

OLE 자동화 설명서

OLE 자동화 설명서

법적 고지

Copyright © 2008 Symantec Corporation. All rights reserved.

Symantec 및 Symantec 로고는 미국 및 기타 국가에서 Symantec Corporation 또는 그 자회사의 상표 또는 등록 상표입니다. 다른 이름은 각 소유권자의 상표일 수 있습니다.

본 Symantec 제품에는 Symantec이 타사에 권한을 제공해야 하는 타사 소프트웨어("타사 프로그램")가 포함될 수 있습니다. 일부 타사 프로그램은 오픈 소스 또는 무료 소프트웨어 라이선스에 따라 사용 가능합니다. 소프트웨어의 라이선스 계약서는 이러한 오픈 소스 또는 무료 소프트웨어 라이선스에 의해 적용될 수 있는 권한 또는 의무를 변경하지 않습니다. 타사 프로그램에 대한 자세한 내용은 본 문서의 타사 법적 고지 부록 또는 본 Symantec 제품의 TPIP ReadMe 파일을 참조하십시오.

이 문서에 설명된 제품은 사용, 복사, 배포 및 디컴파일/리버스 엔지니어링을 제한하는 라이선스 하에 배포됩니다. 이 문서의 어떤 부분도 Symantec Corporation 및 그 라이선스 제공자의 사전 서면 승인 없이 어떤 방식, 어떤 형태로도 복제될 수 없습니다.

이 문서는 "있는 그대로" 제공되며, 상품성, 특정 목적에의 적합성, 비침해성에 대한 묵시적인 보증을 비롯한 모든 명시적/묵시적 조건, 제시, 보증을 하지 않습니다. 단 이러한 조건, 진술 및 보증의 배제가 법적으로 무효가 되는 경우는 예외로 합니다. 시만텍 기업은 본 문서의 공급 이행 또는 사용으로 인한 부수적 또는 결과적 손해에 대해 책임을 지지 않습니다. 이 문서에 포함된 정보는 예고 없이 변경될 수 있습니다.

라이선스를 부여받은 소프트웨어와 사용 설명서는 FAR 12.212에 정의된 상업용 컴퓨터 소프트웨어로 간주되며 FAR 조항 52.227-19 "상업용 컴퓨터 소프트웨어 - 제한된 권리" 및 DFARS 227.7202, "상업용 컴퓨터 소프트웨어 또는 상업용 컴퓨터 소프트웨어 사용 설명서의 권리" 및 이후 모든 규정에 정의된 제한된 권리에 종속됩니다. 라이선스를 부여받은 소프트웨어 및 사용 설명서에 대한 미국 정보의 사용, 수정, 복제 해제, 수행, 표시 또는 공개는 본 계약서의 조건에 준하여 수행해야 합니다.

시만텍코리아(주)
서울특별시 강남구 역삼1동 737
강남파이낸스센터 28층

<http://www.symantec.co.kr>

10987654321

기술 지원

시만텍 기술 지원 그룹은 전세계에 걸쳐 지원 센터를 운영합니다. 기술 지원 그룹의 주된 역할은 제품의 사양과 기능에 대한 질문에 응답하고 온라인 기술 자료의 콘텐츠를 작성하는 것입니다. 기술 지원 그룹은 사용자의 질문에 적시에 응답하기 위해 시만텍 내의 다른 부서와 상호 협력하여 일하고 있습니다. 예를 들어, 기술 지원 그룹은 알람 서비스 (Alerting service) 및 바이러스 정의 업데이트를 제공하기 위해서 제품 개발팀과 시만텍 보안 연구소(Symantec Security Response)와 함께 작업합니다.

시만텍의 유지 관리 프로그램은 다음과 같은 사항을 제공합니다.

- 회사 규모에 맞는 서비스를 선택할 수 있는 다양한 지원 옵션
- 신속한 응답과 최신 정보를 제공하는 전화 및 웹 지원
- 자동 소프트웨어 업그레이드 보호 기능을 제공하는 업그레이드 보증
- 전세계에서 연중무휴로 제공되는 글로벌 지원 서비스
- 계정 관리 서비스(Account Management Service)를 비롯한 고급 기능

시만텍의 유지 관리 프로그램에 대한 자세한 내용은 다음 웹 사이트에서 확인할 수 있습니다.

<http://www.symantec.com/region/kr/techsupp>

기술 지원 그룹에 문의

현재 유지 관리 계약 하에 있는 고객은 다음의 웹 사이트를 통해 기술 지원 그룹에 액세스할 수 있습니다.

<http://www.symantec.com/region/kr/techsupp>

기술 지원에 문의하기 전에 제품 설명서에 적혀 있는 시스템 요건을 갖추고 있는지 확인하십시오. 또, 문제가 발생한 상황을 재현해야 할 경우에 대비해서 해당 시스템을 사용할 수 있는 위치에서 문의해야 합니다.

기술 지원에 문의할 때는 다음 정보가 필요합니다.

- 제품 버전
- 하드웨어 정보
- 사용 가능 메모리, 디스크 공간 및 NIC 정보
- 운영 체제
- 버전 및 패치 번호
- 네트워크 토폴로지
- 라우터, 게이트웨이, IP 주소 정보
- 문제 설명

- 오류 메시지 및 로그 파일
- 시만텍에 문의하기 전에 문제를 해결하기 위해 수행한 작업
- 최근에 수행한 소프트웨어 설정 변경 사항 및 네트워크 변경 사항

제품 라이선스 및 등록

등록이나 라이선스 키가 필요한 제품일 경우 아래의 URL로 기술 지원 웹 페이지에 액세스하십시오.

<http://www.symantec.com/region/kr/techsupp>

고객 서비스

고객 서비스 정보는 다음 URL에서 확인할 수 있습니다.

<http://www.symantec.com/region/kr/techsupp>

고객 서비스를 통해 도움을 받을 수 있는 문제 유형은 다음과 같습니다.

- 제품 라이선스 및 일련 번호 관련 질문
- 주소나 이름 변경과 같은 제품 등록 정보 업데이트
- 일반 제품 정보(기능, 사용 가능 언어, 지역별 제공업체)
- 제품 업데이트 및 업그레이드에 관한 최신 정보
- 업그레이드 보증 및 유지 관리 계약에 관한 정보
- Symantec Buying Programs 정보
- 시만텍 기술 지원 옵션 추가 정보
- 일반 구매에 대한 모든 정보
- CD-ROM 또는 매뉴얼과 관련된 문제

유지 관리 계약 자원

기존 유지 관리 계약에 관한 문의는 아래에 있는 각 지역별 연락 정보로 해당 지역의 유지 관리 담당 팀에 문의하십시오.

아시아 태평양 지역 및 일본

contractsadmin@symantec.com

유럽, 중동 및 아프리카

semea@symantec.com

북미 및 중남미

supportsolutions@symantec.com

추가 엔터프라이즈 서비스

시만텍은 기업의 시만텍 제품에 대한 투자 효과를 극대화하고 전문적 지식과 정보, 통찰력을 통해 비즈니스 위험을 앞서 관리할 수 있도록 하는 종합적인 서비스를 제공하고 있습니다.

제공되는 엔터프라이즈 서비스는 다음과 같습니다.

시만텍 조기 경고 솔루션
(Symantec Early Warning Solutions)

이 솔루션은 사이버 공격에 대한 조기 경고, 포괄적인 위험 요소 분석, 공격 발생 전 예방 조치 등을 제공합니다.

관리형 보안 서비스
(Managed Security Services)

이 서비스는 보안 장치와 이벤트에 대한 관리 및 감시의 부담을 덜어줌으로써 실제 위험 요소에 대해 신속하게 대응할 수 있도록 합니다.

컨설팅 서비스

시만텍 컨설팅 서비스는 시만텍과 파트너사의 기술 전문가를 현장으로 파견하는 서비스를 제공합니다. 시만텍 컨설팅 서비스는 기업의 IT 자원에 대한 무결성과 가용성을 구축 및 관리하는데 중점을 둔 다양한 패키지를 제공하며 이러한 패키지는 평가, 설계, 구현, 감시, 관리와 같은 여러 가지 옵션을 고객 환경에 맞춰 구성할 수 있습니다.

교육 서비스

교육 서비스는 기술 교육, 보안 교육, 보안 인증, 경계 커뮤니케이션 프로그램 등 다양한 프로그램이 다각적으로 운영됩니다.

엔터프라이즈 서비스에 대한 자세한 내용은 다음 웹 사이트에서 확인하십시오.

www.symantec.co.kr

사이트맵에서 국가나 언어를 선택하십시오.

목차

기술 지원	3
1장 Symantec pcAnywhere와 함께 OLE 자동화 사용	9
OLE 자동화 정보	9
pcAnywhere 자동화 서버 정보	10
Symantec 자동화 서버로 수행할 수 있는 작업	10
시작하기 전에	10
원격지 엔진 자동 등록	11
원격지 엔진 수동 등록	11
pcAnywhere 자동화 서버 액세스	11
Visual Basic으로 pcAnywhere 자동화 서버에 액세스	11
Visual C++로 pcAnywhere 자동화 서버에 액세스	12
호스트 및 원격지 OLE 개체 시작	14
추가 정보 찾기	15
2장 Visual Basic 개체 정의	17
Visual Basic 개체 정보	17
CRemoteDataManager 메소드	18
CRemoteData 속성	20
연결 유형 속성	23
전화 걸기 등록 정보	24
COM 장치 등록 정보	25
NetBIOS 장치 속성	28
ISDN via CAPI 2.0 장치 등록 정보	28
CRemoteDataEx 개체	29
원격지 기능의 Visual Basic 샘플 코드	30
CHostDataManager 메소드	32
CHostData 속성	34
연결 유형 속성	36
AssignConnection(ConnectionType) 메소드	38
UnassignConnection(ConnectionType) 메소드	39
전화 걸기 등록 정보	39
COM 장치 등록 정보	40
NetBIOS 장치 속성	43
ISDN via CAPI 2.0 장치 등록 정보	43

CHostDataEx 개체 44

 호스트 기능의 Visual Basic 샘플 코드 50

Awrem32 함수 52

3장

Visual C++ 개체 정의 55

Visual C++ 개체 정보 55

CRemoteDataManager 메소드 56

CRemoteData 개체 59

 Get/Set 메소드 59

 원격지 개체 세부 사항 메소드 61

 원격지 개체 메소드 64

CRemoteDataEx 개체 65

 원격지 함수의 Visual C++ 샘플 코드 65

CHostDataManager 메소드 66

CHostData 개체 70

 Get/Set 메소드 70

 호스트 개체 세부 사항 메소드 71

 호스트 개체 메소드 75

CHostDataEx 개체 77

 호스트 함수의 Visual C++ 샘플 코드 78

Awrem32 함수 79

색인 83

Symantec pcAnywhere와 함께 OLE 자동화 사용

이 장의 내용은 다음과 같습니다.

- [OLE 자동화 정보](#)
- [pcAnywhere 자동화 서버 정보](#)
- [Symantec 자동화 서버로 수행할 수 있는 작업](#)
- [시작하기 전에](#)
- [추가 정보 찾기](#)

OLE 자동화 정보

OLE 자동화는 한 응용 프로그램에서 공개된 함수를 제어하고 자동화할 수 있는 외부 응용 프로그램 또는 기타 개발 도구(예: 스크립트 또는 매크로)를 작성할 수 있는 기술입니다.

OLE 자동화의 구성 요소는 다음과 같습니다.

OLE 자동화 서버	다른 응용 프로그램 또는 개발 도구에서 액세스하거나 제어할 수 있도록 자신의 함수를 공개하는 응용 프로그램이나 소프트웨어 구성 요소
	pcAnywhere 자동화 서버는 OLE 자동화 서버의 예제입니다.
OLE 자동화 제어기	OLE 자동화 서버로 공개된 구성 요소에 액세스하고 이를 제어하는 응용 프로그램이나 개발 도구
	OLE 자동화를 지원하는 모든 프로그래밍 언어를 사용할 수 있습니다. 가장 일반적인 프로그래밍 언어 두 가지는 Microsoft Visual Basic과 Microsoft Visual C++입니다.

외부 응용 프로그램은 서버에 연결한 후 하나 이상의 게시된 인터페이스에 액세스를 요청하여 OLE 자동화 서버에 액세스합니다. 인터페이스는 하나 이상의 관련 메소드 또는 속성에 액세스할 수 있는 진입 위치입니다. 한 응용 프로그램이 외부 응용 프로그램의 일부이더라도 서버에서 인터페이스를 불러오면 모든 내부 인터페이스 메소드를 호출할 수 있습니다.

pcAnywhere 자동화 서버 정보

pcAnywhere 자동화 서버는 외부 응용 프로그램에서 pcAnywhere 원격지 및 호스트 파일을 관리하도록 하여 원격지 제어 및 파일 전송 작업을 자동화할 수 있습니다. pcAnywhere 자동화 서버는 Symantec pcAnywhere 사용자 인터페이스에 대해 프로그램이 가능한 대체 수단으로써 대부분의 기본 동작을 미리링합니다.

예를 들어 pcAnywhere에 호스트 개체를 만들면 사용할 수 있는 첫 번째 모델 TAPI 장치가 기본값으로 할당됩니다. 유사한 방법으로 pcAnywhere 자동화 서버를 사용하여 호스트 개체를 만들고, 할당된 연결 목록을 통해 차례로 열거하면 사용할 수 있는 첫 번째 모델 TAPI 장치가 이미 할당되어 있습니다.

Symantec 자동화 서버로 수행할 수 있는 작업

pcAnywhere 자동화 서버를 사용하여 다양한 관리 및 성능 향상 작업을 자동화할 수 있습니다.

pcAnywhere 자동화 서버를 사용하여 다음을 수행할 수 있습니다.

- 네트워크에 있는 여러 시스템에 소프트웨어 업데이트를 자동으로 배포 및 설치
- 감사 또는 보관의 목적으로 각 시스템간의 자동 파일 전송 예약
- 네트워크에 있는 모든 pcAnywhere 호스트의 허용된 호출자 목록에 이름을 자동으로 추가 또는 삭제

이 설명서에는 pcAnywhere 자동화 서버의 연결 및 사용 방법을 Visual Basic과 Visual C++로 기록하여 설명한 여러 가지 예제가 있습니다.

17페이지의 “[Visual Basic 개체 정보](#)” 참조

55페이지의 “[Visual C++ 개체 정보](#)” 참조

시작하기 전에

pcAnywhere 자동화 서버 및 해당 인터페이스에 연결하는 동안, 식별자 매개 변수인 GUID(Globally Unique Identifier)가 자동화 라이브러리 API 함수로 전송됩니다. 개별 GUID가 pcAnywhere 자동화 서버와 각각의 공개된 인터페이스에 할당됩니다. 외부 응용 프로그램을 pcAnywhere 자동화 서버 및 해당 인터페이스에 연결하려면 이러한 GUID가 시스템 레지스트리에 존재해야 합니다.

Symantec pcAnywhere가 설치된 시스템에 외부 응용 프로그램을 실행하고 있을 경우 자동으로 GUID 항목을 등록할 수 있습니다. 또는 GUID 항목을 직접 입력해야 합니다.

11페이지의 “원격지 엔진 자동 등록” 참조

11페이지의 “원격지 엔진 수동 등록” 참조

원격지 엔진 자동 등록

OLE 클라이언트를 통해 처음으로 다른 시스템에 연결할 때, 원격지 엔진을 자동 등록해야 합니다. Symantec pcAnywhere에서 원격지 개체를 시작할 때 자동으로 원격지 엔진 등록도 함께 이루어집니다.

