Erweiterungen fest, die für die Kommunikation erforderlich sind.

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

Methoden für Remote-Objekte

In der folgenden Tabelle werden die normalen Methoden des Remote-Objekts beschrieben (diese Methoden werden nicht zum Abrufen und Festlegen von Eigenschaften verwendet):

Methode	Parameter oder Rückgabewert	Beschreibung
short ConnectionTypes();	Rückgabewert = Short	Die Zahl der Verbindungstypen, die auf diesem Computer verfügbar ist.

Methode	Parameter oder Rückgabewert	Beschreibung
BSTR FirstConnectionType(); and BSTR NextConnectionType();	Rückgabewert = BSTR	FirstConnectionType() und NextConnectionType() werden für die Aufzählung der verfügbaren Verbindungstypen verwendet. Diese Funktionen geben einen BSTR, d. h., den Namen eines verfügbaren Verbindungstyps zurück. Die zurückgegebenen Verbindungstypen können mit der Funktion SetConnectionType() verwendet werden. Der Rückgabewert ist der Name eines unterstützten Gerätetyps für den Verbindungsaufbau
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Parameter = LPCTSTR lpszConnectionTyp Rückgabewert = BOOL	Der Name eines Gerätetyps für den Verbindungsaufbau. Der Rückgabewert ist TRUE, wenn es den übermittelten Verbindungstyp auf dem Computer gibt
short CountryCodes();	Rückgabewert = Short	Gibt die Zahl der verfügbaren Landeskennzahlen zurück
BSTR FirstCountryCode(); und BSTR NextCountryCode();	Rückgabewert = BSTR	FirstCountryCode() und NextCountryCode() werden für die Aufzählung der verfügbaren Landeskennzahlen verwendet. Diese Funktionen geben einen BSTR, d. h., den Namen einer verfügbaren Landeskennzahl zurück. Die zurückgegebenen Werte können mit der Funktion SetCountryCode() verwendet werden Der Rückgabewert ist der erste oder nächste String für die Landeskennzahl
BOOL ReadObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword Rückgabewert = BOOL	Liest die Objektdaten aus der Remote-Objektdatei Der Parameter ist das Objektkennwort Der Rückgabewert ist TRUE, wenn das Objekt erfolgreich gelesen wird

Methode	Parameter oder Rückgabewert	Beschreibung
BOOL WriteObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword Rückgabewert = BOOL	Schreibt die Objektdaten in die Remote-Objektdatei. Der Parameter ist das Objektkennwort Der Rückgabewert ist TRUE, wenn das Objekt erfolgreich geschrieben wird

CRemoteDataEx-Objekt

Das CRemoteDataEx-Objekt beinhaltet die gesamte Funktionalität des CRemoteData-Objekts sowie die folgenden zusätzlichen Get- und Set-Methoden:

```
BSTR GetPrivateKey(); //Gibt die den privaten Schlüssel betreffenden
    Informationen zurück
void SetPrivateKey(LPCTSTR lpszNewValue);
BSTR GetCertificationName(); //Gibt den Namen des Zertifikats zurück
void SetCertificationName(LPCTSTR lpszNewValue);
short GetEncryptionLevel(); //Gibt den Wert der Verschlüsselungsstufe
    zurück
void SetEncryptionLevel(short nNewValue);
BOOL GetDenyLowerEncrypt(); //Gibt den Wert für DenyLowerEncrypt zurück
void SetDenyLowerEncrypt(BOOL bNewValue);
BSTR GetAutoDomain(); //Gibt den Wert für AutoDomain zurück
void SetAutoDomain(LPCTSTR lpszNewValue);
```

Visual C++ Beispielcode für Remote-Funktionen

Die folgende Visual C++ Beispielfunktion erstellt ein Remote-Objekt, setzt den Verbindungstyp auf TCP/IP, setzt den Computernamen auf die an die Funktion übergebene TCP/IP-Adresse und startet dann das Remote-Objekt:

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

    CRemoteDataManager remoteDM;

