

Guide d'automatisation OLE

Guide d'automatisation OLE

Mentions légales

Copyright © 2008 Symantec Corporation. Tous droits réservés.

Symantec et le logo Symantec sont des marques commerciales ou déposées de Symantec Corporation ou de ses filiales, aux Etats-Unis et dans d'autres pays., Les autres noms peuvent être des marques de leurs détenteurs respectifs.

Ce produit Symantec peut contenir un logiciel tiers pour lequel Symantec est requis pour l'attribution à tiers ("Programmes tiers"). Certains programmes tiers sont disponibles sous licences de source ouverte ou de logiciel libre. le contrat de licence accompagnant le Logiciel ne modifie aucun droit ou engagement que vous pouvez avoir sous des licences de source ouverte ou de logiciel libre. Veuillez consulter l'annexe des mentions légales de cette documentation ou le fichier lisez-moi TPIP accompagnant ce produit Symantec pour plus d'informations sur les programmes tiers.

Le produit décrit dans ce document est distribué aux termes d'une licence limitant son utilisation, sa copie, sa distribution et sa décompilation/ingénierie inverse. Ce document ne peut, en tout ou partie, être reproduit sous aucune forme et par aucun moyen sans l'autorisation préalable écrite de Symantec Corporation et de ses détenteurs de licence éventuels.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT", ET TOUTE GARANTIE OU CONDITION D'AUCUNE SORTE, EXPRESSE OU IMPLICITE, Y COMPRIS, SANS QUE CELA SOIT LIMITATIF, LES GARANTIES OU CONDITIONS IMPLICITES DE QUALITE MARCHANDE, D'ADEQUATION A UN USAGE PARTICULIER OU DE RESPECT DES DROITS DE PROPRIETE INTELLECTUELLE EST TENUE POUR JURIDIQUEMENT NON VALIDE. SYMANTEC CORPORATION NE POURRA PAS ETRE TENUE RESPONSABLE DE DOMMAGES DIRECTS OU INDIRECTS EN RELATION AVEC LE CONTENU OU L'UTILISATION DE LA PRESENTE DOCUMENTATION. LES INFORMATIONS PRESENTES DANS CETTE DOCUMENTATION SONT SUJETTES A MODIFICATION SANS PREAVIS.

Le logiciel et la documentation sous licence sont considérés "logiciel informatique commercial" et "documentation de logiciel informatique commercial" t sujets aux droits restreints selon les définitions des sections FAR 12.212 et DFARS 227.7202 autant qu'applicables, ainsi que toute régulation issue de celles-ci. Toute utilisation, modification, reproduction de version, performance, affichage ou divulgation du logiciel et de la documentation sous licence, par le gouvernement des Etats-Unis, ne peut s'effectuer que selon les termes de cet accord.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

<http://www.symantec.fr>

Fabriqué en Irlande.

10 9 8 7 6 5 4 3 2 1

Support technique

Le support technique de Symantec gère les centres de support au niveau mondial. Le support technique a pour rôle essentiel de répondre à des demandes spécifiques sur les caractéristiques et les fonctions des produits. Il est également responsable de la création des articles de notre base de données en ligne. Le groupe de support technique collabore avec les autres entités de Symantec pour répondre rapidement à vos questions. Il travaille par exemple avec le département Ingénierie et Symantec Security Response pour fournir des services d'alerte et des mises à jour des définitions de virus.

Parmi les offres de maintenance de Symantec, il faut citer :

- Une gamme d'options de support pour vous permettre de sélectionner le niveau de service adapté à votre entreprise.
- Un support par téléphone et sur le Web apportant des réponses rapides et des informations de dernière minute.
- Une assurance Mise à niveau fournissant une protection de la mise à niveau automatique de logiciels.
- Un support global de 24 heures sur 24 et 7 jours sur 7.
- Des prestations évoluées telles que la gestion de compte de service.

Pour plus d'informations sur les programmes de maintenance proposés par Symantec, visitez notre site Web à l'adresse suivante :

www.symantec.com/techsupp/

Entrer en contact avec le support technique

Les clients avec un contrat de maintenance en cours peuvent accéder à des informations de support technique à l'URL suivante :

www.symantec.com/techsupp/

Avant d'entrer en contact avec le support technique, assurez-vous d'avoir satisfait la configuration requise indiquée dans votre documentation produit. En outre, vous devriez être à l'ordinateur sur lequel le problème s'est produit, au cas où il serait nécessaire de recréer le problème.