원격지 엔진을 자동으로 등록하려면 다음과 같이 하십시오.

- 1 Symantec pcAnywhere를 열려면 다음 중 하나를 수행하십시오.
 - 바탕 화면에서 Symantec pcAnywhere 프로그램 아이콘을 두 번 누르십시오.
 - Windows 작업 표시줄에서 [시작] > [프로그램] > [Symantec pcAnywhere]를 누르십시오.
- 2 [pcAnywhere 관리자] 창에서 [원격지]를 누르십시오.
- 3 원격지 연결 항목을 두 번 누르십시오.

이렇게 하면 원격지 엔진이 등록됩니다. 사용자가 연결을 완료할 필요가 없습니다.

원격지 엔진 수동 등록

외부 응용 프로그램을 실행 중인 시스템에 pcAnywhere가 설치되어 있지 않으면 pcAnywhere 자동화 서버 실행 파일(Winawsvr.exe)을 실행하여 GUID를 수동으로 등록해야 합니다. 레지스트리에 GUID를 추가하기 위해 실행 파일을 한 번만 실행하면 됩니다. Winawsvr.exe 파일은 설치 디렉터리에 있습니다.

pcAnywhere 자동화 서버 액세스

OLE 자동화를 지원하는 모든 언어 플랫폼을 통해 pcAnywhere 자동화 서버에 액세스할 수 있습니다. OLE 자동화를 지원하는 가장 일반적인 두 가지 언어 플랫폼은 Visual Basic과 Visual C++입니다.

Visual Basic 환경에서는 많은 하위 수준 작업이 Visual Basic 런타임 시스템에 의해 백라운드에서 수행되기는 하지만, 이 두 가지 플랫폼의 코딩 원칙은 유사합니다.

Visual Basic으로 pcAnywhere 자동화 서버에 액세스

Visual Basic 프로그램 언어는 pcAnywhere 자동화 서버와 같은 OLE 자동화 서버와의 상호 작용을 지원하는 내장 함수를 가지고 있습니다. Standard Exe 프로젝트를 작성한 후 각 메소드에 코드를 입력하여 pcAnywhere 자동화 서버에 액세스합니다. Visual Basic은 원

본 파일에서 고급 메소드 호출을 가져온 후, 해당 하위 수준의 OLE 자동화 메소드 호출에 내부적으로 이를 확장합니다.

Visual Basic으로 pcAnywhere 자동화 서버에 액세스하려면 다음과 같이 하십시오.

- 1 액세스할 각각의 pcAnywhere 개체에 대해 [Object] 변수 조합을 추가하십시오.

예를 들어 원격지 개체로 작업할 때, **RemoteDataManager** 및 **RemoteDataObject**를 [Object]로 DIM하십시오.

- 2 RemoteDataManager를 사용하여 원격지 개체의 데이터 관리자에 첨부하십시오.

예를 들어 **WINAWSVR.REMOTEDATAMANAGER**를 매개 변수로 사용하여 **CreateObject** 메소드를 호출하십시오.

Visual Basic은 레지스트리에서 관리자의 식별자를 찾기 위해 텍스트 매개 변수를 사용하고, 해당 관리자로 인터페이스를 반환합니다.

- 3 올바른 데이터 관리자 개체가 있으면, 그 개체를 사용하여 다음 중 하나를 실행하십시오.

- 현재 디렉터리 식별.
- 다른 디렉터리로 변경.
- 현재 디렉터리에 있는 관련 데이터 개체 파일을 순서대로 열거.
- 데이터 개체 파일의 작성, 검색 또는 삭제.

- 4 데이터 개체를 만들거나 검색한 후, 개체의 속성을 불러오거나 설정할 수 있습니다.

Visual Basic 구문은 속성 이름의 값을 불러오고 설정하는 것을 구별하기 위해 속성 이름을 사용하지 않습니다. 대신에 할당 연산자에 관한 속성의 위치가 기본적인 방법 호출이 Get인지 Set인지 식별합니다.

다음 예제는 Get과 Set을 설명합니다.

- 개체의 전화 번호 값을 불러오려면 속성 이름을 할당 연산자의 오른쪽에 배치하십시오.

예를 들어 `s = RemoteData.PhoneNumber()`이고, 여기서 s는 문자열 변수입니다.

- 전화 번호를 설정하려면 속성 이름을 할당 연산자의 왼쪽에 배치하십시오.

예를 들어 `RemoteData.PhoneNumber = "555-1212"`입니다.

Visual C++로 pcAnywhere 자동화 서버에 액세스

pcAnywhere 자동화 서버는 Visual C++로 쓰여진 형식 라이브러리를 사용하여 인터페이스와 메소드에 관한 정보를 자동화 클라이언트에 공개합니다. 이러한 형식 라이브러리는 Visual C++ ClassWizard를 사용하여 사용자의 응용 프로그램으로 가져올 수 있는 MFC(Microsoft Foundation Classes)를 사용합니다.

형식 라이브러리에서 제공되는 데이터 관리자 클래스는 **pcAnywhere** 자동화 서버로 인터페이스를 불러오는 데 필요한 기능을 제공하며, 인터페이스의 관련 개체 유형에서 고급 작업을 수행합니다.

데이터 관리자 개체를 사용하여 다음 중 하나를 수행하십시오.

- 현재 디렉터리를 확인 또는 변경.
- 현재 디렉터리에 있는 데이터 개체 파일 목록을 차례로 열거.
- 이름이 지정된 개체의 작성, 검색 또는 삭제.

개체를 만들거나 검색하면 이 개체는 공개된 모든 속성을 확인하거나, 수정하기 위해 관련된 개체 클래스를 사용합니다. 이러한 속성의 대부분은 **Get** 또는 **Set**으로 시작하는 메소드의 조합을 통해 공개됩니다. 예를 들어 사용자는 개체의 현재 전화 번호 속성을 확인하기 위해 **GetPhoneNumber** 방법을 호출하고, 설정하기 위해 **SetPhoneNumber**를 호출합니다.

클래스 가져오기 및 보기

pcAnywhere 자동화 서버는 다음 형식 라이브러리를 사용합니다.

Winawsvr.tlb **pcAnywhere** 자동화 서버에 연결하고 해당 인터페이스에 액세스하는 데 필요한 정보를 제공합니다.

Awrem32.tlb **pcAnywhere** 연결을 제어하는 데 필요한 정보를 제공합니다.

다음 절차는 **pcAnywhere** 자동화 서버 형식 라이브러리에서 **MFC** 응용 프로그램으로 클래스 정의를 가져온 후, 응용 프로그램에 추가된 클래스를 보는 방법을 설명합니다.

클래스를 가져오려면 다음과 같이 하십시오.

- 1 Visual C++에서 **MFC** 응용 프로그램을 만드십시오.
- 2 [View] 메뉴에서 **[Class Wizard]**를 누르십시오.
- 3 [Class Wizard] 대화 상자에서 **[Add Class]**와 **[From a type library]**를 누르십시오.
- 4 **[winawsvr.tlb]**를 두 번 누르십시오.
- 5 모든 클래스 정의를 가져오려면 [Confirm Classes] 대화 상자에서 **[OK]**를 누르십시오.
- 6 [Class Wizard] 대화 상자에서 **[Add Class]**와 **[From a type library]**를 누르십시오.
- 7 **[awrem32.tlb]**를 두 번 누르십시오.

- 8 모든 클래스 정의를 가져오려면 **[Confirm Classes]** 대화 상자에서 **[OK]**를 누르십시오.
- 9 가져오기 프로세스를 완료하려면 **[Class Wizard]** 대화 상자에서 **[OK]**를 누르십시오.
응용 프로그램에 클래스를 추가합니다. 이러한 클래스를 통해 개체를 조작하고 연결을 관리할 수 있습니다.

형식 라이브러리에서 클래스 정의를 가져오면 응용 프로그램에 지원 파일도 추가합니다. 이러한 파일에는 **pcAnywhere** 자동화 서버의 클래스 정의 및 구현 소스 코드가 들어 있습니다.

14페이지의 “클래스 정의 및 구현 파일 보기” 참조

추가된 클래스를 보려면 다음과 같이 하십시오.

- 1 Visual C++에서 MFC 응용 프로그램을 여십시오.
- 2 [Workspace] 창에서 **[ClassView]** 탭을 누르십시오.

클래스 정의 및 구현 파일 보기

응용 프로그램으로 **pcAnywhere** 자동화 서버 형식 라이브러리를 가져오면 다음 파일이 추가됩니다.

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

이러한 파일에는 **pcAnywhere** 자동화 서버의 클래스 정의 및 구현 소스 코드가 들어 있습니다. 이러한 파일을 편집할 필요는 없지만 **pcAnywhere** 자동화 서버 메소드 호출을 포함하는 각 응용 프로그램 원본 파일에는 **Winawsvr.h**를 포함시켜야 합니다.

클래스 정의 및 구현 파일을 보려면 다음과 같이 하십시오.

- 1 Visual C++에서 MFC 응용 프로그램을 여십시오.
- 2 [Workspace] 창에서 **[FileView]** 탭을 누르십시오.

호스트 및 원격지 OLE 개체 시작

Symantec pcAnywhere에서는 모든 호스트와 원격지 개체를 기본 데이터 디렉터리에 저장해야 합니다. OLE를 사용하여 만든 호스트나 원격지 개체를 실행하기 전에 해당 개체가 pcAnywhere의 기본 데이터 디렉터리에 있는지 확인하십시오. 기본 디렉터리는 Windows XP의 경우 All Users 폴더, Vista의 경우 ProgramData 폴더입니다.

추가 정보 찾기

OLE 자동화에 대한 자세한 정보는 다음을 참조하십시오.

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

Visual Basic 개체 정의

이 장의 내용은 다음과 같습니다.

- Visual Basic 개체 정보
- CRemoteDataManager 메소드
- CRemoteData 속성
- CRemoteDataEx 개체
- CHostDataManager 메소드
- CHostData 속성
- CHostDataEx 개체
- Awrem32 함수

Visual Basic 개체 정보

pcAnywhere 자동화 서버는 OLE 자동화를 지원하기 위해 다음 구성 요소를 제공합니다.

Winawsvr	pcAnywhere 자동화 서버에 연결하고 해당 인터페이스에 액세스하는 데 필요한 정보를 제공합니다.
Awrem32	pcAnywhere 연결을 제어하는 데 필요한 정보를 제공합니다.

표 2-1에서는 Winawsvr을 구성하는 개체를 설명합니다.

표 2-1 Winawsvr 개체

Object	설명	참조 사항
CRemoteDataManager	CRemoteData 개체를 만들고, 열고, 수정하고, 저장하고, 삭제하는 메소드를 제공합니다.	18페이지의 “CRemoteDataManager 메소드” 참조
CRemoteData	pcAnywhere 원격지 함수에 액세스하고 이를 제어하는 매개 변수를 정의합니다.	20페이지의 “CRemoteData 속성” 참조 29페이지의 “CRemoteDataEx 개체” 참조
CHostDataManager	CHostData 개체를 만들고, 열고, 수정하고, 저장하고, 삭제하는 메소드를 제공합니다.	32페이지의 “CHostDataManager 메소드” 참조
CHostData	pcAnywhere 호스트 함수에 액세스하고 이를 제어하는 매개 변수를 정의합니다.	34페이지의 “CHostData 속성” 참조 44페이지의 “CHostDataEx 개체” 참조

Awrem32는 8개의 인터페이스로 구성된 하나의 개체로 원격 제어 및 파일 전송 세션을 지원합니다.

52페이지의 “Awrem32 함수” 참조

일부 함수는 더 이상 지원되지 않습니다. 단, pcAnywhere 이전 버전에 사용할 수 있도록 개체 정의가 제공됩니다.

암호가 필요한 함수의 경우, 암호 값을 설정할 수는 있지만 검색할 수는 없습니다. 이는 보안 유지를 위한 조치입니다.

CRemoteDataManager 메소드

CRemoteDataManager 메소드는 CRemoteData 개체에 액세스하고 이를 제어할 수 있는 매개 변수와 반환 값을 제공합니다.

표 2-2 CRemoteDataManager 메소드

CRemoteDataManager 메소드	매개 변수 또는 반환 값	설명
CurrentDirectory()	반환 값 = String	pcAnywhere의 원격지 개체가 저장되어 있는 현재 디렉터리의 전체 경로 이름을 반환합니다.
ChangeDirectory(NewDirectory)	매개 변수 = NewDirectory	pcAnywhere 원격지 개체가 저장된 현재 디렉터를 변경합니다.

표 2-2 CRemoteDataManager 메소드 (계속)

CRemoteDataManager 메소드	매개 변수 또는 반환 값	설명
FindFirst(Pattern)	매개 변수 = Pattern as string	현재 디렉터리에서 첫 번째 pcAnywhere 원격지 개체 파일(*.chf)을 찾습니다. 파일 이름 패턴 필터는 지정한 파일 이름 패턴을 기반으로 합니다(별표[*]는 현재 디렉터리의 모든 파일을 찾음).
FindFirst(Name string)	매개 변수 = Name as string	지정된 패턴에 일치하는 원격지 개체 파일(*.chf)의 전체 경로 이름에 대해 반환된 버퍼
FindNext(Name)	매개 변수 = Name as string	원래의 FindFirst() 호출에서 지정한 패턴에 부합하는 원격지 개체 파일의 전체 경로 이름에 대한 반환 버퍼
FindNext(Name)	반환 값 = Boolean	FindFirst() 호출에서 지정한 패턴에 부합하는 또다른 원격지 개체 파일을 찾았을 경우 TRUE Name에 일치하는 파일의 전체 경로 이름을 저장합니다.
RetrieveObject(Name)	매개 변수 = Name as string	기준에 완전히 일치하는 볼러올 원격지 개체 파일 이름
RetrieveObject(AccessMode)	매개 변수 = AccessMode as integer	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
RetrieveObject>Password)	매개 변수 = Password as string	개체 암호. NULL일 수 있음
RetrieveObjectEx(Name)	매개 변수 = Name as string	기준에 완전히 일치하는 볼러올 원격지 개체 파일 이름

표 2-2 CRemoteDataManager 메소드 (계속)

CRemoteDataManager 메소드	매개 변수 또는 반환 값	설명
RetrieveObjectEx(AccessMode)	매개 변수 = AccessMode as integer	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
RetrieveObjectEx(Password)	매개 변수 = Password as string	개체 암호. NULL일 수 있음
RetrieveObjectEx	반환 값 = 개체	지정된 파일의 CRemoteDataEx 개체
CreateObject(Name)	매개 변수 = Name as string	CRemoteData 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 원격지 개체 파일 이름
CreateObject	반환 값 = 개체	CRemoteData
CreateObjectEx(Name)	매개 변수 = Name as string	CRemoteDataEx 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 원격지 개체 파일 이름
CreateObjectEx	반환 값 = 개체	CRemoteDataEx
DeleteObject(Name)	매개 변수 = Name as string	기준에 완전히 일치하는 삭제할 개체의 원격지 개체 파일 이름
DeleteObject(Password)	매개 변수 = Password as string	개체 암호
DeleteObject	반환 값 = Boolean	개체를 삭제하면 참

CRemoteData 속성

CRemoteData 개체에 사용할 수 있는 속성 및 매개 변수는 다음과 같습니다.