    CRemoteData remoteData;
```

```
// Erstellen Sie zuerst den CRemoteDataManager
remoteDM.CreateDispatch( _T( "WINAWSVR.RemoteDataManager" ) );
// Erstellen Sie als nächstes CRemoteData und hängen Sie es an
remoteData.AttachDispatch( remoteDM.CreateObject("Test", 0) );

// Legen Sie jetzt die erforderlichen Eigenschaften fest
remoteData.SetConnectionType("TCP/IP");

remoteData.SetComputerName(lpszAddress);

// Speichern Sie die Objektdaten
if (remoteData.WriteObject(0))
{
// Rufen Sie das Objekt auf
if (remoteData.Launch())
    bReturn = TRUE;
}
// Geben Sie das Remote-Objekt frei
remoteData.ReleaseDispatch();

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

return bReturn;
}
```

CHostDataManager-Methoden

Die folgende Tabelle beschreibt die CHostDataManager-Methoden einschließlich der Parameter und Rückgabewerte für das Zugreifen auf und das Steuern der CHostData-Objekte:

Methode	Parameter oder Rückgabewert	Beschreibung
BSTR CurrentDirectory();	Rückgabewert = BSTR	Gibt den vollständigen Pfadnamen des aktuellen Verzeichnisses zurück, in dem pcAnywhere-Hostobjekte gespeichert sind

Methode	Parameter oder Rückgabewert	Beschreibung
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Parameter = LPCTSTR lpszNewDirectory Rückgabewert = BOOL	Ändert das aktuelle Verzeichnis, in dem pcAnywhere-Hostobjekte gespeichert sind Der Parameter ist der Name eines vorhandenen Verzeichnisses Der Rückgabewert ist TRUE, wenn erfolgreich
BOOL FindFirst(LPCTSTR lpszPattern);	Parameter = LPCTSTR lpszPattern Rückgabewert = BOOL	Ermittelt anhand des vorgegebenen Suchmusters für Dateinamen die erste pcAnywhere-Hostobjektdatei (*.bhf) im aktuellen Verzeichnis Der Parameter ist das Suchmuster für Dateinamen, mit dem Objektdateien gefiltert werden (ein Sternchen [*] findet alle Hostdateien im aktuellen Verzeichnis) Der Rückgabewert ist TRUE, wenn eine Hostobjektdatei, die dem angegebenen Muster entspricht, gefunden wird. Der vollständige Pfadname der übereinstimmenden Datei wird in pbstrFullQualName gespeichert
BOOL FindFirst(BSTR FAR* pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName	Rückgabepuffer für den vollständigen Pfadnamen der Remote-Objektdatei, die mit dem angegebenen Muster übereinstimmt. Der Rückgabewert ist TRUE, wenn eine Hostobjektdatei, die dem angegebenen Muster entspricht, gefunden wird. Der vollständige Pfadname der übereinstimmenden Datei wird in pbstrFullQualName gespeichert.

Methoden	Parameter oder Rückgabewert	Beschreibung
BOOL FindNext(BSTR FAR * pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName Rückgabewert = BOOL	Nachdem FindFirst() erfolgreich aufgerufen wurde, um den Namen einer Hostobjektdatei im aktuellen Verzeichnis zu ermitteln, kann FindNext() aufgerufen werden, um die nächste Datei zu ermitteln, die mit dem Muster übereinstimmt Der Parameter ist der Rückgabepuffer für den vollständigen Pfadnamen der Hostobjektdatei, die mit dem Muster übereinstimmt, das im ursprünglichen Aufruf für FindFirst() angegeben wurde Rückgabewert ist TRUE, wenn eine andere Hostobjektdatei gefunden wird, die mit dem Muster übereinstimmt, das im Aufruf für FindFirst() angegeben wurde. Der vollständige Pfadname der übereinstimmenden Datei wird in pbstrFullQualName gespeichert
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Lädt ein CHostData-Objekt anhand des Dateinamens. Der Parameter ist der vollständige Name der zu ladenden Hostobjektdatei
LPDISPATCH RetrieveObject(short wAccessMode);	Parameter = short wAccessMode	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz. Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Objektkennwort. Darf NULL sein

Methoden	Parameter oder Rückgabewert	Beschreibung
LPDISPATCH RetrieveObject	Rückgabewert = LPDISPATCH	Zeiger auf ein OLE-Dispatch-Objekt. Das Objekt ist ein CHostData-Objekt. Weitere Informationen hierzu finden Sie unter Visual C++ Beispielcode für Hostfunktionen .
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Lädt ein CHostDataEx-Objekt anhand des Dateinamens Der Parameter ist der vollständige Name der zu ladenden Hostobjektdatei