Quand vous entrez en contact avec le support technique, ayez s'il vous plaît les informations suivantes disponibles :

- Niveau de version du produit
- Informations concernant le matériel
- Mémoire disponible, espace disque, informations NIC

- Système d'exploitation
- Version et niveau de correctif
- Topologie réseau
- Routeur, passerelle, et informations d'adresse IP
- Description du problème :
 - Messages d'erreur et fichiers journaux
 - Dépannage effectué avant d'entrer en contact avec Symantec
 - Changements de configuration récents du logiciel et changements de réseau

Licence et enregistrement

Si votre produit Symantec requiert un enregistrement ou une clé de licence, visitez la page Web de notre support technique à l'adresse suivante :

www.symantec.com/techsupp/

Service clientèle

Les informations de Service clientèle sont disponibles à l'URL suivante :

www.symantec.com/techsupp/

Le Service clientèle peut vous aider à résoudre des problèmes des types suivants :

- licence et numéros de série des produits ;
- mise à jour des enregistrements de produit en cas de changement de nom ou d'adresse ;
- informations générales sur les produits (caractéristiques, langues disponibles, distributeurs) ;
- dernières informations sur les mises à jour et mises à niveau des produits ;
- informations sur les assurances de mise à jour et les contrats de maintenance ;
- informations au sujet de l'achat de programmes Symantec ;
- conseil sur les options de support technique de Symantec ;
- questions avant-vente non techniques ;
- questions concernant les CD-ROM ou les manuels.

Ressources des accords de maintenance

Pour toute information sur les accords de maintenance, veuillez contacter l'équipe administrative responsable de votre région :

Asie-Pacifique et Japon	contractsadmin@symantec.com
Europe, Moyen-Orient et Afrique	semea@symantec.com
Amérique du Nord et Amérique latine	supportolutions@symantec.com

Services supplémentaires destinés aux entreprises

Symantec propose un ensemble complet de services qui vous permettront de rentabiliser au maximum vos produits Symantec, de développer vos compétences et d'avoir une meilleure vue d'ensemble pour gérer plus efficacement les risques auxquels votre entreprise doit faire face.

Les services destinés aux entreprises sont les suivants :

Symantec Early Warning Solutions	Ces solutions permettent que vous soyez prévenu précocement en cas de cyber-attaque et vous appuyer sur une analyse exhaustive des menaces et des contre-mesures pour prévenir les attaques avant qu'elles ne frappent.
Services de supervision de la sécurité	Ces services évitent l'obligation de gérer et superviser les périphériques et les événements de sécurité, garantissant ainsi une réponse rapide aux menaces réelles.
Services de conseil	Symantec Consulting Services garantissent une expertise technique sur site apportée directement par Symantec ou ses partenaires agréés. Symantec Consulting Services propose une série d'options préconçues et personnalisables qui incluent des fonctions d'évaluation, de conception, de mise en oeuvre, de surveillance et de gestion. Chacune porte sur l'établissement et la mise à jour de l'intégrité et de la disponibilité de vos ressources informatiques.
Services de formation	Les services de formations fournissent une gamme complète de programmes de formation technique, de formation à la sécurité, de certification de sécurité et de communication de sensibilisation.

Pour accéder à des informations supplémentaires concernant les services d'entreprise, visitez notre site Web à l'adresse :

www.symantec.com

Sélectionnez le pays et la langue de votre choix dans l'index du site.

Table des matières

Support technique	4	
Chapitre 1	Utilisation de l'automatisation OLE avec Symantec pcAnywhere	9
	A propos de l'automatisation OLE	9
	A propos du serveur d'automatisation pcAnywhere	10
	Ce que vous pouvez faire avec le serveur d'automatisation pcAnywhere	10
	Avant de commencer	11
	Inscription automatique du moteur distant	11
	Inscription manuelle du moteur distant	12
	Accès au serveur d'automatisation pcAnywhere	12
	Accès au serveur d'automatisation pcAnywhere avec Visual Basic	12
	Accès au serveur d'automatisation pcAnywhere avec Visual C++	14
	Démarrage des objets OLE hôte et distant	16
	Pour plus d'informations	16
Chapitre 2	Définitions d'objets Visual Basic	17
	A propos des objets Visual Basic	17
	Méthodes CRemoteDataManager	18
	Propriétés CRemoteData	21
	Propriétés du type de connexion	23
	Propriétés de numérotation	25
	Propriétés de périphérique COM	26
	Propriétés de périphérique NetBIOS	29
	Propriétés de périphérique RNIS via CAPI 2.0	30
	objet CRemoteDataEx	31
	Exemple de code Visual Basic pour la fonction distant	32
	Méthodes CHostDataManager	34
	Propriétés CHostData	37
	Propriétés du type de connexion	39
	Méthode AssignConnection(type de connexion)	42
	Méthode UnassignConnection (type de connexion)	42