표 2-3에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-3 CRemoteData 속성 및 매개 변수

속성	매개 변수	설명
<CRemoteData>.ComputerName(String)	String	호스트 시스템의 IP 주소와 시스템 이름을 설정합니다.
<CRemoteData>.PhoneNumber(String)	String	호스트 시스템의 전화 번호를 설정합니다.
<CRemoteData>.UseDialingProperties(Bool)	Bool	시스템 전화 걸기 속성을 설정합니다.
<CRemoteData>.RedialCount(Integer)	Integer	호출을 취소하기 전에 재다이얼을 시도할 수 있는 횟수를 설정합니다.
<CRemoteData>.RedialDelay(Integer)	Integer	재다이얼을 시도하는 사이에 대기할 시간(초)을 설정합니다.
<CRemoteData>.AutoLoginName(String)	String	자동 로그인 사용자 이름을 설정합니다. 도메인 로그인 사용에 대한 자세한 내용은 다음을 참조하십시오. 29페이지의 “CRemoteDataEx 개체” 참조
<CRemoteData>.AutoLoginPassword(String)	String	원격지 개체의 자동 로그인 암호를 설정합니다. 보안상의 이유로 pcAnywhere 자동화 서버는 암호 값을 읽을 수 있는 함수를 제공하지 않습니다. 암호 값은 반환되지 않습니다.
<CRemoteData>.Password(String)	String	ExecuteProtection, ReadProtection, WriteProtection 설정과 함께 사용할 원격지 개체의 암호를 설정합니다. 보안상의 이유로 pcAnywhere 자동화 서버는 암호 값을 읽을 수 있는 함수를 제공하지 않습니다. 암호 값은 반환되지 않습니다.
<CRemoteData>.ExecuteProtection(Bool)	Bool	개체를 실행하는 데 암호가 필요하도록 설정합니다. 암호로 설정하십시오.
<CRemoteData>.ReadProtection(Bool)	Bool	원격지 개체의 속성을 보는 데 암호가 필요하도록 설정합니다. 암호로 설정하십시오.
<CRemoteData>.WriteProtection(Bool)	Bool	원격지 개체에 변경 사항을 저장하는 데 암호가 필요하도록 설정합니다. 암호로 설정하십시오.
<CRemoteData>.LogSession(Bool)	Bool	세션 기록을 활성화/비활성화합니다.
<CRemoteData>.RecordFile(String)	String	활성 세션을 기록하는 파일 위치의 전체 경로와 이름을 설정합니다.
<CRemoteData>.RecordSession(Bool)	Bool	자동 세션 기록을 활성화/비활성화합니다.

표 2-3 CRemoteData 속성 및 매개 변수 (계속)

속성	매개 변수	설명
<CRemoteData>.ReadObject(String)	String	개체의 암호를 설정합니다. 이 속성을 사용하여 원격지 개체의 로컬 데이터 복사본을 새로고침 하십시오.
<CRemoteData>.WriteObject(String)	String	개체의 암호를 설정합니다. 이 속성을 사용하여 원격지 개체를 작성하거나 원격지 개체에 적용한 변경 사항을 기록하십시오.

CRemoteData의 속성 및 반환 값은 다음 표에서 볼 수 있습니다.

표 2-4에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-4 CRemoteData 속성 및 반환 값

속성	반환 값	설명
String = <CRemoteData>.ComputerName	String	호스트 시스템의 IP 주소와 시스템 이름을 반환합니다.
String = <CRemoteData>.PhoneNumber	String	호스트 시스템의 전화 번호를 반환합니다.
Bool = <CRemoteData>.UseDialingProperties	Bool	원격지 개체에 설정된 시스템 전화 걸기 속성을 반환합니다.
Integer = <CRemoteData>.RedialCount	Integer	원격지 개체에 설정된 재다이얼 시도 횟수를 반환합니다.
Integer = <CRemoteData>.RedialDelay	Integer	재다이얼 시도 간격(초)을 반환합니다.
String = <CRemoteData>.AutoLoginName	String	자동 로그인에 사용되는 로그인 이름을 반환합니다.
Bool = <CRemoteData>.ExecuteProtection	Bool	ExecuteProtection 설정 값을 반환합니다.
Bool = <CRemoteData>.ReadProtection	Bool	ReadProtection 설정 값을 반환합니다.
Bool = <CRemoteData>.WriteProtection	Bool	WriteProtection 설정 값을 반환합니다.
Bool = <CRemoteData>.LogSession	Bool	세션 기록이 실행되어 있으면 참을 반환합니다.
String = <CRemoteData>.RecordFile	String	세션 기록 파일의 전체 경로와 이름을 반환합니다.
Bool = <CRemoteData>.RecordSession	Bool	세션 기록 설정 값을 반환합니다.

연결 유형 속성

다음 표에서는 연결 유형 속성 및 매개 변수를 정의합니다.

[표 2-5](#)에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-5 연결 유형 속성 및 매개 변수

속성	매개 변수	설명
<CRemoteData>.ConnectionType(String)	String	원격지의 연결 유형을 설정합니다. 전송되는 값은 FirstConnectionType()과 NextConnectionType() 함수에서 정의된 올바른 연결 유형이어야 합니다. 다음은 유효한 연결 유형 예제입니다. ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ CAPI 2.0을 사용하는 ISDN ■ 모뎀 이름(시스템에 표시된 이름)

다음 표에서는 연결 유형 속성 및 반환 값을 정의합니다.

[표 2-6](#)에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-6 연결 유형 속성 및 반환 값

속성	반환 값	설명
String = <CRemoteData>.ConnectionTypes	String	원격지 개체의 연결 유형을 반환합니다.
Integer = <CRemoteData>.ConnectionType	Integer	사용할 수 있는 연결 유형 수를 반환합니다. 다음은 유효한 연결 유형 예제입니다. ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ CAPI 2.0을 사용하는 ISDN ■ 모뎀 이름(시스템에 표시된 이름)

표 2-6 연결 유형 속성 및 반환 값 (계속)

속성	반환 값	설명
String = <CRemoteData>.FirstConnectionType	String	처음으로 사용할 수 있는 연결 유형을 반환합니다.
String = <CRemoteData>.NextConnectionType	String	다음으로 사용할 수 있는 연결 유형을 반환합니다. <CRemoteData>.ConnectionType에 설정된 연결 유형 번호에 따라 순서대로 호출하여 모든 연결 유형을 차례로 열거합니다.
Bool = <CRemoteData>.FindConnectionType(ConnectionType)	Bool	이름이 지정된 연결 유형이 사용 가능한 연결 유형 목록에 있으면 참을 반환합니다.

전화 걸기 등록 정보

다음 표에서는 모뎀 연결을 위해 전화 걸기 속성을 설정하는 속성과 매개 변수를 정의합니다.

표 2-7에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-7 전화 걸기 속성의 속성 및 매개 변수

속성	매개 변수	설명
<CRemoteData>.AreaCode(String)	String	모뎀 연결의 지역 코드 전화 걸기 속성을 설정합니다.
<CRemoteData>.CountryCode(String)	String	모뎀 연결의 국가 코드 전화 걸기 속성을 설정합니다.

다음 표에서는 모뎀 전화 걸기 속성의 속성과 반환 값을 정의합니다.

표 2-8에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-8 전화 걸기 등록 정보의 속성 및 반환 값

속성	반환 값	설명
String = <CRemoteData>.AreaCode	String	지역 코드 전화 걸기 속성을 반환합니다.
String = <CRemoteData>.CountryCode	String	전화 걸기 속성 국가 코드를 반환합니다.
Integer = <CRemoteData>.CountryCodes	Integer	사용할 수 있는 국가 코드 번호를 반환합니다.
String = <CRemoteData>.FirstCountryCode	String	운영 체제에 나열된 목록 중에서 처음으로 사용할 수 있는 국가 코드를 반환합니다.

표 2-8 전화 걸기 등록 정보의 속성 및 반환 값 (계속)

속성	반환 값	설명
String = <CRemoteData>.NextCountryCode	String	다음으로 사용할 수 있는 국가 코드를 반환합니다. <CRemoteData.CountryCodes>에 설정된 국가코드 번호에 따라 순서대로 호출하여 모든 국가 코드를 차례로 열거합니다.

COM 장치 등록 정보

다음 표에서는 모뎀과 다른 COM 기반 연결에 대한 포트 설정을 사용자 정의할 수 있는 속성들과 매개 변수를 설명합니다.

표 2-9에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-9 COM 장치 속성 및 매개 변수

속성	매개 변수	설명
<CRemoteData>.ComParity(String)	String	통신 패리티를 설정합니다. 올바른 값은 다음과 같습니다. ■ <없음> ■ 홀수 ■ 짝수 ■ 기호 ■ 공백
<CRemoteData>.ComFlowControl(String)	String	COM 기반 연결 유형의 흐름 제어를 설정합니다. 올바른 값은 다음과 같습니다. ■ <없음> ■ XONXOFF ■ RTS/CTS ■ 모두

표 2-9 COM 장치 속성 및 매개 변수 (계속)

속성	매개 변수	설명
<CRemoteData>.ComStartedBy(String)	String	COM 기반 연결 유형의 시작 설정을 구성합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI) ■ 2 <CR> 수신 ■ 모뎀 응답
<CRemoteData>.ComEndedBy(String)	String	COM 기반 연결 유형의 종료 설정을 구성합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI)
<CRemoteData>.ComSpeed(Long)	Long	최대 COM 속도 설정을 포함합니다. 올바른 값은 다음과 같습니다. ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

다음 표에서는 COM 장치 속성 및 반환 값을 설명합니다.

표 2-10에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-10 COM 장치 속성 및 반환 값

속성	반환 값	설명
String = <CRemoteData>.ComParity	String	다음 통신 패리티 값 중 하나를 반환합니다. ■ <없음> ■ 홀수 ■ 짝수 ■ 기호 ■ 공백
String = <CRemoteData>.ComFlowControl	String	원격지 개체의 Com 흐름 설정을 반환합니다. 올바른 값은 다음과 같습니다. ■ <없음> ■ XONXOFF ■ RTS/CTS ■ 모두
String = <CRemoteData>.ComStartedBy	String	COM 기반 연결 유형의 Com 시작 제어를 반환합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI) ■ 2 <CR> 수신 ■ 모뎀 응답
String = <CRemoteData>.ComEndedBy	String	COM 기반 연결 유형의 Com 종료 제어를 반환합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI)

표 2-10COM 장치 속성 및 반환 값 (계속)

속성	반환 값	설명
Long = <CRemoteData>.ComSpeed	Long	원격지 개체의 현재 Com 속도 설정을 반환합니다. 올바른 값은 다음과 같습니다. ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

NetBIOS 장치 속성

다음 표에서는 NetBIOS 네트워크 장치의 속성과 매개 변수를 정의합니다.

표 2-11에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-11NetBIOS 속성 및 매개 변수

속성	매개 변수	설명
<CRemoteData>.LanaNumber(Integer)	Integer	NetBIOS 연결의 LANA(LAN 어댑터)번호를 설정합니다.

다음 표에서는 NetBIOS 속성 및 반환 값을 정의합니다.

표 2-12에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-12NetBIOS 속성 및 반환 값

속성	반환 값	설명
Integer = <CRemoteData>.LanaNumber	Integer	NetBIOS 연결의 현재 LANA(LAN 어댑터)번호 설정을 반환합니다.

ISDN via CAPI 2.0 장치 등록 정보

다음 표에서는 유럽 ISDN 연결의 속성과 매개 변수를 정의합니다.

표 2-13에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-13 CAPI 2.0을 사용하는 ISDN 속성 및 매개 변수

속성	매개 변수	설명
<CRemoteData>.CapiChannelBonding(Bool)	Bool	ISDN CAPI 장치의 채널 연결을 활성화/비활성화합니다.
<CRemoteData>.CapiExtensions(String)	String	통신에 필요한 추가 CAPI 내선 번호를 설정합니다.

다음 표에서는 유럽 ISDN 연결의 속성과 반환 값을 정의합니다.

표 2-14에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-14 CAPI 2.0을 사용하는 ISDN 속성 및 반환 값

속성	반환 값	설명
Bool= <CRemoteData>.CapiChannelBonding	Bool	원격지 개체에서 현재 ISDN CAPI 채널 연결 설정을 반환합니다.
String = <CRemoteData>.CapiExtensions	String	원격지 개체의 현재 CAPI 내선 번호 목록을 반환합니다.

CRemoteDataEx 개체

CRemoteDataEx 개체는 CRemoteData 개체와 동일한 기능을 가지며 여기에 일부 기능이 추가되어 있습니다.

다음 표에서는 속성과 매개 변수를 설명합니다.

표 2-15에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-15 CRemoteDataEx 매개 변수

속성	매개 변수	설명
<CRemoteData>.PrivateKey(String)	String	사용할 개인 키 컨테이너 이름을 설정합니다.
<CRemoteData>.CertificateName(String)	String	사용할 개인 키의 공통 이름을 설정합니다.
<CRemoteData>.EncryptionLevel(Byte)	Byte	암호화 수준을 설정합니다. 올바른 값은 다음과 같습니다. ■ -1: 없음 ■ 0: pcAnywhere ■ 1: 대칭 ■ 2: 공용 키

표 2-15CRemoteDataEx 매개 변수 (계속)

속성	매개 변수	설명
<CRemoteData>.DenyLowerEncrypt(Bool)	Bool	원격지 시스템이 낮은 수준의 암호화를 사용하는 호스트 시스템에 연결하는 것을 허용할지 정의합니다.
<CRemoteData>.AutoDomain(String)	String	자동 로그인용 도메인 이름을 설정합니다. 이 옵션은 NT 인증 및 Windows 인증과 함께 사용됩니다.

다음 표에서는 CRemoteDataEx 개체의 속성과 반환 값을 설명합니다.

표 2-16에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-16CRemoteDataEx 속성 및 반환 값

속성	반환 값	설명
String = <CRemoteData>.PrivateKey	String	현재 실행 중인 개인 키 컨테이너 이름을 반환합니다.
String = <CRemoteData>.CertificateName	String	실행 중인 개인 키 컨테이너의 공통 이름을 반환합니다.
Byte = <CRemoteData>.EncryptionLevel	Byte	다음 암호화 설정 중 하나를 반환합니다. ■ -1: 없음 ■ 0: pcAnywhere ■ 1: 대칭 ■ 2: 공용 키
Bool = <CRemoteData>.DenyLowerEncrypt	Bool	낮은 암호화 설정 거부 값을 반환합니다.
String = <CRemoteData>.AutoDomain	String	자동 로그인용 도메인 이름 설정을 반환합니다. 이 옵션은 NT 인증 및 Windows 인증과 함께 사용됩니다.

원격지 기능의 Visual Basic 샘플 코드

다음 Visual Basic 샘플 코드는 원격지 데이터 개체를 검색하고 해당 속성을 수정합니다.

```
Private Sub Command1_Click()  
Dim RemoteDataManager as Object  
Dim RemoteData as Object  
Dim s as string
```

```
'Create CRemoteDataManager object
Set RemoteDataManager = CreateObject(WINAWSVR.REMOTEDATAMANAGER)
```

```
'display and change current directory
s = RemoteDataManager.CurrentDirectory()
MsgBox ( s )
RemoteDataManager.ChangeDirectory ( "C:\dev\bin.w32\data")
s = RemoteDataManager.CurrentDirectory()
MsgBox ( s )
```

```
'retrieve remote data object
Set RemoteData = RemoteDataManager.RetrieveObjectEx("pod.CHF",
2, 0)
```

```
'display some properties
s = RemoteData.AreaCode()
MsgBox (s)
s = RemoteData.PhoneNumber()
MsgBox (s)
```

```
'set some properties
RemoteData.AreaCode = "212"
RemoteData.PhoneNumber = "555-5555"
```

```
'write object to disk
RemoteData.WriteObject (0)
End Sub
```

FindFirst와 **FindNext** 메소드를 사용하여 다음과 같이 디렉터리에 원격지 파일을 표시하십시오.

```
Private Sub Command5_Click()

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

```

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

```

원격지 개체를 작성하십시오. 다음과 같이 TCP/IP에 대한 연결 유형과 "Host1"이라는 시스템 이름을 설정한 후 시작하십시오.