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parameter = short wAccessMode	Gibt an, wie dieses Objekt zu verwenden ist. Dieser Parameter bezieht sich auf den Kennwortschutz. Folgende Optionen sind verfügbar: ■ 0 = Nicht angegeben ■ 1 = Nur anzeigen ■ 2 = Anzeigen und ändern ■ 3 = Ausführen
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Objektkennwort. Darf NULL sein
LPDISPATCH RetrieveObjectEx	Rückgabewert = LPDISPATCH	Zeiger auf ein OLE-Dispatch-Objekt. Das Objekt ist ein CHostDataEx-Objekt Weitere Informationen hierzu finden Sie unter Visual C++ Beispielcode für Hostfunktionen .
LPDISPATCH CreateObject(LPCTSTR lpszName);	Parameter = LPCTSTR lpszFQName Rückgabewert = LPDISPATCH	Erstellt ein CHostData-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück Der Parameter ist der vollständige Name der Hostobjektdatei für das neue Objekt Der Rückgabewert ist ein Zeiger auf ein OLE-Dispatch-Objekt. Das Objekt ist ein CHostData-Objekt Weitere Informationen hierzu finden Sie unter Visual C++ Beispielcode für Hostfunktionen .

Methode	Parameter oder Rückgabewert	Beschreibung
LPDISPATCH CreateObjectEx(LPCTSTR lpszName);	Parameter = LPCTSTR lpszFQName Rückgabewert = LPDISPATCH	Erstellt ein CHostDataEx-Objekt und gibt einen LPDISPATCH-Zeiger an das Objekt zurück Der Parameter ist der vollständige Name der Hostobjektdatei für das neue Objekt Der Rückgabewert ist ein Zeiger auf ein OLE-Dispatch-Objekt. Das Objekt ist ein CHostDataEx-Objekt. Weitere Informationen hierzu finden Sie unter Visual C++ Beispielcode für Hostfunktionen .
BOOL DeleteObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Löscht eine Hostobjektdatei Der Parameter ist der vollständige Name der Hostobjektdatei des zu löschenden Objekts
BOOL DeleteObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Objektkennwort. Darf NULL sein
BOOL DeleteObject	Rückgabewert = BOOL	TRUE, wenn das Objekt gelöscht wurde.
BOOL Launch(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Rückgabewert = BOOL	Startet eine Hostobjektdatei, die das pcAnywhere-Hostterminal-Fenster öffnet Der Parameter ist die vollständige Hostobjektdatei des zu startenden Objekts Der Rückgabewert ist TRUE, wenn das Objekt erfolgreich gestartet wird

CHostData-Objekt

Verwenden Sie dieses Objekt für die Bearbeitung von Host-Objektdaten.

Get- und Set-Methoden

Mit den folgenden Methoden werden Eigenschaften des CHostData-Objekts abgerufen und festgelegt.

Die Telefonnummer ist die Nummer, die gewählt werden muss, um eine Modemverbindung mit einem pcAnywhere Remote-Computer herzustellen.

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

Gibt an, ob TAPI-Wählparameter verwendet werden sollen (Standortinformation) (TRUE), oder ob der Telefonnummernstring exakt wie angezeigt zu verwenden ist (FALSE).

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

Wenn Wählparameter verwendet werden sollen, ist dies die Ortskennzahl der anzurufenden Nummer.

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

Wenn Wählparameter verwendet werden sollen, ist dies die Landeskenzahl der anzurufenden Nummer.

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

Die Zahl der Wahlwiederholungsversuche für diese Nummer, wenn der Anruf fehlschlägt.

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

Der Zeitraum (in Sekunden), der zwischen Wahlwiederholungsversuchen gewartet werden soll.

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

Hiermit wird festgelegt, ob Sitzungen, die dieses Objekt verwenden, aufgezeichnet werden.

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

Hiermit wird festgelegt, ob Sitzungen, die dieses Objekt verwenden, von Beginn an aufgezeichnet werden.

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

Der Name der Aufzeichnungsdatei für Sitzungen, die dieses Objekt verwenden.

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

Detailmethoden für Hostobjekte

Wenn einem Host-Objekt ein Verbindungstyp zugeordnet wird, werden für die Gerätedetails die gültigen Standardwerte eingestellt.