Propriétés de numérotation	43
Propriétés de périphérique COM	44
Propriétés de périphérique NetBIOS	47
Propriétés de périphérique RNIS via CAPI 2.0	48
objet CHostDataEx	48
Exemple de code Visual Basic pour la fonctionnalité hôte.	56
fonctions awrem32	58
Chapitre 3	Définitions d'objets Visual C++
	61
A propos des objets Visual C++	61
Méthodes CRemoteDataManager	62
objet CRemoteData	66
Méthodes Get et Set (obtenir et définir)	66
Méthodes de détail d'objet distant	68
Méthodes d'objet distant	71
objet CRemoteDataEx	72
Exemple de code Visual C++ pour la fonctionnalité du	
distant	72
Méthodes CHostDataManager	73
objet CHostData	77
Méthodes Get et Set (obtenir et définir)	77
Méthodes de détail d'objet hôte	79
Méthodes d'objet hôte	83
objet CHostDataEx	85
Exemple de code Visual C++ pour la fonctionnalité Hôte	87
fonctions awrem32	88
Index	91

Utilisation de l'automatisation OLE avec Symantec pcAnywhere

Ce chapitre traite des sujets suivants :

- [A propos de l'automatisation OLE](#)
- [A propos du serveur d'automatisation pcAnywhere](#)
- [Ce que vous pouvez faire avec le serveur d'automatisation pcAnywhere](#)
- [Avant de commencer](#)
- [Pour plus d'informations](#)

A propos de l'automatisation OLE

L'automatisation OLE est une technologie qui permet de créer une application externe ou un autre outil de développement (par exemple un script ou une macro) capable de contrôler et d'automatiser toute fonction exposée d'une application.

L'automatisation OLE est constituée des composants suivants :

un serveur d'automatisation OLE ;	une application ou composant logiciel qui expose sa fonction afin d'être accessible ou de pouvoir être contrôlé par d'autres applications ou d'autres outils de développement.
-----------------------------------	--

Le serveur d'automatisation pcAnywhere est un exemple de serveur d'automatisation OLE ;

un contrôleur
d'automatisation OLE ;

une application ou outil de développement pouvant
atteindre et contrôler les composants exposés par le serveur
d'automatisation OLE.

Vous pouvez utiliser tout langage de programmation
prenant en charge l'automatisation OLE. Les deux langages
de programmation les plus courants sont Microsoft Visual
Basic et Microsoft Visual C++.

Une application externe accède à un serveur d'automatisation en se connectant
au serveur, puis en demandant l'accès à une ou plusieurs de ses interfaces publiées.
Une interface est un point d'entrée permettant d'accéder à une ou plusieurs
méthodes ou propriétés rattachées. Une fois que l'application a obtenu une
interface vers le serveur, elle peut appeler toute méthode d'interface comme si
celle-ci faisait partie de l'application externe.

A propos du serveur d'automatisation pcAnywhere

Le serveur d'automatisation pcAnywhere permet aux applications externes de
gérer les fichiers pcAnywhere Hôte et Distant afin d'automatiser les tâches de
téléintervention et de transfert de fichiers. Le serveur d'automatisation
pcAnywhere fonctionne comme un substitut programmable de l'interface
utilisateur de Symantec pcAnywhere et reprend dans une large mesure son
comportement par défaut.

Par exemple, lorsque vous créez un objet hôte dans pcAnywhere, le premier modem
TAPI disponible est attribué par défaut. De même, si vous créez un objet hôte à
l'aide du serveur d'automatisation pcAnywhere, puis que vous énumérez la liste
des connexions attribuées, le premier modem TAPI disponible est déjà attribué.