```

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

Set RemoteDataManager =
CreateObject("WINAHSV.R. REMOTEDATAMANAGER")
MsgBox (RemoteDataManager.CurrentDirectory())

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

```

CHostDataManager 메소드

CHostDataManager 메소드는 ChostData 개체에 액세스하고 이를 제어할 수 있는 매개 변수와 반환 값을 제공합니다.

CHostDataManager 메소드	매개 변수 또는 반환 값	설명
CurrentDirectory()	반환 값 = String	pcAnywhere 호스트 개체를 저장한 현재 pcAnywhere 데이터 디렉터리의 전체 경로 이름
FindFirst(Pattern)	매개 변수 = Pattern as string	개체 파일을 필터링하는 파일 이름 패턴(별표[*]는 현재 디렉터리에 있는 모든 파일을 찾음)
FindFirst(Name string)	매개 변수 = Name as string	지정된 패턴에 일치하는 호스트 개체 파일의 전체 경로 이름에 대해 반환된 버퍼
FindFirst	반환 값 = Boolean	지정된 패턴에 일치하는 호스트 개체 파일을 찾으면 참.Name에 일치하는 파일의 전체 경로 이름 저장.
FindNext(Name)	매개 변수 = Name as string	FindFirst()의 원래 호출에 지정된 패턴에 일치하는 호스트 개체 파일의 전체 경로 이름에 대해 반환된 버퍼
FindNext	반환 값 = Boolean	FindFirst()의 호출에 지정된 패턴에 일치하는 다른 호스트 개체 파일을 찾으면 참.Name에 일치하는 파일의 전체 경로 이름을 저장합니다.
RetrieveObject(Name)	매개 변수 = Name as string	로드할 전체 호스트 개체 파일 이름
RetrieveObject(AccessMode)	매개 변수 = AccessMode as integer	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
RetrieveObject(Password)	매개 변수 = Password as string	개체 암호.NULL일 수 있음
RetrieveObject	반환 값 = Object	지정된 파일의 CHostData 개체
RetrieveObjectEx(Name)	매개 변수 = Name as string	로드할 전체 호스트 개체 파일 이름

CHostDataManager 메소드	매개 변수 또는 반환 값	설명
RetrieveObjectEx(AccessMode)	매개 변수 = AccessMode as integer	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
RetrieveObjectEx(Password)	매개 변수 = Password as string	개체 암호. NULL일 수 있음
RetrieveObjectEx	반환 값 = Object	지정된 파일의 CHostDataEx 개체
CreateObject(Name)	매개 변수 = Name as string	CHostData 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 호스트 개체 파일 이름입니다.
CreateObject	반환 값 = 개체	CHostData
CreateObjectEx(Name)	매개 변수 = Name as string	CHostDataEx 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 호스트 개체 파일 이름입니다.
CreateObjectEx	반환 값 = 개체	CHostDataEx
DeleteObject(Name)	매개 변수 = Name as string	삭제할 개체의 전체 호스트 개체 파일 이름
DeleteObject(Password)	매개 변수 = Password as string	개체 암호
DeleteObject	반환 값 = Boolean	개체를 삭제하면 참
Launch(Name)	매개 변수 = Name as string	pcAnywhere 호스트 터미널 창을 여는 호스트 개체 파일을 시작합니다. 매개 변수는 시작할 개체의 전체 호스트 개체 파일입니다.
Launch	반환 값 = Boolean	개체를 성공적으로 시작하면 참

CHostData 속성

다음 표에서는 CHostData 개체에 사용할 수 있는 속성들과 매개 변수를 설명합니다.

표 2-17에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-17 CHostData 속성 및 매개 변수

속성	매개 변수	설명
<CHostData>.PhoneNumber(String)	String	호스트 시스템의 전화 번호를 설정합니다.
<CHostData>.UseDialingProperties(Bool)	Bool	시스템 전화 걸기 속성을 설정합니다.
<CHostData>.RedialCount(Integer)	Integer	호출을 취소하기 전에 재다이얼을 시도할 수 있는 횟수를 설정합니다.
<CHostData>.RedialDelay(Integer)	Integer	재다이얼을 시도하는 사이에 대기할 시간(초)을 설정합니다.
<CHostData>.Password(String)	String	ExecuteProtection, ReadProtection, WriteProtection 설정과 함께 사용할 호스트 개체의 암호를 설정합니다. 보안상의 이유로 pcAnywhere 자동화 서버는 암호 값을 읽을 수 있는 함수를 제공하지 않습니다. 암호 값은 반환되지 않습니다.
<CHostData>.ExecuteProtection(Bool)	Bool	개체를 실행하는 데 암호가 필요하도록 설정합니다. 암호로 설정하십시오.
<CHostData>.ReadProtection(Bool)	Bool	호스트 개체의 속성을 보는 데 암호가 필요하도록 설정합니다. 암호로 설정하십시오.
<CHostData>.WriteProtection(Bool)	Bool	호스트 개체에 변경 사항을 저장하는 데 암호가 필요하도록 설정합니다. 암호로 설정하십시오.
<CHostData>.LogSession(Bool)	Bool	세션 기록을 활성화/비활성화합니다.
<CHostData>.RecordFile(String)	String	활성 세션을 기록하는 파일 위치의 전체 경로와 이름을 설정합니다.
<CHostData>.RecordSession(Bool)	Bool	자동 세션 기록을 활성화/비활성화합니다.
<CHostData>.ReadObject(String)	String	개체의 암호를 설정합니다. 이 속성을 사용하여 호스트 개체의 로컬 데이터 복사본을 새로고침 할 수 있습니다.
<CHostData>.WriteObject(String)	String	개체의 암호를 설정합니다. 이 속성을 사용하여 호스트 개체를 작성하거나 호스트 개체에 적용한 변경 사항을 기록하십시오.

다음 표에서는 CHostData 속성들과 반환 값을 설명합니다.

[표 2-18](#)에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-18 CHostData 속성 및 반환 값

속성	반환 값	설명
String = <CHostData>.PhoneNumber	String	호스트 시스템의 전화 번호를 반환합니다.
Bool = <CHostData>.UseDialingProperties	Bool	호스트 개체에 설정된 시스템 전화 걸기 속성의 사용을 반환합니다.
Integer = <CHostData>.RedialCount	Integer	호스트 개체에 설정된 재다이얼 시도 횟수를 반환합니다.
Integer = <CHostData>.RedialDelay	Integer	재다이얼 시도 간격(초)을 반환합니다.
Bool = <CHostData>.ExecuteProtection	Bool	ExecuteProtection 설정 값을 반환합니다.
Bool = <CHostData>.ReadProtection	Bool	ReadProtection 설정 값을 반환합니다.
Bool = <CHostData>.WriteProtection	Bool	Write Protection 설정 값을 반환합니다.
Bool = <CHostData>.LogSession	Bool	세션 기록이 실행되어 있으면 참을 반환합니다.
String = <CHostData>.RecordFile	String	세션 기록 파일의 전체 경로와 이름을 반환합니다.
Bool = <CHostData>.RecordSession	Bool	세션 기록 설정 값을 반환합니다.

연결 유형 속성

다음 표에서는 연결 유형 속성 및 매개 변수를 설명합니다.

표 2-19에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-19 연결 유형 속성 및 매개 변수

속성	매개 변수	설명
<CHostData>.ConnectionType(String)	String	호스트의 연결 유형을 설정합니다. 전송되는 값은 FirstConnectionType()과 NextConnectionType() 함수에서 정의된 올바른 연결 유형이어야 합니다. 다음은 유효한 연결 유형 예제입니다. ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ CAPI 2.0을 사용하는 ISDN ■ 모뎀 이름(시스템에 표시된 이름)
<CHostData>.AssignConnection(String)	String	문자열에 정의된 연결 유형을 활성 상태로 설정합니다.
<CHostData>.UnassignConnection(String)	String	문자열에 정의된 연결 유형을 비활성 상태로 설정합니다.

다음 표에서는 연결 유형 속성 및 반환 값을 설명합니다.

표 2-20에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-20 연결 유형 속성 및 반환 값

속성	반환 값	설명
String = <CHostData>.ConnectionTypes	String	호스트 개체의 연결 유형을 반환합니다.

표 2-20 연결 유형 속성 및 반환 값 (계속)

속성	반환 값	설명
Integer = <CHostData>.ConnectionType	Integer	사용할 수 있는 연결 유형 수를 반환합니다. 다음은 유효한 연결 유형 예제입니다. ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ CAPI 2.0을 사용하는 ISDN ■ 모뎀 이름(시스템에 표시된 이름)
String = <CHostData>.FirstConnectionType	String	처음으로 사용할 수 있는 연결 유형을 반환합니다.
String = <CHostData>.NextConnectionType	String	다음으로 사용할 수 있는 연결 유형을 반환합니다. <CHostData>.ConnectionTypes에 설정된 연결 유형 번호에 따라 순서대로 호출하여 모든 연결 유형을 차례로 열거합니다.
Bool = <CHostData>.FindConnectionType(ConnectionType)	Bool	이름이 지정된 연결 유형이 사용 가능한 연결 유형 목록에 있으면 참을 반환합니다.
Integer = <CHostData>.MaxAssignedConnections	Integer	현재 호스트에서 활성화할 수 있는 최대 연결 유형 수를 반환합니다.
String = <CHostData>.FirstAssignedConnection	String	첫 번째로 할당된 활성 연결 유형을 반환합니다.
String = <CHostData>.NextAssignedConnection	String	다음으로 할당된 활성 연결 유형을 반환합니다.
Bool = <CHostData>.FindAssignedConnection(ConnectionType)	Bool	전송되는 연결 유형이 활성 연결 유형과 일치하면 참을 반환합니다.

AssignConnection(ConnectionType) 메소드

AssignConnection(ConnectionType) 방법은 연속적인 장치 지정 방법 호출을 처리할 때, 할당된 연결 유형의 호스트 개체 목록에 요청된 연결 유형을 배치하고, 이를 현재 연결 유형으로 설정합니다.

요청된 연결 유형이 할당된 연결 목록에 이미 있으면 할당된 연결 목록이 변경되지 않습니다. 현재 연결 유형이 요청된 유형으로 변경되지만 합니다. 연결 지정 값을 불러오고 설정하는 과정에서 여러 번 동일한 개체에서 AssignConnection 메소드를 호출하는 것은 일관적입니다.

AssignConnection은 전송된 연결 유형이 시스템에 있고, 이미 할당되었거나, 할당할 수 있는 경우, 참을 반환합니다. 요청된 연결 유형이 시스템에 없거나, 현재 할당된 연결 수가 허용된 최대 수준에 이미 도달한 경우, 거짓을 반환합니다.

pcAnywhere 호스트 개체는 할당된 연결 유형을 두 개까지 지원할 수 있습니다. 이러한 시도가 제한 값을 초과한 경우, **AssignConnection** 메소드는 거짓을 반환합니다.

다음 표에서는 **AssignConnection** 매개 변수를 정의합니다.

매개 변수	설명
ConnectionType as string	할당할 연결 장치 유형 이름

다음 표에서는 **AssignConnection** 반환 값을 정의합니다.

반환 값	설명
Boolean	이 장치 유형을 사용할 수 있고 할당된 최대 허용 연결 수에 아직 도달하지 않은 경우, 참을 반환합니다.

UnassignConnection(ConnectionType) 메소드

전송된 연결 유형이 할당된 연결 유형 목록에서 성공적으로 제거되면 **UnassignConnection(ConnectionType)** 메소드는 참을 반환합니다.

다음 표에서는 **UnassignConnection** 매개 변수를 정의합니다.

매개 변수	설명
ConnectionType as string	할당을 제거할 연결 장치 유형 이름

다음 표에서는 **UnassignConnection** 반환 값을 정의합니다.

반환 값	설명
Boolean	이 장치 유형의 할당을 제거했으면 참

전화 걸기 등록 정보

다음 표에서는 모뎀 연결을 위해 전화 걸기 속성을 설정하는 속성과 반환 값을 정의합니다.

[표 2-21](#)에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-21모뎀 전화 걸기 속성의 속성 및 매개 변수

속성	매개 변수	설명
<CHostData>.AreaCode(String)	String	모뎀 연결의 지역 코드를 설정합니다.
<CHostData>.CountryCode(String)	String	모뎀 연결의 국가 코드를 설정합니다.

다음 표에서는 모뎀 전화 걸기 속성의 속성과 반환 값을 정의합니다.
표 2-22에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-22모뎀 전화 걸기 속성의 속성 및 반환 값

속성	반환 값	설명
String = <CHostData>.AreaCode	String	지역 코드를 반환합니다.
String = <CHostData>.CountryCode	String	국가 코드를 반환합니다.
Integer = <CHostData>.CountryCodes	Integer	사용할 수 있는 국가 코드 번호를 반환합니다.
String= <CHostData>.FirstCountryCode	String	운영 체제에 나열된 목록 중에서 처음으로 사용할 수 있는 국가 코드를 반환합니다.
String = <CHostData>.NextCountryCode	String	다음으로 사용할 수 있는 국가 코드를 반환합니다. <CHostData>.CountryCodes에 설정된 국가 코드 번호에 따라 순서대로 호출하여 모든 국가 코드를 차례로 열거합니다.

COM 장치 등록 정보

다음 표에서는 모뎀과 다른 COM 기반 연결에 대한 포트 설정을 사용자 정의할 수 있는 속성들과 매개 변수를 정의합니다.

표 2-23에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-23COM 장치 속성 및 매개 변수

속성	매개 변수	설명
<CHostData>.ComParity(String)	String	통신 패리티를 설정합니다. 올바른 값은 다음과 같습니다. ■ <없음> ■ 홀수 ■ 짝수 ■ 기호 ■ 공백

표 2-23 COM 장치 속성 및 매개 변수 (계속)

속성	매개 변수	설명
<CHostData>.ComFlowControl(String)	String	COM 기반 연결 유형의 흐름 제어를 설정합니다. 올바른 값은 다음과 같습니다. ■ <없음> ■ XONXOFF ■ RTS/CTS ■ 모두
<CHostData>.ComStartedBy(String)	String	COM 기반 연결 유형의 시작 설정을 구성합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI) ■ 2 <CR> 수신 ■ 모뎀 응답
<CHostData>.ComEndedBy(String)	String	COM 기반 연결 유형의 종료 설정을 구성합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI)
<CHostData>.ComSpeed(Long)	Long	최대 COM 속도 설정을 구성합니다. 올바른 값은 다음과 같습니다. ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

다음 표에서는 COM 장치 속성 및 반환 값을 설명합니다.

표 2-24에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-24 COM 장치 속성 및 반환 값

속성	반환 값	설명
String = <CHostData>.ComParity	String	다음 값 중 하나를 통신 패리티로 반환합니다. ■ <없음> ■ 홀수 ■ 짝수 ■ 기호 ■ 공백