Die folgenden Verbindungstypen weisen erweiterte Konfigurationsoptionen auf, die in der Anwendung festgelegt werden können:

- COM-Geräte
- NetBIOS-Geräte
- NASI/NCSI-Geräte
- ISDN über CAPI 2.0-Geräte (nur europäisches ISDN)

Details für das COM-Gerät

Der folgende Code fügt den angeforderten Verbindungstyp in die Liste mit den zugeordneten Verbindungstypen des Host-Objekts ein und macht diesen Verbindungstyp bei der Verarbeitung nachfolgender gerätespezifischer Methodenaufrufe zum aktuellen Verbindungstyp.

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

Falls der angeforderte Verbindungstyp bereits in der Liste der zugeordneten Verbindungen vorhanden ist, wird diese Liste nicht geändert. Nur der aktuelle Verbindungstyp wird an den angeordneten Verbindungstyp angepasst. Es ist nicht ungewöhnlich, die Methode "AssignConnection" während des Abrufens und Definierens verbindungsspezifischer Werte mehrmals auf demselben Objekt aufzurufen.

"AssignConnection" gibt TRUE zurück, wenn es den übergebenen Verbindungstyp auf dem Computer gibt und er entweder erfolgreich zugewiesen wird oder bereits zugewiesen ist. Die Methode gibt FALSE zurück, wenn es entweder den angeforderten Verbindungstyp auf diesem Computer nicht gibt oder der aktuelle

Zähler für die zugeordnete Verbindung bereits seine maximal zulässige Höchstgrenze erreicht hat.

Ein pcAnywhere-Host-Objekt kann bis zu zwei zugeordnete Verbindungstypen unterstützen. Die Methode "AssignConnection" gibt FALSE zurück, wenn sie registriert, dass versucht wurde, diese Grenze zu überschreiten.

Folgende Verbindungstypen stehen zur Verfügung:

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

Der Name eines TAPI-Geräts kann auch als Verbindungstyp verwendet werden. DEFAULT TAPI verwendet das erste TAPI-Gerät, das im System gefunden wird. Suchen Sie mit FirstConnectionType() und NextConnectionType() nach verfügbaren Geräten, wenn Sie ein bestimmtes TAPI-Gerät verwenden möchten.

Der folgende Code hebt die Zuordnung eines Verbindungstyps auf. Nachdem die Zuordnung eines Verbindungstyps aufgehoben wurde, wird die verbleibende zugeordnete Verbindung, sofern vorhanden, zum aktuellen Verbindungstyp für nachfolgende gerätespezifische Methodenaufrufe.

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

Der folgende Code legt die Paritätsebenen für die Kommunikation fest:

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

Folgende Paritätswerte für die Kommunikation sind verfügbar:

- Keine
- Odd
- Even
- Mark

- Space

Der folgende Code legt die Flusssteuerungsebene fest:

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

Folgende Flusssteuerungswerte sind verfügbar:

- <Keine>
- XONXOFF
- RTS/CTS
- BOTH

Der folgende Code legt die Werte für den Verbindungsstart fest:

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

Folgende Werte für den Verbindungsstart sind verfügbar:

- Immer verbunden
- Empfangssignalpegel (DCD)
- Sendebereitschaft (CTS)
- Betriebsbereitschaft (DSR)
- Ankommender Anruf (RI)
- Empfangen von 2 <CR>'s
- Modemantwort

Der folgende Code legt die Werte für das Verbindungsende fest:

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

Folgende Werte für das Verbindungsende sind verfügbar:

- Immer verbunden
- Empfangssignalpegel (DCD)
- Sendebereitschaft (CTS)
- Betriebsbereitschaft (DSR)
- Ankommender Anruf (RI)

Der folgende Code legt die Werte für die Verbindungsgeschwindigkeit fest:

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

Folgende Werte für die Verbindungsgeschwindigkeit sind verfügbar:

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

Details für das NetBIOS-Gerät

Legt die für diese Verbindung zu verwendende LANA-Nummer (LANA = LAN Adapter) fest.

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

Details für das NASI/NCSI-Gerät

Legt den Benutzernamen für den NASI-Server fest.

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

Legt das Benutzerkennwort für den NASI-Server fest.

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

Legt den Namen der NASI-Sitzung fest.

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