Ce que vous pouvez faire avec le serveur d'automatisation pcAnywhere

Le serveur d'automatisation pcAnywhere vous permet d'automatiser diverses
tâches administratives et de productivité.

Vous pouvez utiliser serveur d'automatisation pcAnywhere pour effectuer les
actions suivantes :

- Distribution et installation automatiques de mises à jour logicielles sur
plusieurs ordinateurs du réseau
- Programmation de transferts automatiques de fichiers entre ordinateurs à des
fins d'audit ou d'archivage

- Ajouter ou supprimer automatiquement un nom de la liste des appelants sur chaque hôte pcAnywhere de votre réseau

Ce document comporte plusieurs exemples, écrits à la fois en Visual Basic et en Visual C++, afin d'illustrer la manière de se connecter au serveur d'automatisation pcAnywhere et de l'utiliser.

Se reporter à ["A propos des objets Visual Basic"](#) à la page 17.

Se reporter à ["A propos des objets Visual C++"](#) à la page 61.

Avant de commencer

Au cours d'une connexion au serveur d'automatisation pcAnywhere et à ses interfaces, les paramètres d'identification, appelés GUID (Globally Unique Identifiers - identificateurs globalement uniques), sont transmis aux fonctions API de la bibliothèque d'automatisation. Un GUID distinct est attribué au serveur d'automatisation pcAnywhere et à chaque interface exposée. Ces GUID doivent être présents dans le registre système pour connecter une application externe au serveur d'automatisation pcAnywhere et à ses interfaces.

Si vous exécutez l'application externe sur un ordinateur sur lequel Symantec pcAnywhere est installé, vous pouvez inscrire automatiquement les entrées GUID. Dans le cas contraire, vous devez enregistrer les entrées GUID manuellement.

Se reporter à ["Inscription automatique du moteur distant"](#) à la page 11.

Se reporter à ["Inscription manuelle du moteur distant"](#) à la page 12.

Inscription automatique du moteur distant

Avant de vous connecter à un autre ordinateur pour la première fois en utilisant votre client OLE, vous devez inscrire automatiquement le moteur distant. Cette opération peut s'effectuer automatiquement lorsque vous démarrez un objet distant dans Symantec pcAnywhere.

Pour inscrire automatiquement le moteur distant

- 1 Pour ouvrir Symantec pcAnywhere, effectuez l'une des opérations suivantes :
 - Sur le bureau, cliquez deux fois sur l'icône du programme Symantec pcAnywhere.

- Dans la barre des tâches de Windows, cliquez sur **Démarrer> Programmes> Symantec pcAnywhere**.
- 2 Dans la fenêtre du Gestionnaire pcAnywhere, cliquez sur **Distants**.
- 3 Cliquez deux fois sur un élément de connexion distante.

Cette procédure inscrit le moteur distant. Il n'est pas nécessaire de terminer la connexion.

Inscription manuelle du moteur distant

Si pcAnywhere n'est pas installé sur l'ordinateur sur lequel vous exécutez l'application externe, vous devez inscrire manuellement les GUID en exécutant le fichier exécutable (winawsvr.exe) du serveur d'automatisation pcAnywhere. Il suffit de lancer l'exécutable une seule fois pour ajouter les GUID au registre. Le fichier Winawsvr.exe se trouve dans le répertoire d'installation.

Accès au serveur d'automatisation pcAnywhere

Vous pouvez accéder au serveur d'automatisation de pcAnywhere en utilisant toute plate-forme prenant en charge l'automatisation OLE. Les deux plates-formes les plus répandues qui prennent en charge l'automatisation OLE sont Visual Basic et Visual C++.

Les principes de codage de ces deux plates-formes sont similaires, bien que dans l'environnement Visual Basic une part importante des tâches de bas niveau est exécutée en arrière-plan par l'exécutable Visual Basic.

Accès au serveur d'automatisation pcAnywhere avec Visual Basic

Le langage de programmation Visual Basic intègre la prise en charge de l'interaction avec les serveurs d'automatisation OLE tels que le serveur d'automatisation pcAnywhere. Vous pouvez créer un projet Exe standard puis, dans chaque méthode, entrer le code pour accéder au serveur d'automatisation pcAnywhere. Visual Basic prend les appels de méthode de haut niveau dans les fichiers source et les développe de façon interne dans les appels de méthode de bas niveau correspondants de l'automatisation OLE.