String = <CHostData>.ComFlowControl	String	호스트 개체의 Com 흐름 설정을 반환합니다. 올바른 값은 다음과 같습니다. ■ <없음> ■ XONXOFF ■ RTS/CTS ■ 모두
String = <CHostData>.ComStartedBy	String	COM 기반 연결 유형의 Com 시작 제어를 반환합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI) ■ 2 <CR> 수신 ■ 모뎀 응답
String = <CHostData>.ComEndedBy	String	COM 기반 연결 유형의 Com 종료 제어를 반환합니다. 올바른 값은 다음과 같습니다. ■ 항상 연결됨 ■ 반송파 탐지 (DCD) ■ 전송 클리어 (CTS) ■ 데이터 세트 준비 (DSR) ■ 링 표시기 (RI)

표 2-24 COM 장치 속성 및 반환 값 (계속)

속성	반환 값	설명
Long = <CHostData>.ComSpeed	Long	호스트 개체의 현재 Com 속도 설정을 반환합니다. 올바른 값은 다음과 같습니다. ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

NetBIOS 장치 속성

다음 표에서는 NetBIOS 네트워크 장치의 속성과 매개 변수를 설명합니다.

[표 2-25](#)에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-25 NetBIOS 속성 및 매개 변수

속성	매개 변수	설명
<CHostData>.LanaNumber(Integer)	Integer	NetBIOS 연결의 LANA(LAN 어댑터) 번호를 설정합니다.

다음 표에서는 NetBIOS 네트워크 장치의 속성과 반환 값을 정의합니다.

[표 2-26](#)에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-26 NetBIOS 반환 값

속성	반환 값	설명
Integer = <CHostData>.LanaNumber	Integer	NetBIOS 연결의 현재 LANA(LAN 어댑터) 번호 설정을 반환합니다.

ISDN via CAPI 2.0 장치 등록 정보

다음 표에서는 유럽 ISDN 연결의 속성과 매개 변수를 정의합니다.

[표 2-27](#)에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-27 CAPI 2.0을 사용하는 ISDN 속성 및 매개 변수

속성	매개 변수	설명
<CHostData>.CapiChannelBonding(Bool)	Bool	ISDN CAPI 장치의 채널 연결을 활성화/비활성화합니다.
<CHostData>.CapiExtensions(String)	String	통신에 필요한 추가 CAPI 내선 번호를 설정합니다.

표 2-28에서는 유럽 ISDN 연결의 속성과 반환 값을 정의합니다.

표 2-28 CAPI 2.0을 사용하는 ISDN 속성 및 반환 값

속성	반환 값	설명
Bool= <CHostData>.CapiChannelBonding	Bool	호스트 개체의 현재 ISDN CAPI 채널 연결 설정을 반환합니다.
String = <CHostData>.CapiExtensions	String	호스트 개체의 현재 CAPI 내선 번호 목록을 반환합니다.

CHostDataEx 개체

CHostDataEx 개체는 CHostData 개체와 동일한 기능을 가지며 일부 기능이 추가되어 있습니다.

다음 표에서는 속성과 매개 변수를 설명합니다.

표 2-29에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-29 CHostDataEx 속성 및 매개 변수

속성	매개 변수	설명
<CHostData>.CryptPrivateKey(String)	String	사용할 개인 키 컨테이너 이름을 설정합니다.
<CHostData>.CryptCommonName(String)	String	사용할 개인 키의 공통 이름을 설정합니다.
<CHostData>.CryptReqLevel(Byte)	Byte	암호화 수준을 설정합니다. 올바른 값은 다음과 같습니다. ■ -1: 없음 ■ 0: pcAnywhere ■ 1: 대칭 ■ 2: 공용 키

표 2-29 CHostDataEx 속성 및 매개 변수 (계속)

속성	매개 변수	설명
<CHostData>.CryptRefuseLower(Bool)	Bool	호스트 시스템이 낮은 수준의 암호화를 사용하는 원격지 시스템에서 수신되는 연결을 허용할지 정의합니다.
<CHostData>.CallersPath(String)	String	호출자 파일의 전체 경로를 설정합니다.
<CHostData>.ConfirmConnect(Bool)	Bool	호스트 사용자에게 연결 확인 메시지를 표시할지 정의합니다.
<CHostData>.ConfirmTimeout(Byte)	Byte	연결 확인 메시지를 표시하는 시간 제한이 만료될 시간(초)을 설정합니다.
<CHostData>.ConfirmDeny(Bool)	Bool	연결 확인 메시지를 표시하는 시간 제한이 만료된 경우 연결을 종료할지 정의합니다.
<CHostData>.PwCaseSensitive(Bool)	Bool	대소문자를 구분하는 암호를 사용하도록 설정합니다.
<CHostData>.PwAttempts(Byte)	Byte	연결을 종료하기 전에 연속 로그인 시도 실패를 몇 번까지 허용할지 설정합니다.
<CHostData>.PwTimeout(Byte)	Byte	연결이 끊어지기 전에 사용자가 로그인을 완료해야 할 시간(분)을 설정합니다.
<CHostData>.ActiveKbds(Byte)	Byte	연결되어 있는 동안 활성화할 마우스와 키보드를 정의합니다. 올바른 값은 다음과 같습니다. ■ 0: 호스트 및 원격지 ■ 1: 호스트 ■ 2: 원격지
<CHostData>.InactiveTimeout(Byte)	Byte	연결이 끊어지기 전에 키보드와 마우스를 유휴 상태로 둘 수 있는 시간(분)을 설정합니다.
<CHostData>.LockSystemWhileWait(Bool)	Bool	시작할 때 호스트 시스템 잠금 설정을 구성합니다.
<CHostData>.MinimizeOnLaunch(Bool)	Bool	아이콘 형태로 호스트 실행 시작 옵션을 설정합니다.
<CHostData>.RunAsService(Bool)	Bool	서비스로 실행할 호스트를 활성화합니다.
<CHostData>.ConnLostWait(Byte)	Byte	다른 호출자에게 연결을 허용하기 전에 대기할 시간(분)을 설정합니다.

표 2-29 CHostDataEx 속성 및 매개 변수 (계속)

속성	매개 변수	설명
<CHostData>.ConnLostHostOpts(Bool)	Bool	세션이 비정상적으로 종료된 경우 다른 연결에 대기할지 아니면 호스트를 취소할지 정의합니다. 올바른 값은 다음과 같습니다. ■ FALSE(거짓): 대기 ■ TRUE(참): 호스트 취소
<CHostData>.EnableConnLostSecurity(Bool)	Bool	비정상적으로 종료된 세션 종료 보안 옵션을 활성화/비활성화합니다.
<CHostData>.AuthenticationType(Byte)	Byte	인증 유형을 설정합니다. 올바른 값은 다음과 같습니다. ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS (Active Directory 서비스) ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID 인증 값을 설정하지 않거나 설정 값이 올바르지 않으면(예, 사용할 수 없는 인증 유형) 기본값으로 pcAnywhere 인증이 사용됩니다.
<CHostData>.ConnLostSecurity(Byte)	Byte	비정상적인 세션 종료를 처리할 보안 옵션을 설정합니다. 올바른 값은 다음과 같습니다. ■ 1: 사용자 로그오프 ■ 2: 호스트 시스템 재시작 ■ 3: 시스템 잠금
<CHostData>.CallbkDelay(Byte)	Byte	호스트 모뎀이 다시 원격지를 호출하기 전에 대기할 시간(초)을 설정합니다.

표 2-29 CHostDataEx 속성 및 매개 변수 (계속)

속성	매개 변수	설명
<CHostData>.EndSessHostOpts(Bool)	Bool	세션이 비정상적으로 종료된 후 호스트를 다른 연결에 대기시킬지 아니면 취소할지 정의합니다. 올바른 값은 다음과 같습니다. ■ FALSE(거짓): 다음 연결에 대기 ■ TRUE(참): 호스트 취소
<CHostData>.EnableEndSessSecurity(Bool)	Bool	비정상적인 세션 종료에 대한 보안 옵션을 활성화/비활성화합니다.
<CHostData>.EndSessSecurity(Byte)	Byte	정상적인 세션 종료에 대한 보안 옵션을 설정합니다. 올바른 값은 다음과 같습니다. ■ 1: 사용자 로그오프 ■ 2: 호스트 시스템 재시작 ■ 3: 시스템 잠금
<CHostData>.BlankHost(Bool)	Bool	호스트 시스템에서 화면 숨김을 설정합니다. 이 옵션을 지원하지 않는 비디오 카드도 있습니다.
<CHostData>.AllowRemoteMouse(Bool)	Bool	세션을 수행하는 동안 원격지 시스템의 마우스 사용을 중지합니다.
<CHostData>.RebootOnDisconnect(Bool)	Bool	참이면 세션 종료 이후 시스템을 다시 시작하도록 설정합니다.
<CHostData>.PasswordAfterDisc(Bool)	Bool	세션 종료 후 사용자를 로그오프합니다.
<CHostData>.LogFailures(Bool)	Bool	암호 시도 실패를 기록합니다.
<CHostData>.AllowDriveSecurity(Bool)	Bool	드라이브 보안 옵션을 실행합니다. 이 설정은 NTFS 파일 시스템에서만 유효합니다.
<CHostData>.UseDirectoryServices(Bool)	Bool	인증에 대한 디렉터리 서비스 사용을 실행합니다.
<CHostData>.DirectoryServiceEntry(String)	String	디렉터리 서비스 설정을 구성합니다.

다음 표에서는 CHostDataEx 개체의 속성과 반환 값을 설명합니다.

표 2-30에서 꺾쇠 괄호 안에 있는 정보는 실제 값으로 대체하십시오.

표 2-30 CHostDataEx 속성 및 반환 값

속성	반환 값	설명
String = <CHostDataEx>.CryptPrivateKey	String	현재 실행 중인 개인 키 컨테이너를 반환합니다.
String = <CHostDataEx>.CryptCommonName	String	실행 중인 개인 키 컨테이너의 공통 이름을 반환합니다.
Byte = <CHostDataEx>.CryptReqLevel	Byte	다음 암호화 설정 중 하나를 반환합니다. ■ -1: 없음 ■ 0: pcAnywhere ■ 1: 대칭 ■ 2: 공용 키
Bool = <CHostDataEx>.CryptRefuseLower	Bool	낮은 암호화 설정 거부 값을 반환합니다.
String = <CHostDataEx>.CallersPath	String	현재 실행 중인 호출자의 파일 경로를 반환합니다.
Bool = <CHostDataEx>.ConfirmConnect	Bool	연결 확인 설정 값을 반환합니다.
Byte = <CHostDataEx>.ConfirmTimeout	Byte	낮은 암호화 설정 거부 값을 반환합니다.
Bool = <CHostDataEx>.ConfirmDeny	Bool	시간 만료 후 연결 끊기 설정 값을 반환합니다.
Bool = <CHostDataEx>.PwCaseSensitive	Bool	대소문자 구별 암호 사용 설정 값을 반환합니다.
Byte = <CHostDataEx>.PwAttempts	Byte	호출 당 로그인 시도 횟수 제한 설정 값을 반환합니다.
Byte = <CHostDataEx>.PwTimeout	Byte	로그인 완료 시간 제한(분 단위) 설정 값을 반환합니다.
Byte = <CHostDataEx>.ActiveKbds	Byte	실행 중인 키보드 및 마우스 설정을 반환합니다. 올바른 값은 다음과 같습니다. ■ 0: 호스트 및 원격지 ■ 1: 호스트 ■ 2: 원격지
Byte = <CHostDataEx>.InactiveTimeout	Byte	유휴 시간 제한이 만료되면 연결을 끊기 전에 대기할 시간(분)을 반환합니다.
Bool = <CHostDataEx>.LockSystemWhileWait	Bool	시작할 때 호스트 시스템 잠금 설정을 반환합니다.
Bool = <CHostDataEx>.MinimizeOnLaunch	Bool	아이콘 형태로 호스트 실행 시작 설정을 반환합니다.
Bool = <CHostDataEx>.RunAsService	Bool	서비스로 호스트 실행 설정을 반환합니다.

표 2-30 CHostDataEx 속성 및 반환 값 (계속)

속성	반환 값	설명
Byte = <CHostDataEx>.ConnLostWait	Byte	다른 연결을 허용하기 전에 대기할 시간(분)을 반환합니다.
Bool = <CHostDataEx>.ConnLostHostOpts	Bool	세션이 비정상적으로 종료된 경우 다른 연결에 대기할지 아니면 호스트를 취소할지 반환합니다. 올바른 값은 다음과 같습니다. ■ FALSE(거짓): 대기 ■ TRUE(참): 호스트 취소
Bool = <CHostDataEx>.ConnLostWait	Bool	비정상적인 세션 종료를 처리하기 위해 설정된 보안 옵션 값을 반환합니다.
Byte = <CHostDataEx>.AuthenticationType	Byte	인증 유형 참조 번호를 반환합니다. 올바른 값은 다음과 같습니다. ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS (Active Directory 서비스) ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID 인증 값을 설정하지 않거나 설정 값이 올바르지 않으면(예, 사용할 수 없는 인증 유형) 기본값으로 pcAnywhere 인증이 사용됩니다.
Byte = <CHostDataEx>.ConnLostSecurity	Byte	비정상적인 세션 종료를 처리하기 위해 설정된 보안 수준의 숫자 값을 반환합니다. 올바른 값은 다음과 같습니다. ■ 1: 사용자 로그오프 ■ 2: 호스트 시스템 재시작 ■ 3: 시스템 잠금
Byte = <CHostDataEx>.CallbkDelay	Byte	호스트 모뎀이 다시 원격지를 호출하기 전에 대기할 시간(초)을 반환합니다.

표 2-30CHostDataEx 속성 및 반환 값 (계속)

속성	반환 값	설명
Bool = <CHostDataEx>.EndSessHostOpts	Bool	세션이 비정상적으로 종료된 후 호스트를 다른 연결에 대기할지 아니면 취소할지 반환합니다. 올바른 값은 다음과 같습니다. ■ FALSE(거짓): 다음 연결에 대기 ■ TRUE(참): 호스트 취소
Bool = <CHostDataEx>.EnableEndSessSecurity	Bool	정상적인 세션 종료에 대한 세션 보안 옵션 종료의 실행 여부를 반환합니다.
Byte = <CHostDataEx>.EndSessSecurity	Byte	정상적인 세션 종료에 대해 설정된 보안 옵션을 반환합니다. 올바른 값은 다음과 같습니다. ■ 1: 사용자 로그오프 ■ 2: 호스트 시스템 재시작 ■ 3: 시스템 잠금
Bool = <CHostDataEx>.BlankHost	Bool	설정된 화면 지움 옵션을 반환합니다.
Bool = <CHostDataEx>.AllowRemoteMouse	Bool	세션을 수행하는 동안 원격지 사용자의 키보드와 마우스 제어 여부를 반환합니다.
Bool = <CHostDataEx>.RebootOnDisconnect	Bool	세션 종료 이후 호스트 시스템의 재시작 여부를 반환합니다.
Bool = <CHostDataEx>.PasswordAfterDisc	Bool	세션 종료 이후 호스트 사용자의 로그오프 여부를 반환합니다.
Bool = <CHostDataEx>.LogFailures	Bool	암호 실패 기록 옵션의 설정 여부를 반환합니다.
Bool = <CHostDataEx>.AllowDriveSecurity	Bool	드라이브 보안 옵션의 실행 여부를 반환합니다.
Bool = <CHostDataEx>.UseDirectoryServices	Bool	디렉터리 서비스 옵션의 실행 여부를 반환합니다.
String = <CHostDataEx>.DirectoryServiceEntry	String	디렉터리 서비스 설정을 반환합니다.

호스트 기능의 Visual Basic 샘플 코드

다음 Visual Basic 샘플 코드는 호스트 데이터 개체를 검색하고 해당 속성을 수정합니다.

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

```
'Create CHostDataManager object
Set HostDataManager = CreateObject(WINAWSVR.BEHOSTDATAMANAGER)

'display and change current directory
s = HostDataManager.CurrentDirectory()
MsgBox ( s )
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
s = HostDataManager.CurrentDirectory()
MsgBox ( s )

'retrieve remote data object
Set HostData = HostDataManager.RetrieveObject("pod.BHF", 2, 0)

'display some properties
s = HostData.AreaCode()
MsgBox (s)
s = HostData.PhoneNumber()
MsgBox (s)
'set some properties
RemoteData.HostData = "212"
RemoteData.HostData = "555-5555"

'write object to disk
HostData.WriteObject (0)
End Sub
```

FindFirst와 **FindNext** 메소드를 사용하여 다음과 같이 디렉터리에 호스트 파일을 표시하십시오.