Gibt den Namen des zu verwendenden NASI-Servers an.

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

Gibt den Namen des zu verwendenden NASI-Dienstes an.

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

Gibt den Namen des zu verwendenden NASI-Anschlusses an.

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

Details für ISDN über CAPI 2.0-Gerät

Mit TRUE wird die Kanalbündelung (verwendet zwei ISDN-Kanäle für eine Verbindung) aktiviert.

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

Legt zusätzliche CAPI-Erweiterungen fest, die für die Kommunikation erforderlich sind.

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

Methoden für Hostobjekte

In der folgenden Tabelle werden die normalen Methoden des Objekts beschrieben (diese Methoden werden nicht zum Abrufen und Festlegen von Eigenschaften verwendet):

Methoden	Parameter oder Rückgabewert	Beschreibung
short ConnectionTypes();	Rückgabewert = Short	Gibt die Anzahl der Verbindungstypen zurück, die auf diesem Computer verfügbar ist

Methode	Parameter oder Rückgabewert	Beschreibung
BSTR FirstConnectionType(); und BSTR NextConnectionType();	Rückgabewert = BSTR	FirstConnectionType() und NextConnectionType() werden für die Aufzählung der verfügbaren Verbindungstypen verwendet. Die Funktionen geben einen BSTR, d. h. den Namen eines verfügbaren Verbindungstyps zurück. Die zurückgegebenen Verbindungstypen können mit der Funktion SetConnectionType() verwendet werden Der Rückgabewert ist der Name eines unterstützten Gerätetyps für den Verbindungsaufbau
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Parameter = LPCTSTR lpszConnectionTyp Rückgabewert = BOOL	Der Parameter ist der Name eines Gerätetyps für den Verbindungsaufbau Der Rückgabewert ist TRUE, wenn dieser Gerätetyp verfügbar ist
short MaxAssignedConnections()	Rückgabewert = Short	Gibt die maximal zulässige Anzahl der Verbindungstypen zurück, die jeweils gleichzeitig zugeordnet werden können (derzeit zwei)
short AssignedConnections()	Rückgabewert = Short	Gibt die Zahl der zugeordneten Verbindungstypen zurück.
BSTR FirstAssignedConnection(); und BSTR NextAssignedConnection();	Rückgabewert = BSTR	FirstAssignedConnection() und NextAssignedConnection() werden für die Aufzählung der zugeordneten Verbindungen verwendet. Diese Funktionen geben einen BSTR, d. h., den Namen eines zugeordneten Verbindungstyps zurück. Die zurückgegebenen Verbindungstypen können mit der Funktion AssignConnection() verwendet werden Der Rückgabewert ist der Name eines unterstützten Gerätetyps für den Verbindungsaufbau

Methode	Parameter oder Rückgabewert	Beschreibung
BOOL FindAssignedConnection (LPCTSTR lpszConnectionType);	Parameter = LPCTSTR lpszConnectionTyp Rückgabewert = BOOL	Der Parameter ist der Name eines Gerätetyps für den Verbindungsaufbau Der Rückgabewert ist TRUE, wenn dieser Gerätetyp gegenwärtig zugeordnet ist
short CountryCodes();	Rückgabewert = Short	Gibt die Zahl der verfügbaren Landeskennzahlen zurück
BSTR FirstCountryCode(); und BSTR NextCountryCode();	Rückgabewert = BSTR	FirstCountryCode() und NextCountryCode() werden für die Aufzählung der verfügbaren Landeskennzahlen verwendet. Die Funktionen geben einen BSTR, d. h. den Namen einer verfügbaren Landeskennzahl zurück. Die zurückgegebenen Werte können mit der Funktion SetCountryCode() verwendet werden. Der Rückgabewert ist der erste oder nächste String für die Landeskennzahl
BOOL ReadObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword Rückgabewert = BOOL	Liest die Objektdaten aus der Hostobjektdatei Der Parameter ist das Objektkennwort Der Rückgabewert ist TRUE, wenn das Objekt erfolgreich gelesen wird
BOOL WriteObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword Rückgabewert = BOOL	Schreibt die Objektdaten in die Hostobjektdatei Der Parameter ist das Objektkennwort Der Rückgabewert ist TRUE, wenn das Objekt erfolgreich geschrieben wird

CHostDataEx-Objekt

Das CHostDataEx-Objekt beinhaltet die gesamte Funktionalität des CHostData-Objekts sowie die folgenden zusätzlichen Get- und Set-Methoden:

```

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

```

```
void SetWriteProtection(BOOL bNewValue);
BSTR GetPassword(); //Returns "NICHT IMPLEMENTIERT"
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); //Legt ActiveKbds fest
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);

```

Visual C++ Beispielcode für Hostfunktionen

Die folgende Visual C++ Beispielfunktion erstellt ein Hostobjekt, setzt den Verbindungstyp auf TCP/IP, setzt den Computernamen auf die an die Funktion übergebene TCP/IP-Adresse und startet dann das Hostobjekt:

```

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

    CHostDataManager hostDM;
    CHostData hostData;

    // Erstellen Sie zuerst den CHostDataManager
    hostDM.CreateDispatch( _T( "WINAWSVR.BeHostDataManager" ) );

```

```
// Erstellen Sie als nächstes CRemoteData und hängen Sie es an
hostData.AttachDispatch(hostDM.CreateObject("Test", 0) );

// Legen Sie jetzt die erforderlichen Eigenschaften fest
hostData.SetConnectionType("TCP/IP");

// Speichern Sie die Objektdaten
if (hostData.WriteObject(0))
{
// Rufen Sie das Objekt auf
if (hostData.Launch())
    bReturn = TRUE;
}

// Geben Sie das Hostobjekt frei.
hostData.ReleaseDispatch();