Pour accéder au serveur d'automatisation pcAnywhere avec Visual Basic

- 1 Ajoutez une paire de variables objet pour chaque objet pcAnywhere auquel accéder.

Par exemple, pour travailler avec des objets distant, dimensionnez un objet **RemoteDataManager** et **RemoteDataObject**.

- 2 Utilisez RemoteDataManager pour le rattachement au gestionnaire de données de l'objet distant.

Par exemple, appelez la méthode CreateObject avec **WINAWSVR.REMOTEDATAMANAGER** comme paramètre.

Visual Basic utilise le paramètre textuel pour repérer l'identificateur du gestionnaire dans la base de registres et renvoie l'interface vers ce gestionnaire.

- 3 Une fois qu'il existe un objet gestionnaire de données valide, utilisez-le pour effectuer les opérations suivantes :

- Déterminer le répertoire en cours
- Accéder à un autre répertoire
- Enumérer les fichiers objet de données associés dans le répertoire en cours
- Créer, extraire ou supprimer un fichier objet de données

- 4 Après la création ou la récupération d'un objet de données, vous pouvez obtenir (Get) ou définir (Set) ses propriétés.

La syntaxe de Visual Basic n'utilise pas de nom de propriété pour différencier l'obtention et la définition de sa valeur. C'est la position de la propriété par rapport à l'opérateur d'affectation qui détermine si l'appel de méthode sous-jacent est un Get ou un Set.

Les exemples suivants illustrent un Get et un Set :

- Pour obtenir la valeur du numéro de téléphone d'un objet, placez le nom de la propriété à droite de l'opérateur d'affectation.
Par exemple, `s = RemoteData.PhoneNumber()`, où `s` est une variable de type chaîne.
- Pour configurer le numéro de téléphone, placez le nom de la propriété à gauche de l'opérateur d'affectation.
Par exemple, `RemoteData.PhoneNumber = "555-1212"`

Accès au serveur d'automatisation pcAnywhere avec Visual C++

Le serveur d'automatisation pcAnywhere utilise des bibliothèques de type pour exposer les informations concernant ses interfaces et ses méthodes vers les clients d'automatisation, qui sont écrits en Visual C++. Ces bibliothèques de type utilisent des MFC (Microsoft Foundation Classes), qui peuvent être importées dans votre application via l'assistant ClassWizard de Visual C++.

Les classes du gestionnaire de données fournies dans les bibliothèques de type contiennent la fonction nécessaire pour obtenir une interface vers le serveur d'automatisation pcAnywhere et effectuer des opérations de haut niveau sur le type d'objet associé à l'interface.

Utiliser l'objet gestionnaire de données pour effectuer les opérations suivantes :

- Déterminer ou modifier le répertoire en cours.
- Enumérer les fichiers objet de données du répertoire en cours.
- Créer, extraire ou supprimer un objet nommé.

Une fois créé ou récupéré, un objet utilise la classe d'objet de données associée pour examiner ou modifier ses propriétés exposées. La plupart de ces propriétés sont exposées à travers une paire de méthodes qui commencent par le mot Get ou Set. Par exemple, un utilisateur appelle la méthode GetPhoneNumber pour examiner la propriété de numéro de téléphone en cours de l'objet et appelle SetPhoneNumber pour la définir.

Importation et affichage des classes

Le serveur d'automatisation pcAnywhere utilise les bibliothèques de type suivantes :

Winawsvr.tlb	Contient les informations nécessaires pour se connecter au serveur d'automatisation pcAnywhere et accéder à ses interfaces.
Awrem32.tlb	Contient les informations nécessaires au contrôle des connexions pcAnywhere.

Les procédures suivantes expliquent comment importer les bibliothèques de type du serveur d'automatisation pcAnywhere et afficher les classes ayant été ajoutées à votre application.

Pour importer des classes

- 1 Créez une application MFC dans Visual C++.
- 2 Dans le menu Affichage, cliquez sur **Assistant de classe**.