```
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 함수

Awrem32 함수는 호스트와 원격지 시스템 간의 연결을 처리하기 위한 매개 변수와 반환 값을 제공합니다.

Awrem32 함수	매개 변수 또는 반환 값	설명
awConnect(FileName)	매개 변수 = Name as string	호스트 시스템에 관한 정보를 포함한 기준에 완전히 일치하는 .chf 파일 이름
awConnect	반환 값 = Boolean	명령을 실행했으면 참
awDisconnect()	반환 값 = Boolean	이 함수를 호출한 후, 호출 프로그램은 개체를 삭제해야 합니다(++-deleteIAwrem32X*, VB -set ObjectName = Nothing.).
FileXferFromHost(HostFile)	매개 변수 = HostFile as string	호스트 시스템에서 복사할 전체 경로 및 파일 이름을 포함합니다.
FileXferFromHost(RemoteFile)	매개 변수 = RemoteFile as string	기준에 완전히 일치하는 대상 경로와 파일 이름을 포함합니다. HostFile과 RemoteFile 문자열이 일치할 필요는 없습니다.
FileXferFromHost	반환 값 = Boolean	명령을 실행했으면 참
FileXferToHost(HostFile)	매개 변수 = HostFile as string	원격지 시스템에 있는 파일을 호스트 시스템에 복사합니다. 매개 변수에 와일드 카드 문자를 포함할 수 있습니다. 매개 변수는 전체 대상 경로와 파일 이름을 포함합니다.
FileXferToHost(RemoteFile)	매개 변수 = RemoteFile as string	원격지 시스템에서 복사할 전체 경로 및 파일 이름을 포함합니다. HostFile과 RemoteFile 문자열이 일치할 필요는 없습니다.

Awrem32 함수	매개 변수 또는 반환 값	설명
CreateFolderOnHost(FolderName)	매개 변수 = FolderName as string	호스트 시스템에 폴더를 만들도록 드라이브와 경로를 포함합니다.
CreateFolderOnHost	반환 값 = Boolean	명령을 실행했으면 참
ExecuteHostFile(FileName)	매개 변수 = FileName as string	호스트 시스템에 있는 기존의 파일을 실행합니다. 이 함수는 배치, 명령, 실행 파일만 실행합니다. 실행 프로그램과 관련된 파일은 실행하지 않습니다. 예를 들어, 이 함수는 .doc 파일을 실행했을 때 Microsoft Word를 열지 않습니다. 매개 변수는 호스트 시스템에 있는 파일의 전체 경로를 포함합니다.
ExecuteHostFile	반환 값 = Boolean	명령을 실행했으면 참
GetError()	반환 값 = String	Awrem32에서 마지막으로 생성된 오류를 반환합니다.
ConnectionStatus()	반환 값 = Short	호스트 시스템으로 현재 사용자 연결 상태를 반환합니다. 사용할 수 있는 값은 다음과 같습니다. ■ -1 = 손실된 연결 ■ 0 = 연결 안됨 ■ 1 = 세션 연결됨

Visual C++ 개체 정의

이 장의 내용은 다음과 같습니다.

- Visual C++ 개체 정보
- CRemoteDataManager 메소드
- CRemoteData 개체
- CRemoteDataEx 개체
- CHostDataManager 메소드
- CHostData 개체
- CHostDataEx 개체
- Awrem32 함수

Visual C++ 개체 정보

pcAnywhere 자동화 서버는 OLE 자동화를 지원하기 위해 다음 구성 요소를 제공합니다.

Winawsvr	pcAnywhere 자동화 서버에 연결하고 해당 인터페이스에 액세스하는 데 필요한 정보를 제공합니다.
Awrem32	pcAnywhere 연결을 제어하는 데 필요한 정보를 제공합니다.

표 3-1에서 설명하는 개체는 Winawsvr를 구성합니다.

표 3-1 Winawsvr 개체

Object	설명	참조 사항
CRemoteDataManager	CRemoteData 개체를 만들고, 열고, 수정하고, 저장하고, 삭제하는 메소드를 제공합니다.	56페이지의 “CRemoteDataManager 메소드” 참조
CRemoteData	pcAnywhere 원격지 함수에 액세스하고 이를 제어하는 매개 변수를 정의합니다.	59페이지의 “CRemoteData 개체” 참조 65페이지의 “CRemoteDataEx 개체” 참조
CHostDataManager	CHostData 개체를 만들고, 열고, 수정하고, 저장하고, 삭제하는 메소드를 제공합니다.	66페이지의 “CHostDataManager 메소드” 참조
CHostData	pcAnywhere 호스트 함수에 액세스하고 이를 제어하는 매개 변수를 정의합니다.	59페이지의 “CRemoteData 개체” 참조 65페이지의 “CRemoteDataEx 개체” 참조

Awrem32는 8개의 인터페이스로 구성된 하나의 개체로 원격 제어 및 파일 전송 세션을 지원합니다.

79페이지의 “Awrem32 함수” 참조

일부 함수는 더 이상 지원되지 않습니다. 그러나 이전 버전에 사용할 수 있는 개체 정의는 제공됩니다.

암호가 필요한 함수의 경우, 암호 값을 설정할 수는 있지만 검색할 수는 없습니다. 이는 보안 유지를 위한 조치입니다.

CRemoteDataManager 메소드

CRemoteDataManager 메소드는 CRemoteData 개체에 액세스하고 이를 제어할 수 있는 매개 변수와 반환 값을 제공합니다.

방법	매개 변수 또는 반환 값	설명
BSTR CurrentDirectory();	매개 변수 = BSTR	pcAnywhere 원격지 개체가 저장된 현재 pcAnywhere 데이터 디렉터리의 전체 경로 이름
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	매개 변수 = LPCTSTR lpszNewDirectory 반환 값 = BOOL	pcAnywhere 원격지 개체가 저장된 현재 디렉터를 변경합니다. 매개 변수는 기존 디렉터리의 이름입니다. 성공한 경우 반환 값은 TRUE입니다.
BOOL FindFirst(LPCTSTR lpszPattern, BSTR FAR*);	매개 변수 = LPCTSTR lpszPattern	개체 파일을 필터링하는 파일 이름 패턴(별표[*]는 현재 디렉터리에 있는 모든 파일을 찾음)

방법	매개 변수 또는 반환 값	설명
BOOL FindFirst(pbstrFullQualName);	매개 변수 = BSTR FAR * pbstrFullQualName	지정된 패턴에 일치하는 원격지 개체 파일의 전체 경로 이름에 대해 반환된 버퍼
BOOL FindFirst	반환 값 = BOOL	지정된 패턴에 일치하는 원격지 개체 파일을 찾으면 참. pbstrFullQualName에 일치하는 파일의 전체 경로 이름을 저장합니다.
BOOL FindNext(BSTR FAR * pbstrFullQualName);	매개 변수 = BSTR FAR * pbstrFullQualName 반환 값 = BOOL	FindFirst()를 성공적으로 호출하여 원격지 개체 파일의 이름을 확인하였으면 그 다음에는 패턴에 맞는 다음 파일이 있을 경우 FindNext()를 호출하여 그 파일을 찾을 수 있습니다. FindFirst()의 호출에 지정된 패턴에 일치하는 다른 원격지 개체 파일을 찾으면 반환 값은 TRUE입니다. pbstrFullQualName에 일치하는 파일의 전체 경로 이름을 저장합니다.
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName	파일 이름으로 CRemoteData 개체를 검색합니다. 매개 변수는 로드할 전체 원격지 개체 파일 이름입니다.
LPDISPATCH RetrieveObject(short wAccessMode);	매개 변수 = short wAccessMode	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword	개체 암호. NULL일 수 있음
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName	파일 이름으로 CRemoteDataEx 개체를 검색합니다. 매개 변수는 로드할 전체 원격지 개체 파일 이름입니다.

방법	매개 변수 또는 반환 값	설명
LPDISPATCH RetrieveObjectEx(short wAccessMode);	매개 변수 = short wAccessMode	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword	개체 암호. NULL일 수 있음
LPDISPATCH RetrieveObjectEx	반환 값 = LPDISPATCH	OLE 디스패치 개체의 포인터. 개체는 CRemoteDataEx 개체입니다. 원격지 함수의 Visual C++ 샘플 코드 부분을 참조하십시오.
LPDISPATCH CreateObject(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName 반환 값 = LPDISPATCH	CRemoteData 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 원격지 개체 파일 이름입니다. 반환 값은 OLE 디스패치 개체의 포인터입니다. 개체는 CRemoteData 개체입니다. 원격지 함수의 Visual C++ 샘플 코드 부분을 참조하십시오.
LPDISPATCH CreateObjectEx(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName 반환 값 = LPDISPATCH	CRemoteDataEx 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 원격지 개체 파일 이름입니다. 반환 값은 OLE 디스패치 개체의 포인터입니다. 개체는 CRemoteDataEx 개체입니다. 원격지 함수의 Visual C++ 샘플 코드 부분을 참조하십시오.
BOOL DeleteObject(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName	원격지 개체 파일을 삭제합니다. 매개 변수는 삭제할 개체의 전체 원격지 개체 파일 이름입니다.

방법	매개 변수 또는 반환 값	설명
<code>BOOL DeleteObject(LPCTSTR lpszPassword);</code>	매개 변수 = LPCTSTR lpszPassword	개체 암호. NULL일 수 있음
<code>BOOL DeleteObject</code>	BOOL	개체를 삭제하면 참
<code>BOOL Launch(LPCTSTR lpszFQName);</code>	매개 변수 = LPCTSTR lpszFQName 반환 값 = BOOL	pcAnywhere 원격지 터미널 창을 여는 원격지 개체 파일을 실행합니다. 매개 변수는 시작할 개체의 전체 원격지 개체 파일입니다. 개체 시작에 성공한 경우 반환 값은 TRUE입니다.

CRemoteData 개체

이 개체를 사용하여 원격지 개체 데이터를 수정하십시오.

Get/Set 메소드

CRemoteData 개체의 속성을 불러오고 설정하는 데 다음 메소드를 사용합니다.

시스템 이름은 원격지 개체를 시작할 때 호출하는 **pcAnywhere** 호스트 시스템의 이름입니다.

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

전화 번호는 **pcAnywhere** 호스트 시스템에 모뎀 연결을 수행할 때 전화 거는 번호입니다.

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

TAPI 전화 걸기 속성을 사용해야 할지(위치 정보)(참) 또는 전화 번호 문자열을 보이는 그대로 정확하게 사용해야 할지(거짓) 그 여부를 나타냅니다.

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

전화 걸기 속성을 사용하면 이것은 전화 번호의 지역 코드입니다.

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

전화 걸기 속성을 사용하면 이것은 전화 번호의 국가 코드입니다.

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

호출에 실패하는 경우, 이 전화 번호를 재다이얼할 횟수.

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

재다이얼 시도 간의 대기 시간(초).

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

연결하면 호스트에 전송할 로그인 이름. 이 항목이 빈 칸이면 연결할 때 사용자에게 로그인 이름을 묻는 메시지를 표시합니다.

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

연결하면 호스트에 전송할 로그인 암호. 이 항목이 빈 칸이면 연결할 때 사용자에게 암호를 묻는 메시지를 표시합니다.

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

이 개체의 암호.

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

암호를 사용하는 경우에만 개체를 시작할 수 있습니다(참).

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

정확한 암호를 사용하는 경우에만 개체를 볼 수 있습니다(참).

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

정확한 암호를 사용하는 경우에만 개체를 기록할 수 있습니다(참).

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

이 개체를 사용하는 세션을 기록할지 제어합니다.

```

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

```

이 개체를 사용하는 세션을 처음부터 기록할지 제어합니다.

```

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

```

이 개체를 사용하는 세션의 기록 파일 이름.

```

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

```

연결 유형을 설정합니다.

```

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

```

다음 연결 유형을 사용할 수 있습니다.

- COM1
- COM2
- COM3
- COM4
- CAPI 2.0을 사용하는 ISDN
- NetBIOS
- SPX
- 기본 TAPI
- TCP/IP

TAPI 장치 이름도 연결 유형으로 사용할 수 있습니다.[기본 TAPI]는 시스템에서 찾은 첫 번째 TAPI 장치를 사용합니다. 특정 TAPI 장치를 사용하려면 사용할 수 있는 장치를 검색하도록 FirstConnectionType() / NextConnectionType()을 사용하십시오.

원격지 개체 세부 사항 메소드

원격지 개체를 연결 유형에 할당하면 장치 세부 사항을 유효한 기본값으로 설정합니다.

다음 연결 유형에는 응용 프로그램에서 설정할 수 있는 고급 설정 옵션이 들어 있습니다.

- COM 장치
- NetBIOS 장치
- CAPI 2.0을 사용하는 ISDN(유럽 ISDN 전용) 장치

COM 장치 세부 사항

통신 패리티 단계를 설정합니다.

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

통신 패리티 값은 다음과 같습니다.

- 없음
- 홀수
- 짝수
- 기호
- 공백

흐름 제어 단계를 설정합니다.

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

흐름 제어 값은 다음과 같습니다.

- <없음>
- XONXOFF
- RTS/CTS
- 모두

연결 시작 설정을 구성합니다.

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

연결 시작 값은 다음과 같습니다.

- 항상 연결됨
- 반송파 탐지 (DCD)
- 전송 클리어 (CTS)
- 데이터 세트 준비 (DSR)
- 링 표시기 (RI)
- 2 <CR> 수신
- 모뎀 응답

연결 종료 값을 설정합니다.

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

연결 종료 값은 다음과 같습니다.

- 항상 연결됨
- 반송과 탐지 (DCD)
- 전송 클리어 (CTS)
- 데이터 세트 준비 (DSR)
- 링 표시기 (RI)

연결 속도를 설정합니다.

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

연결 속도 값은 다음과 같습니다.

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

NetBIOS 장치 세부 사항

현재 연결에 사용할 LANA(LAN 어댑터) 번호를 설정합니다.

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

ISDN via CAPI 2.0 장치 세부 사항

채널 연결을 활성화합니다(한 연결에 2 ISDN 채널 사용)(참).

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

통신에 필요한 추가 CAPI 내선 번호를 설정합니다.

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

원격지 개체 메소드

다음 표에서는 원격지 개체의 일반적인 메소드를 설명합니다. 이러한 메소드는 속성을 가져오고 설정하는 데 사용하지 않습니다.