return (bReturn);
}
```

Awrem32-Funktionen

Die folgende Tabelle beschreibt die Awrem32-Funktionen einschließlich der Parameter und Rückgabewerte zum Verwalten von Verbindungen zwischen einem Host- und Remote-Computer:

Methode	Parameter oder Rückgabewert	Beschreibung
boolean awConnect(BSTR FileName);	Parameter = Name als Zeichenkette Rückgabewert = Boolean	Erstellt die Verbindung zum Hostcomputer Der Parameter ist der vollständige Name der .chf-Datei, die Informationen über den Hostcomputer enthält Der Rückgabewert führt den Befehl aus

Methode	Parameter oder Rückgabewert	Beschreibung
boolean awDisconnect();	Rückgabewert = Boolean	Trennt die Verbindung zum Hostcomputer Nach Aufruf dieser Funktion muss das aufrufende Programm das Objekt löschen (C++ - delete IAwrem32X*, VB - set ObjectName = Nothing;)
boolean FileXferFromHost(BSTR HostFile);	Parameter = HostFile als Zeichenkette	Kopiert eine Datei vom Hostcomputer auf den Remote-Computer. Die Parameter können Platzhalter enthalten Der Parameter enthält den vollständigen Pfad und Namen der Datei, die vom Hostcomputer kopiert werden soll. Die HostFile- und RemoteFile-Strings müssen nicht identisch sein
boolean FileXferFromHost(BSTR RemoteFile);	Parameter = RemoteFile als Zeichenkette	Kopiert eine Datei vom Hostcomputer auf den Remote-Computer. Die Parameter können Platzhalter enthalten Der Parameter enthält den vollständigen Zielpfad und Dateinamen. Die HostFile- und RemoteFile-Strings müssen nicht identisch sein.
boolean FileXferFromHost	Rückgabewert = Boolean	TRUE nach Ausführen des Befehls
boolean FileXferToHost(BSTR HostFile);	Parameter = HostFile als Zeichenkette	Kopiert eine Datei vom Remote-Computer auf den Hostcomputer. Die Parameter können Platzhalter enthalten Der Parameter enthält den vollständigen Zielpfad und Dateinamen. Die HostFile- und RemoteFile-Strings müssen nicht identisch sein

Methode	Parameter oder Rückgabewert	Beschreibung
boolean FileXferToHost(BSTR RemoteFile);	Parameter = RemoteFile als Zeichenkette	Kopiert eine Datei vom Remote-Computer auf den Hostcomputer. Die Parameter können Platzhalter enthalten Der Parameter enthält den vollständigen Pfad und Namen der Datei, die vom Remote-Computer kopiert werden soll. Die HostFile- und RemoteFile-Strings müssen nicht identisch sein
boolean FileXferToHost	Rückgabewert = Boolean	TRUE nach Ausführen des Befehls
boolean CreateFolderOnHost(BSTR FolderName);	Parameter = FolderName als Zeichenkette Rückgabewert = Boolean	Erstellt einen neuen Ordner auf dem Hostcomputer. Diese Funktion erstellt einen temporären Ordner auf dem Remote-Computer und kopiert diesen Ordner anschließend auf den Hostcomputer Der Parameter enthält das Laufwerk und den Pfad zum Erstellen des Ordners auf dem Hostcomputer Der Rückgabewert ist TRUE nach Ausführen des Befehls
boolean ExecuteHostFile(BSTR FileName);	Parameter =Dateiname als Zeichenkette Rückgabewert = Boolean	Führt eine vorhandene Datei auf dem Hostcomputer aus. Diese Funktion führt nur Stapel-, Befehls- und ausführbare Dateien aus. Es werden keine Dateien ausgeführt, die mit ausführbaren Dateien verknüpft sind. Beispiel: Wenn Sie eine .DOC-Datei ausführen, wird Microsoft Word nicht geöffnet Der Parameter enthält den vollständigen Pfad zu der auf dem Hostcomputer gespeicherten Datei Der Rückgabewert ist TRUE nach Ausführen des Befehls

Methode	Parameter oder Rückgabewert	Beschreibung
BSTR GetError();	Rückgabewert = Zeichenkette	Gibt den letzten Fehler zurück, der in Awrem32 als Zeichenkette aufgetreten ist
short ConnectionStatus();	Rückgabewert = Short	Gibt den aktuellen Status der Verbindung mit dem Hostcomputer zurück Die folgenden Werte sind verfügbar: ■ -1 = Verbindung unterbrochen ■ 0 = Keine Verbindung ■ 1 = Verbundene Sitzung

Index

A

- API-Bibliotheken 14