- 3 Dans la boîte de dialogue Assistant de classe, cliquez sur **Ajouter une classe**, puis cliquez sur **Depuis une bibliothèque de type**.
- 4 Cliquez deux fois sur **winawsvr.tlb**.
- 5 Dans la boîte de dialogue Confirmer Classes, cliquez sur **OK** pour importer toutes les définitions de classe.
- 6 Dans la boîte de dialogue Assistant de classe, cliquez sur **Ajouter une classe**, puis cliquez sur **Depuis une bibliothèque de type**.
- 7 Cliquez deux fois sur **awrem32.tlb**.
- 8 Dans la boîte de dialogue Confirmer Classes, cliquez sur **OK** pour importer toutes les définitions de classe.
- 9 Dans la boîte de dialogue Assistant de classe, cliquez sur **OK** pour terminer l'importation.

Les classes sont ajoutées à l'application. Elles vous permettent de manipuler des objets et de gérer des connexions.

L'importation des définitions de classe à partir des bibliothèques de type ajoute également des fichiers de prise en charge à l'application. Ces fichiers contiennent les définitions de classe et le code source de mise en oeuvre du serveur d'automatisation pcAnywhere.

Se reporter à "[Affichage des définitions de classe et des fichiers de mise en oeuvre](#)" à la page 15.

Pour afficher les classes ajoutées

- 1 Dans Visual C++, ouvrez votre application MFC.
- 2 Dans la fenêtre de l'espace de travail, cliquez sur l'onglet **ClassView**.

Affichage des définitions de classe et des fichiers de mise en oeuvre

Lorsque vous importez les bibliothèques de type du serveur d'automatisation pcAnywhere dans votre application, les fichiers suivants sont ajoutés :

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

Ces fichiers contiennent les définitions de classe et le code source de mise en oeuvre du serveur d'automatisation pcAnywhere. Il est inutile d'éditer ces fichiers ;

cependant, chaque fichier source de l'application contenant des appels des méthodes du serveur d'automatisation pcAnywhere doit inclure Winawsvr.h.

Pour afficher les définitions de classe et les fichiers de mise en oeuvre

- 1 Dans Visual C++, ouvrez votre application MFC.
- 2 Dans la fenêtre de l'espace de travail, cliquez sur l'onglet **FileView**.

Démarrage des objets OLE hôte et distant

Symantec pcAnywhere nécessite que tous les objets hôte et distant soient stockés dans le répertoire de données par défaut. Avant de démarrer un objet hôte ou distant que vous avez créé avec OLE, vérifiez qu'il se trouve dans le répertoire de données par défaut de pcAnywhere. Le répertoire par défaut est le dossier All Users dans Windows XP, et le dossier ProgramData dans Vista.

Pour plus d'informations

Pour plus d'informations sur l'automatisation OLE, reportez-vous aux documents de référence suivants (en anglais) :

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

Définitions d'objets Visual Basic

Ce chapitre traite des sujets suivants :

- [A propos des objets Visual Basic](#)
- [Méthodes CRemoteDataManager](#)
- [Propriétés CRemoteData](#)
- [objet CRemoteDataEx](#)
- [Méthodes CHostDataManager](#)
- [Propriétés CHostData](#)
- [objet CHostDataEx](#)
- [fonctions awrem32](#)

A propos des objets Visual Basic

Le serveur d'automatisation pcAnywhere offre les composants suivants pour la prise en charge de l'automatisation OLE :

Winawsvr	contient les informations nécessaires pour se connecter au serveur d'automatisation pcAnywhere et accéder à ses interfaces.
Awrem32	Contient les informations nécessaires au contrôle des connexions pcAnywhere.

Le [Tableau 2-1](#) décrit les objets qui composent Winawsvr.

Tableau 2-1 Objets Winawsvr

Objet	Description	Référence
CRemoteDataManager	Fournit les méthodes pour créer, ouvrir, modifier, enregistrer et supprimer les objets CRemoteData.	Se reporter à " Méthodes CRemoteDataManager " à la page 18.
CRemoteData	Définit les paramètres permettant d'accéder à la fonction distant de pcAnywhere et de la contrôler.	Se reporter à " Propriétés CRemoteData " à la page 21. Se reporter à " objet CRemoteDataEx " à la page 31.
CHostDataManager	Fournit les méthodes pour créer, ouvrir, modifier, enregistrer et supprimer les objets ChostData.	Se reporter à " Méthodes CHostDataManager " à la page 34.
CHostData	Définit les paramètres permettant d'accéder à la fonction hôte de pcAnywhere et de la contrôler.	Se reporter à " Propriétés CHostData " à la page 37. Se reporter à " objet CHostDataEx " à la page 48.