방법	매개 변수 또는 반환 값	설명
short ConnectionTypes();	반환 값 = Short	이 시스템에서 사용할 수 있는 연결 유형 수
BSTR FirstConnectionType();과 BSTR NextConnectionType();	반환 값 = BSTR	사용할 수 있는 연결 유형을 통해 반복하는 데 FirstConnectionType()과 NextConnectionType()을 사용합니다. 이들 함수는 사용할 수 있는 연결 유형의 이름인 BSTR을 반환합니다. 이러한 반환된 연결 유형은 SetConnectionType() 함수에 사용할 수 있습니다. 반환 값은 지원되는 연결 장치 종류 이름입니다.
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	매개 변수 = LPCTSTR lpzConnectionTyp 반환 값 = BOOL	연결 장치 유형 이름 전달된 연결 유형이 시스템에 있으면 반환 값은 TRUE입니다.
short CountryCodes();	반환 값 = Short	사용할 수 있는 국가 코드 번호를 반환합니다.
BSTR FirstCountryCode();와 BSTR NextCountryCode();	반환 값 = BSTR	사용할 수 있는 국가 코드를 통해 반복하는 데 FirstCountryCode()와 NextCountryCode()를 사용합니다. 이들 함수는 사용할 수 있는 국가 코드의 이름인 BSTR을 반환합니다. 이러한 반환된 연결 유형은 SetCountryCode() 함수에 사용할 수 있습니다. 반환 값은 첫 번째나 그 다음 국가 코드 문자열입니다.

방법	매개 변수 또는 반환 값	설명
BOOL ReadObject(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword 반환 값 = BOOL	원격지 개체 파일에서 개체 데이터를 읽습니다. 매개 변수는 개체 암호입니다. 개체 읽기에 성공한 경우 반환 값은 TRUE입니다.
BOOL WriteObject(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword 반환 값 = BOOL	개체 데이터를 원격지 개체 파일에 기록합니다. 매개 변수는 개체 암호입니다. 개체 쓰기에 성공한 경우 반환 값은 TRUE입니다.

CRemoteDataEx 개체

CRemoteDataEx 개체는 CRemoteData 개체와 동일한 함수를 가지며 다음과 같은 **Get** 및 **Set** 메소드가 추가적으로 포함되어 있습니다.

```
BSTR GetPrivateKey(); //Returns the PrivateKey information
void SetPrivateKey(LPCTSTR lpszNewValue);
BSTR GetCertificationName(); //Returns the Certification Name
void SetCertificationName(LPCTSTR lpszNewValue);
short GetEncryptionLevel(); //Returns the encryption level value
void SetEncryptionLevel(short nNewValue);
BOOL GetDenyLowerEncrypt(); //Returns the DenyLowerEncrypt value
void SetDenyLowerEncrypt(BOOL bNewValue);
BSTR GetAutoDomain(); //Returns the AutoDomain value
void SetAutoDomain(LPCTSTR lpszNewValue);
```

원격지 함수의 Visual C++ 샘플 코드

다음의 샘플 C++ 함수는 원격지 개체를 생성하고 그 연결 유형을 **TCP/IP**에 설정합니다. 그리고 함수에 전달된 **TCP/IP** 주소에 시스템 이름을 설정한 후 원격지 개체를 실행합니다.

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

```
CRemoteDataManager remoteDM;

CRemoteData remoteData;

// First, create the CRemoteDataManager
remoteDM.CreateDispatch( _T( "WINAWSVR.RemoteDataManager" ) );
// Next, create CRemoteData and attach it
remoteData.AttachDispatch( remoteDM.CreateObject("Test", 0) );

// Now, set the required properties
remoteData.SetConnectionType("TCP/IP");

remoteData.SetComputerName(lpszAddress);

// Save the object data
if (remoteData.WriteObject(0))
{
    // And launch it
    if (remoteData.Launch())
        bReturn = TRUE;
}
// Release the remote object.
remoteData.ReleaseDispatch();

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

return bReturn;
}
```

CHostDataManager 메소드

다음 표에서는 CHostData 개체를 액세스하고 제어하기 위한 매개 변수 및 반환 값을 포함하여 CHostDataManager 메소드를 설명합니다.

방법	매개 변수 또는 반환 값	설명
BSTR CurrentDirectory();	반환 값 = BSTR	pcAnywhere 호스트 개체를 저장한 현재 디렉터리의 전체 경로 이름을 반환합니다.
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	매개 변수 = LPCTSTR lpszNewDirectory 반환 값 = BOOL	pcAnywhere 호스트 개체가 저장된 현재 디렉터리를 변경합니다. 매개 변수는 기존 디렉터리의 이름입니다. 성공한 경우 반환 값은 TRUE입니다.
BOOL FindFirst(LPCTSTR lpszPattern);	매개 변수 = LPCTSTR lpszPattern 반환 값 = BOOL	지정된 파일 이름 패턴을 근거로 현재 디렉터리에서 첫 번째 pcAnywhere 호스트 개체 파일(*.bhf)을 찾습니다. 매개 변수는 개체 파일을 필터링하는 파일 이름 패턴입니다(별표[*]는 현재 디렉터리에 있는 모든 호스트 파일을 찾음). 지정된 패턴에 일치하는 호스트 개체 파일을 찾으면 반환 값은 TRUE입니다. pbstrFullQualName에 일치하는 파일의 전체 경로 이름을 저장합니다.
BOOL FindFirst(BSTR FAR* pbstrFullQualName);	매개 변수 = BSTR FAR* *pbstrFullQualName	지정된 패턴에 일치하는 원격지 개체 파일의 전체 경로 이름에 대해 반환된 버퍼 지정된 패턴에 일치하는 호스트 개체 파일을 찾으면 반환 값은 TRUE입니다. pbstrFullQualName에 일치하는 파일의 전체 경로 이름을 저장합니다.
BOOL FindNext(BSTR FAR* pbstrFullQualName);	매개 변수 = BSTR FAR* *pbstrFullQualName 반환 값 = BOOL	FindFirst()를 성공적으로 호출하여 호스트 개체 파일의 이름을 확인하였으면 그 다음에는 패턴에 맞는 다음 파일이 있을 경우 FindNext()를 호출하여 그 파일을 찾을 수 있습니다. 매개 변수는 FindFirst()의 원래 호출에 지정된 패턴에 일치하는 호스트 개체 파일의 전체 경로 이름에 대한 반환 버퍼입니다. FindFirst()의 호출에 지정된 패턴에 일치하는 다른 호스트 개체 파일을 찾으면 반환 값은 TRUE입니다.pbstrFullQualName에 일치하는 파일의 전체 경로 이름을 저장합니다.

방법	매개 변수 또는 반환 값	설명
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName	파일 이름으로 CHostData 개체를 검색합니다. 매개 변수는 로드할 전체 호스트 개체 파일 이름입니다.
LPDISPATCH RetrieveObject(short wAccessMode);	매개 변수 = short wAccessMode	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword	개체 암호. NULL일 수 있음
LPDISPATCH RetrieveObject	반환 값 = LPDISPATCH	OLE 디스패치 개체의 포인터. 개체는 CHostData 개체입니다. 호스트 함수의 Visual C++ 샘플 코드 부분을 참조하십시오.
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName	파일 이름으로 CHostDataEx 개체를 검색합니다. 매개 변수는 로드할 전체 호스트 개체 파일 이름입니다.
LPDISPATCH RetrieveObjectEx(short wAccessMode);	매개 변수 = short wAccessMode	이 개체를 사용할 방법을 지정합니다. 이는 암호 보호와 관련이 있습니다. 옵션은 다음과 같습니다. ■ 0 = 지정되지 않음 ■ 1 = 보기 전용 ■ 2 = 보기 및 수정 ■ 3 = 실행
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword	개체 암호. NULL일 수 있음

방법	매개 변수 또는 반환 값	설명
LPDISPATCH RetrieveObjectEx	반환 값 = LPDISPATCH	OLE 디스패치 개체의 포인터. 개체는 CHostDataEx 개체입니다. 호스트 함수의 Visual C++ 샘플 코드 부분을 참조하십시오.
LPDISPATCH CreateObject(LPCTSTR lpszName);	매개 변수 = LPCTSTR lpszFQName 반환 값 = LPDISPATCH	CHostData 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 호스트 개체 파일 이름입니다. 반환 값은 OLE 디스패치 개체의 포인터입니다. 개체는 CHostData 개체입니다. 호스트 함수의 Visual C++ 샘플 코드 부분을 참조하십시오.
LPDISPATCH CreateObjectEx(LPCTSTR lpszName);	매개 변수 = LPCTSTR lpszFQName 반환 값 = LPDISPATCH	CHostDataEx 개체를 만들고, 그 개체에 LPDISPATCH 포인터를 반환합니다. 매개 변수는 새 개체의 전체 호스트 개체 파일 이름입니다. 반환 값은 OLE 디스패치 개체의 포인터입니다. 개체는 CHostDataEx 개체입니다. 호스트 함수의 Visual C++ 샘플 코드 부분을 참조하십시오.
BOOL DeleteObject(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName	호스트 개체 파일을 삭제합니다. 매개 변수는 삭제할 개체의 전체 호스트 개체 파일 이름입니다.
BOOL DeleteObject(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword	개체 암호. NULL일 수 있음
BOOL DeleteObject	반환 값 = BOOL	개체를 삭제하면 참
BOOL Launch(LPCTSTR lpszFQName);	매개 변수 = LPCTSTR lpszFQName 반환 값 = BOOL	pcAnywhere 호스트 터미널 창을 여는 호스트 개체 파일을 시작합니다. 매개 변수는 시작할 개체의 전체 호스트 개체 파일입니다. 개체 시작에 성공한 경우 반환 값은 TRUE입니다.

CHostData 개체

이 개체를 사용하여 호스트 개체 데이터를 수정하십시오.

Get/Set 메소드

CHostData 개체의 속성을 불러오고 설정하는 데 다음 메소드를 사용합니다.

전화 번호는 **pcAnywhere** 원격지 시스템에 모뎀 연결을 수행할 때 거는 번호입니다.

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

TAPI 전화 걸기 속성을 사용해야 할지(위치 정보)(참) 또는 전화 번호 문자열을 보이는 그대로 정확하게 사용해야 할지(거짓) 그 여부를 나타냅니다.

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

전화 걸기 속성을 사용하면 이것은 전화 번호의 지역 코드입니다.

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

전화 걸기 속성을 사용하면 이것은 전화 번호의 국가 코드입니다.

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

호출에 실패하는 경우, 이 전화 번호를 재다이얼할 횟수.

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

재다이얼 시도 간의 대기 시간(초).

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

이 개체를 사용하는 세션을 기록할지 제어합니다.

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

이 개체를 사용하는 세션을 처음부터 기록할지 제어합니다.

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

이 개체를 사용하는 세션의 기록 파일 이름.

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

호스트 개체 세부 사항 메소드

호스트 개체를 연결 유형에 할당하면 장치 세부 사항을 유효한 기본값으로 설정합니다.

다음 연결 유형에는 응용 프로그램에서 설정할 수 있는 고급 설정 옵션이 들어 있습니다.

- COM 장치
- NetBIOS 장치
- NASI/NCIS 장치
- CAPI 2.0을 사용하는 ISDN (유럽 ISDN 전용) 장치

COM 장치 세부 사항

다음 코드는 연속적인 장치 지정 방법 호출을 처리할 때, 할당된 연결 유형의 호스트 개체 목록에 요청된 연결 유형을 배치하고 현재 연결 유형으로 설정합니다.

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

요청된 연결 유형이 할당된 연결 목록에 이미 있으면 할당된 연결 목록이 변경되지 않습니다. 현재 연결 유형이 요청된 유형으로 변경되기만 합니다. 연결 지정 값을 불러오고 설정하는 과정에서 여러 번 동일한 개체에서 **AssignConnection** 메소드를 호출하는 것은 일반적입니다.

AssignConnection은 전송된 연결 유형이 시스템에 있고, 이미 할당되었거나, 할당할 수 있는 경우, 참을 반환합니다. 요청된 연결 유형이 시스템에 없거나, 현재 할당된 연결 수가 허용된 최대 수준에 이미 도달한 경우, 거짓을 반환합니다.

pcAnywhere 호스트 개체는 할당된 연결 유형을 두 개까지 지원할 수 있습니다. 이러한 시도가 제한 값을 초과한 경우, **AssignConnection** 메소드는 거짓을 반환합니다.

다음 연결 유형을 사용할 수 있습니다.

- COM1
- COM2
- COM3
- COM4
- SPX
- NetBIOS
- TCP/IP

- CAPI 2.0을 사용하는 ISDN

- 기본 TAPI

TAPI 장치 이름도 연결 유형으로 사용할 수 있습니다. [기본 TAPI]는 시스템에서 찾은 첫 번째 TAPI 장치를 사용합니다. 특정 TAPI 장치를 사용하려면 사용할 수 있는 장치를 검색하도록 `FirstConnectionType()`과 `NextConnectionType()`을 사용하십시오.

다음 코드는 연결 유형 할당을 제거합니다. 연결 종류의 할당을 제거한 후, 남아있는 할당된 연결이 있으면 연속적인 장치 지정 메소드 호출의 현재 연결 유형으로 설정됩니다.

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

다음 코드는 통신 패리티 단계를 설정합니다.

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

통신 패리티 값은 다음과 같습니다.

- 없음
- 홀수
- 짝수
- 기호
- 공백

다음 코드는 흐름 제어 단계를 설정합니다.

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

흐름 제어 값은 다음과 같습니다.

- <없음>
- XONXOFF
- RTS/CTS
- 모두

다음 코드는 연결 시작 값을 설정합니다.

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

연결 시작 값은 다음과 같습니다.

- 항상 연결됨

- 반송파 탐지 (DCD)
- 전송 클리어 (CTS)
- 데이터 세트 준비 (DSR)
- 링 표시기 (RI)
- 2 <CR> 수신
- 모뎀 응답

다음 코드는 연결 종료 값을 설정합니다.

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

연결 종료 값은 다음과 같습니다.

- 항상 연결됨
- 반송파 탐지 (DCD)
- 전송 클리어 (CTS)
- 데이터 세트 준비 (DSR)
- 링 표시기 (RI)

다음 코드는 연결 속도를 설정합니다.

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

연결 속도 값은 다음과 같습니다.

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

NetBIOS 장치 세부 사항

현재 연결에 사용할 LANA(LAN 어댑터) 번호를 설정합니다.

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

NAS/NCSI 장치 세부 사항

NAS 서버의 사용자 이름을 설정합니다.

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

NAS 서버의 사용자 암호를 설정합니다.

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

NAS 세션 이름을 설정합니다.

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

사용할 NAS 서버를 지정합니다.

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

사용할 NAS 서비스를 지정합니다.

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

사용할 NAS 포트를 지정합니다.

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

ISDN via CAPI 2.0 장치 세부 사항

채널 연결을 활성화합니다(한 연결에 2 ISDN 채널 사용)(참).

```

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

```

통신에 필요한 추가 CAPI 내선 번호를 설정합니다.

```

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

```

호스트 개체 메소드

다음 표에서는 개체의 일반적인 메소드를 설명합니다. 이러한 메소드는 속성을 가져오고 설정하는 데 사용하지 않습니다.

방법	매개 변수 또는 반환 값	설명
short ConnectionTypes();	반환 값 = Short	이 시스템에서 사용할 수 있는 연결 유형 수를 반환합니다.
BSTR FirstConnectionType(); 과 BSTR NextConnectionType();	반환 값 = BSTR	사용할 수 있는 연결 유형을 통해 반복하는 데 FirstConnectionType()과 NextConnectionType()을 사용합니다. 함수는 사용할 수 있는 연결 유형의 이름인 BSTR을 반환합니다. 이러한 반환된 연결 유형은 SetConnectionType() 함수에 사용할 수 있습니다.
		반환 값은 지원되는 연결 장치 종류 이름입니다.
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	매개 변수 = LPCTSTR lpszConnectionType 반환 값 = BOOL	매개 변수는 연결 장치 유형의 이름입니다. 이 장치 유형을 사용할 수 있으면 반환 값은 TRUE입니다.
short MaxAssignedConnections()	반환 값 = Short	동시에 할당할 수 있는 최대 연결 유형 수를 반환합니다(현재는 2).
short AssignedConnections()	반환 값 = Short	할당된 연결 유형 수를 반환합니다.