- Automatisierungs-Controller
 - Informationen zu Visual Basic 12
 - Informationen zu Visual C++ 14
- Automatisierungs-Server.. *Siehe* pcAnywhere
- Automatisierungs-Server
- Awrem32
 - Funktionen 58, 89

C

- C++.. *Siehe* Visual C++
- CAPI-Eigenschaften
 - Host-Eigenschaften 48, 84
 - Remote-Eigenschaften 30, 71
- CHostData
 - Verwenden von C++ 78
 - Verwenden von Visual Basic 37
- CHostDataEx
 - Verwenden von C++ 86
 - Verwenden von Visual Basic 48
- CHostDataManager
 - Verwenden von C++ 74
 - Verwenden von Visual Basic 34
- Codebeispiele
 - Visual Basic 32, 56
 - Visual C++ 73, 88
- COM-Geräte
 - Details für das Host-Gerät 44, 80
 - Details für das Remote-Gerät 26, 69
- CRemoteData
 - Verwenden von C++ 66
 - Verwenden von Visual Basic 21
- CRemoteDataEx
 - Verwenden von C++ 73
 - Verwenden von Visual Basic 31
- CRemoteDataManager
 - Verwenden von C++ 62
 - Verwenden von Visual Basic 18

D

- Dateien
 - Ausführen auf Host 59, 91
- Dateiübertragung
 - von Host 58, 90
 - zu Host 59, 90
- Datenflusssteuerung 27, 44, 69, 81

F

- Fehler
 - Zurückwechseln 59, 92

G

- Geräte.. *Siehe* Verbindungstypen
- GUIDs 11

H

- Hostobjekte
 - Ermitteln 35–36, 75, 77
 - Erstellen 77
 - Gerätedetails 80
 - Kennwörter 86
 - löschen 36, 78
 - Methoden 84
 - Starten 37, 78
 - Verzeichnisse 34, 74
 - Wählparameter 43

I

- ISDN
 - CAPI-Eigenschaften
 - Host-Eigenschaften 48, 84
 - Remote-Eigenschaften 30, 71

K

- Kennungen. *Siehe* GUIDs
- Kennwörter
 - für Host-Objekte 86
 - für Remote-Objekte 72

Klassendefinitionen
 Anzeigen 15
 Importieren 14

L

Landeskennzahl
 auf dem Host 86
 für Remote 72

M

Modemverbindungen
 auf dem Host
 COM-Eigenschaften 44, 80
 Landeskennzahl 86
 TAPI-Geräte 81
 Wählparameter 43
 für Remote
 COM-Eigenschaften 26, 69
 Landeskennzahl 72
 TAPI-Geräte 68
 Wählparameter 25

N

NASI-Geräte 83
 NCSI-Geräte 83
 NetBIOS-Verbindungen
 Host-Eigenschaften 47, 83
 Remote-Eigenschaften 29, 71
 Netzwerkverbindungen
 auf dem Host
 NetBIOS-Eigenschaften 47, 83
 für Remote
 NetBIOS-Eigenschaften 29, 71

O

OLE-Automatisierung.. *Siehe* pcAnywhere
 Automatisierungs-Server
 Ordner
 Erstellen auf Host 59, 91

P

Parität 27, 44, 69, 81
 pcAnywhere Automatisierungs-Server
 Informationen 10
 Registrierung der GUIDs 11
 Typbibliotheken 14
 Verwendungsbeispiele 10

Zugriff mit Visual Basic 12
 Zugriff mit Visual C++ 14

R

Remote-Modul
 automatische Registrierung 11
 manuelle Registrierung 12
 Remote-Objekte
 Dateien 19–20, 35, 63–64
 Erstellen 65
 Gerätedetails 69
 Kennwörter 72
 löschen 65
 Methoden 71
 Verzeichnisse 19, 62
 Wählparameter 25

S

Status
 Zurückwechseln 59, 92

T

TAPI-Geräte
 auf dem Host 81
 für Remote 68
 Typbibliotheken 14

V

Verbindungen
 beenden 58, 90
 Starten 37, 58, 66, 89
 Zurückgeben des Status 59, 92
 Zurückgeben von Fehlern 92
 Verbindungstypen
 Aufheben der Zuordnung 42
 Host-Eigenschaften 39, 80, 84
 Remote-Eigenschaften 24, 69, 71
 Zuordnen 42
 Verschlüsselungseinstellungen
 auf dem Host 48, 86
 für Remote 31–32, 73
 Visual Basic
 Awrem32-Funktionen 58
 CHostData-Objekt 37
 CHostDataEx-Objekt 48
 CHostDataManager-Objekt 34
 Codebeispiele 32, 56
 CRemoteData-Objekt 21

- CRemoteDataEx-Objekt 31
- CRemoteDataManager-Objekt 18
- Zugriff auf den pcAnywhere
 - Automatisierungs-Server 12

Visual C++

- Awrem32-Funktionen 89
- CHostData-Objekt 78
- CHostDataEx-Objekt 86
- CHostDataManager-Objekt 74
- Codebeispiele 73, 88
- CRemoteData-Objekt 66
- CRemoteDataEx-Objekt 73
- CRemoteDataManager-Objekt 62
- einschließlich Winawsvr.h 16
- Importieren von Klassen 14
- Zugriff auf den pcAnywhere
 - Automatisierungs-Server 14

W

Winawsvr

- ausführbare Datei 12
- Header-Datei 16
- Objekte
 - CHostData 37, 78
 - CHostDataEx 48, 86
 - CHostDataManager 34, 74
 - CRemoteData 21, 66
 - CRemoteDataEx 31, 73
 - CRemoteDataManager 18, 62