Awrem32 possède un objet, constitué de huit interfaces, pour prendre en charge les sessions de téléintervention et de transfert de fichiers.

Se reporter à "[fonctions awrem32](#)" à la page 58.

Quelques fonctions ne sont plus prises en charge. Cependant, les définitions d'objets sont utilisables avec les versions antérieures de pcAnywhere.

Pour les fonctions nécessitant des mots de passe, les valeurs des mots de passe peuvent être définies mais non récupérées, pour des raisons de sécurité.

Méthodes CRemoteDataManager

Les méthodes CRemoteDataManager fournissent les paramètres et renvoient les valeurs permettant d'accéder aux objets CRemoteData et de les contrôler.

Tableau 2-2 Méthodes CRemoteDataManager

Méthode CRemoteDataManager	Paramètre ou valeur renvoyés	Description
CurrentDirectory()	Valeur renvoyée = chaîne	Renvoie le chemin d'accès complet du dossier actuel dans lequel les objets distants pcAnywhere sont stockés.

Méthode CRemoteDataManager	Paramètre ou valeur renvoyés	Description
ChangeDirectory(nouveau répertoire)	Paramètre = NewDirectory	Change le dossier actuel dans lequel les objets distants pcAnywhere sont stockés.
FindFirst(schéma)	Paramètre = schéma en tant que chaîne	Trouve le premier fichier d'objet distant de pcAnywhere (*.chf) dans le dossier en cours ; classe les noms de fichiers de schémas sur base du schéma spécifié de nom de fichier (un astérisque [*] permet de trouver tous les fichiers du dossier en cours)
FindFirst(nom)	Paramètre = Nom comme chaîne	Tampon de retour pour le chemin d'accès complet du fichier d'objet distant (*.chf) correspondant au schéma spécifié
FindNext(nom)	Paramètre = Nom comme chaîne	Tampon de retour pour le chemin d'accès complet du fichier d'objet distant correspondant au schéma spécifié dans l'appel initial de FindFirst().
FindNext(nom)	Valeur renvoyée = booléen	VRAI si un autre fichier d'objet distant correspondant au schéma spécifié dans l'appel de FindFirst() est trouvé. Le chemin d'accès complet du fichier est stocké dans Nom.
RetrieveObject(nom)	Paramètre = Nom comme chaîne	Nom intégralement qualifié du fichier d'objet distant à charger.
RetrieveObject(mode d'accès)	Paramètre = Mode d'accès comme valeur entière	Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes : <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
RetrieveObject(mot de passe)	Paramètre = Mot de passe comme chaîne	Mot de passe de l'objet. Peut être NULL.

Méthode CRemoteDataManager	Paramètre ou valeur renvoyés	Description
RetrieveObjectEx(nom)	Paramètre = Nom comme chaîne	Nom intégralement qualifié du fichier d'objet distant à charger.
RetrieveObjectEx(mode d'accès)	Paramètre = Mode d'accès comme valeur entière	Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes : <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
RetrieveObjectEx(mot de passe)	Paramètre = Mot de passe comme chaîne	Mot de passe de l'objet. Peut être NULL.
RetrieveObjectEx	Valeur renvoyée = Objet	Objet CRemoteDataEx du fichier spécifié.
CreateObject(nom)	Paramètre = Nom comme chaîne	Crée un objet CRemoteData et renvoie un pointeur LPDISPATCH qui le désigne. Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet.
CreateObject	Valeur renvoyée = Objet	CRemoteData
CreateObjectEx(nom)	Paramètre = Nom comme chaîne	Crée un objet CRemoteDataEx et renvoie un pointeur LPDISPATCH qui le désigne. Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet.
CreateObjectEx	Valeur renvoyée = Objet	CRemoteDataEx
DeleteObject(nom)	Paramètre = Nom comme chaîne	Nom intégralement qualifié du fichier d'objet distant de l'objet à supprimer.
DeleteObject(mot de passe)	Paramètre = Mot de passe comme chaîne	Mot de passe de l'objet
DeleteObject	Valeur renvoyée = booléen	VRAI si l'objet est supprimé.