방법	매개 변수 또는 반환 값	설명
BSTR FirstAssignedConnection(); and BSTR NextAssignedConnection();	반환 값 = BSTR	할당된 연결 목록을 통해 반복하는 데 FirstAssignedConnection()과 NextAssignedConnection()을 사용합니다. 이들 함수는 할당된 연결 유형의 이름인 BSTR을 반환합니다. 이러한 반환된 연결 유형은 AssignConnection() 함수에 사용할 수 있습니다. 반환 값은 지원되는 연결 장치 종류 이름입니다.
BOOL FindAssignedConnection (LPCTSTR lpszConnectionType);	매개 변수 = LPCTSTR lpszConnectionTyp 반환 값 = BOOL	매개 변수는 연결 장치 유형의 이름입니다. 이 장치 유형이 현재 할당되었으면 반환 값은 TRUE입니다.
short CountryCodes();	반환 값 = Short	사용할 수 있는 국가 코드 번호를 반환합니다.
BSTR FirstCountryCode();와 BSTR NextCountryCode();	반환 값 = BSTR	사용할 수 있는 국가 코드 목록을 통해 반복하는 데 FirstCountryCode()와 NextCountryCode()를 사용합니다. 함수는 사용할 수 있는 국가 코드의 이름인 BSTR을 반환합니다. 이러한 반환된 연결 유형은 SetCountryCode() 함수에 사용할 수 있습니다. 반환 값은 첫 번째나 그 다음 국가 코드 문자열입니다.
BOOL ReadObject(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword 반환 값 = BOOL	호스트 개체 파일에서 개체 데이터를 읽습니다. 매개 변수는 개체 암호입니다. 개체 읽기에 성공한 경우 반환 값은 TRUE입니다.
BOOL WriteObject(LPCTSTR lpszPassword);	매개 변수 = LPCTSTR lpszPassword 반환 값 = BOOL	개체 데이터를 호스트 개체 파일에 기록합니다. 매개 변수는 개체 암호입니다. 개체 쓰기에 성공한 경우 반환 값은 TRUE입니다.

CHostDataEx 개체

CHostDataEx 개체는 CHostData 개체와 동일한 함수를 가지며 다음과 같은 Get 및 Set 메소드가 추가적으로 포함되어 있습니다.

```
BOOL GetReadProtection();
void SetReadProtection(BOOL bNewValue);
BOOL GetWriteProtection();
void SetWriteProtection(BOOL bNewValue);
BSTR GetPassword(); //Returns "NOT IMPLEMENTED"
void SetPassword(LPCTSTR lpszNewValue);
BSTR GetCallersPath();
void SetCallersPath(LPCTSTR lpszNewValue);
BOOL GetConfirmConnect();
void SetConfirmConnect(BOOL bNewValue);
short GetConfirmTimeout();
void SetConfirmTimeout(short nNewValue);
BOOL GetConfirmDeny();
void SetConfirmDeny(BOOL bNewValue);
BOOL GetPwCaseSensitive();
void SetPwCaseSensitive(BOOL bNewValue);
short GetPwAttempts();
void SetPwAttempts(short nNewValue);
short GetPwTimeout();
void SetPwTimeout(short nNewValue);
short GetActiveKbds();
void SetActiveKbds(short nNewValue); //Sets ActiveKbds
short GetInactiveTimeout();
void SetInactiveTimeout(short nNewValue);
short GetCryptReqLevel();
void SetCryptReqLevel(short nNewValue);
BOOL GetCryptRefuseLower();
void SetCryptRefuseLower(BOOL bNewValue);
short GetAuthenticationType();
void SetAuthenticationType(short nNewValue);
BOOL GetLockSystemWhileWait();
void SetLockSystemWhileWait(BOOL bNewValue);
BOOL GetMinimizeOnLaunch();
void SetMinimizeOnLaunch(BOOL bNewValue);
BOOL GetRunAsService();
void SetRunAsService(BOOL bNewValue);
short GetConnLostWait();
void SetConnLostWait(short nNewValue);
```

```
BOOL GetConnLostHostOpts();  
void SetConnLostHostOpts(BOOL bNewValue);  
BOOL GetEnableConnLostSecurity();  
void SetEnableConnLostSecurity(BOOL bNewValue);  
short GetConnLostSecurity();  
void SetConnLostSecurity(short nNewValue);  
short GetCallbkDelay();  
void SetCallbkDelay(short nNewValue);  
BOOL GetEndSessHostOpts();  
void SetEndSessHostOpts(BOOL bNewValue);  
BOOL GetEnableEndSessSecurity();  
void SetEnableEndSessSecurity(BOOL bNewValue);  
short GetEndSessSecurity();  
void SetEndSessSecurity(short nNewValue);  
BSTR GetCryptPrivateKey();  
void SetCryptPrivateKey(LPCTSTR lpszNewValue);  
BSTR GetCryptCommonName();  
void SetCryptCommonName(LPCTSTR lpszNewValue);  
BOOL GetBlankHost();  
void SetBlankHost(BOOL bNewValue);  
BOOL GetAllowRemoteMouse();  
void SetAllowRemoteMouse(BOOL bNewValue);  
short GetRebootOnDisconnect();  
void SetRebootOnDisconnect(short nNewValue);  
BOOL GetPasswordAfterDisc();  
void SetPasswordAfterDisc(BOOL bNewValue);  
BOOL GetLogFailures();  
void SetLogFailures(BOOL bNewValue);  
BOOL GetAllowDriveSecurity();  
void SetAllowDriveSecurity(BOOL bNewValue);  
BOOL GetExecuteProtection();  
void SetExecuteProtection(BOOL bNewValue);
```

호스트 함수의 Visual C++ 샘플 코드

다음의 샘플 Visual C++ 함수는 호스트 개체를 생성하고 그 연결 유형을 TCP/IP에 설정합니다. 그리고 함수에 전달된 TCP/IP 주소에 시스템 이름을 설정한 후 호스트 개체를 실행합니다.

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

```
CHostDataManager hostDM;  
CHostData hostData;  
  
// First, create the CHostDataManager  
hostDM.CreateDispatch( _T( "WINAWSVR.BeHostDataManager" ) );  
  
// Next, create CRemoteData and attach it  
hostData.AttachDispatch(hostDM.CreateObject("Test", 0) );  
  
// Now, set the required properties  
hostData.SetConnectionType("TCP/IP");  
  
// Save the object data  
if (hostData.WriteObject(0))  
{  
    // And launch it  
    if (hostData.Launch())  
        bReturn = TRUE;  
}  
  
// Release the Host object.  
hostData.ReleaseDispatch();  
  
return (bReturn);  
}
```

Awrem32 함수

다음 표에서는 호스트와 원격지 시스템 간의 연결을 처리하기 위한 매개 변수와 반환 값을 포함한 Awrem32 함수에 대해 설명합니다.

메소드	매개 변수 또는 반환 값	설명
boolean awConnect(BSTR FileName);	매개 변수 = Name as string 반환 값 = Boolean	호스트 시스템으로 연결을 만듭니다. 매개 변수는 호스트 시스템에 관한 정보를 포함한 전체 .chf 파일 이름 반환 값이 명령을 실행합니다.
boolean awDisconnect();	반환 값 = Boolean	호스트 시스템의 연결을 끊습니다. 이 함수를 호출한 후, 호출 프로그램은 개체를 삭제해야 합니다(C++ - delete IAwrem32X*, VB ð set ObjectName = Nothing.).
boolean FileXferFromHost(BSTR HostFile);	매개 변수 = HostFile as string	호스트 시스템에 있는 파일을 원격지 시스템에 복사합니다. 매개 변수에 와일드 카드 문자를 포함할 수 있습니다. 매개 변수는 호스트 시스템에서 복사할 전체 경로 및 파일 이름을 포함합니다. HostFile과 RemoteFile 문자열이 일치할 필요는 없습니다.
boolean FileXferFromHost(BSTR RemoteFile);	매개 변수 = RemoteFile as string	호스트 시스템에 있는 파일을 원격지 시스템에 복사합니다. 매개 변수에 와일드 카드 문자를 포함할 수 있습니다. 매개 변수는 전체 대상 경로와 파일 이름을 포함합니다. HostFile과 RemoteFile 문자열이 일치할 필요는 없습니다.
boolean FileXferFromHost	반환 값 = Boolean	명령을 실행했으면 참
boolean FileXferToHost(BSTR HostFile);	매개 변수 = HostFile as string	원격지 시스템에 있는 파일을 호스트 시스템에 복사합니다. 매개 변수에 와일드 카드 문자를 포함할 수 있습니다. 매개 변수는 전체 대상 경로와 파일 이름을 포함합니다. HostFile과 RemoteFile 문자열이 일치할 필요는 없습니다.
boolean FileXferToHost(BSTR RemoteFile);	매개 변수 = RemoteFile as string	원격지 시스템에 있는 파일을 호스트 시스템에 복사합니다. 매개 변수에 와일드 카드 문자를 포함할 수 있습니다. 매개 변수는 원격지 시스템에서 복사할 전체 경로 및 파일 이름을 포함합니다. HostFile과 RemoteFile 문자열이 일치할 필요는 없습니다.

메소드	매개 변수 또는 반환 값	설명
<code>boolean FileXferToHost</code>	반환 값 = Boolean	명령을 실행했으면 참
<code>boolean CreateFolderOnHost(BSTR FolderName);</code>	매개 변수 = FolderName as string 반환 값 = Boolean	호스트 시스템에 새 폴더를 만듭니다. 이 함수는 원격지 시스템에 임시 폴더를 생성한 후 이를 호스트 시스템에 복사합니다. 매개 변수는 호스트 시스템에 폴더를 만들도록 드라이브와 경로를 포함합니다. 명령이 실행된 경우 반환 값은 TRUE입니다.
<code>boolean ExecuteHostFile(BSTR FileName);</code>	매개 변수 = FileName as string 반환 값 = Boolean	호스트 시스템에 있는 기존의 파일을 실행합니다. 이 함수는 배치, 명령, 실행 파일만 실행합니다. 실행 프로그램과 관련된 파일은 실행하지 않습니다. 예를 들어, .doc 파일을 실행했을 때 Microsoft Word를 열지 않습니다. 매개 변수는 호스트 시스템에 있는 파일의 전체 경로를 포함합니다. 명령이 실행된 경우 반환 값은 TRUE입니다.
<code>BSTR GetError();</code>	반환 값 = String	Awrem32에서 생성된 마지막 오류를 문자열로 반환합니다.
<code>short ConnectionStatus();</code>	반환 값 = Short	호스트 시스템으로 현재 사용자 연결 상태를 반환합니다. 사용할 수 있는 값은 다음과 같습니다. ■ -1 = 손실된 연결 ■ 0 = 연결 안됨 ■ 1 = 세션 연결됨

A

API 라이브러리 13
Awrem32
 함수 52, 79

C

C++.. Visual C++
CAPI 연결
 원격지 속성 28, 63
 호스트 속성 43, 74
CHostData
 Visual Basic 사용 34
 Visual C++ 사용 70
CHostDataEx
 Visual Basic 사용 44
 Visual C++ 사용 77
CHostDataManager
 Visual Basic 사용 32
 Visual C++ 사용 66
COM 장치
 원격지 장치 세부 사항 25, 62
 호스트 장치 세부 사항 40, 71
CRemoteData
 Visual Basic 사용 20
 Visual C++ 사용 59
CRemoteDataEx
 Visual Basic 사용 29
 Visual C++ 사용 65
CRemoteDataManager
 Visual Basic 사용 18
 Visual C++ 사용 56

G

GUID 10

I

ISDN
 CAPI 연결
 원격지 속성 28, 63
 호스트 속성 43, 74

N

NASI 장치 74
NCSI 장치 74
NetBIOS 연결
 원격지 속성 28, 63
 호스트 속성 43, 74

O

OLE 자동화.. pcAnywhere 자동화 서버

P

pcAnywhere 자동화 서버
 GUID 등록 10
 Visual Basic으로 액세스 12
 Visual C++로 액세스 12
 사용 실패 10
 정보 10
 형식 라이브러리 13

T

TAPI 장치
 원격지 61
 호스트 72

V

Visual Basic
 Awrem32 함수 52
 CHostData 개체 34
 CHostDataEx 개체 44
 CHostDataManager 개체 32
 CRemoteData 개체 20
 CRemoteDataEx 개체 29
 CRemoteDataManager 개체 18
 pcAnywhere 자동화 서버 액세스 12
 코드 샘플 30, 50
Visual C++
 Awrem32 함수 79
 CHostData 개체 70
 CHostDataEx 개체 77
 CHostDataManager 개체 66

Visual C++ (계속)

- CRemoteData 개체 59
- CRemoteDataEx 개체 65
- CRemoteDataManager 개체 56
- pcAnywhere 자동화 서버 액세스 12
- Winawsvr.h 포함 14
- 코드 샘플 65, 78
- 클래스 가져오기 13

W

Winawsvr

- 개체
 - CHostData 34, 70
 - CHostDataEx 44, 77
 - CHostDataManager 32, 66
 - CRemoteData 20, 59
 - CRemoteDataEx 29, 65
 - CRemoteDataManager 18, 56
- 실행 파일 11
- 헤더 파일 14

ㄱ

국가 코드

- 원격지 64
- 호스트 76

ㄴ

네트워크 연결

- 원격지
 - NetBIOS 속성 28, 63
- 호스트
 - NetBIOS 속성 43, 74

□

모뎀 연결

- 원격지
 - COM 속성 25, 62
 - TAPI 장치 61
 - 국가 코드 64
 - 전화 걸기 속성 24
- 호스트
 - COM 속성 40, 71
 - TAPI 장치 72
 - 국가 코드 76
 - 전화 걸기 속성 39

人

상태

- 반환 53, 81

- 식별자. GUID

○

암호

- 원격지 개체 65
- 호스트 개체 76

암호화 설정

- 원격지 29-30, 65
- 호스트 44, 77

연결

- 상태 반환 53, 81
- 시작 34, 52, 59, 80
- 오류 반환 81
- 종료 52, 80

연결 유형

- 원격지 속성 23, 61, 64
- 할당 38
- 할당 제거 39
- 호스트 속성 36, 71, 75

오류

- 반환 53, 81

원격지 개체

- 디렉터리 18, 56
- 메소드 64
- 삭제 58
- 암호 65
- 작성 58
- 장치 세부 사항 61
- 전화 걸기 속성 24
- 파일 19-20, 33, 56, 58

원격지 엔진

- 수동 등록 11
- 자동 등록 11

ㄷ

- 자동화 서버.. pcAnywhere 자동화 서버

자동화 제어기

- Visual Basic 정보 11

- Visual C++ 정보 12

- 장치.. 연결 유형

ㄴ

코드 샘플

- Visual Basic 30, 50
- Visual C++ 65, 78

클래스 정의
 가져오기 13
 보기 14

II

파일
 호스트에서 실행 53, 81
파일 전송
 호스트로 52, 80
 호스트에서 52, 80
패리티 25, 40, 62, 72
폴더
 호스트에 작성 53, 81

III

형식 라이브러리 13
호스트 개체
 디렉터리 33, 67
 메소드 75
 삭제 34, 69
 시작 34, 69
 암호 76
 작성 69
 장치 세부 사항 71
 전화 걸기 속성 39
 찾기 33-34, 67, 69
흐름 제어 25, 41, 62, 72