Propriétés CRemoteData

Les propriétés et les paramètres disponibles pour l'objet CRemoteData sont affichés comme suit.

Dans [Tableau 2-3](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-3 CRemoteData - Propriétés et paramètres

Propriété	Paramètre	Description
<CRemoteData>.ComputerName(chaine)	Chaîne	Définit le nom ou l'adresse IP de l'ordinateur hôte.
<CRemoteData>.PhoneNumber(chaine)	Chaîne	Définit le numéro de téléphone de l'ordinateur hôte.
<CRemoteData>.UseDialingProperties(Bool)	Bool	Définit les propriétés de numérotation du système.
<CRemoteData>.RedialCount(entier)	Entier	Définit le nombre de tentatives de rappel avant l'annulation de l'appel.
<CRemoteData>.RedialDelay(entier)	Entier	Définit le nombre de secondes d'attente entre les tentatives de rappel.
<CRemoteData>.AutoLoginName(chaine)	Chaîne	Définit le nom de l'utilisateur pour la connexion automatique. Pour plus d'informations sur l'utilisation de la connexion aux domaines, reportez-vous à la section : Se reporter à " objet CRemoteDataEx " à la page 31.
<CRemoteData>.AutoLoginPassword(chaine)	Chaîne	Définit le mot de passe pour la connexion automatique dans l'objet distant. Pour des raisons de sécurité, le serveur d'automatisation pcAnywhere ne permet pas de lire la valeur du mot de passe. Aucune valeur de mot de passe n'est renvoyée.

Propriété	Paramètre	Description
<CRemoteData>.Password(chaine)	Chaîne	Définit le mot de passe sur l'objet distant, à utiliser avec les méthodes ExecuteProtection, ReadProtection et WriteProtection. Pour des raisons de sécurité, le serveur d'automatisation pcAnywhere ne permet pas de lire la valeur du mot de passe. Aucune valeur de mot de passe n'est renvoyée.
<CRemoteData>.ExecuteProtection (Bool)	Bool	Définit la nécessité d'un mot de passe pour exécuter l'objet. Définir par mot de passe.
<CRemoteData>.ReadProtection(Bool)	Bool	Définit la nécessité d'un mot de passe pour afficher les propriétés de l'objet distant. Définir par mot de passe.
<CRemoteData>.WriteProtection(Bool)	Bool	Définit la nécessité d'un mot de passe pour enregistrer les modification de l'objet distant. Définir par mot de passe.
<CRemoteData>.LogSession (Bool)	Bool	Active et désactive la consignation des sessions.
<CRemoteData>.RecordFile(chaine)	Chaîne	Définit le nom et le chemin d'accès intégralement qualifié du fichier d'enregistrement de la session active.
<CRemoteData>.RecordSession(Bool)	Bool	Active et désactive la consignation automatique de session.
<CRemoteData>.ReadObject(chaine)	Chaîne	Définit le mot de passe de l'objet. Utilisez cette propriété pour actualiser la copie locale des données de l'objet distant.
<CRemoteData>.WriteObject(chaine)	Chaîne	Définit le mot de passe de l'objet. Utilisez cette propriété pour créer l'objet distant ou pour écrire les changements apportés à celui-ci.

Les propriétés et les valeurs renvoyées pour CRemoteData sont affichées dans le tableau suivant.

Dans [Tableau 2-4](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-4 CRemoteData - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CRemoteData>.ComputerName	Chaîne	Renvoie le nom ou l'adresse IP de l'ordinateur hôte
String = <CRemoteData>.PhoneNumber	Chaîne	Renvoie le numéro de téléphone de l'ordinateur hôte
Bool = <CRemoteData>.UseDialingProperties	Bool	Renvoie les propriétés de numérotation du système définies dans l'objet distant
Integer = <CRemoteData>.RedialCount	Entier	Renvoie le nombre de tentatives d'appel défini dans l'objet distant
Integer = <CRemoteData>.RedialDelay	Entier	Renvoie le nombre de secondes d'attente entre les tentatives de rappel.
String = <CRemoteData>.AutoLoginName	Chaîne	Renvoie le nom de l'utilisateur pour la connexion automatique
Bool = <CRemoteData>.ExecuteProtection	Bool	Renvoie la valeur du paramètre ExecuteProtection.
Bool = <CRemoteData>.ReadProtection	Bool	Renvoie la valeur du paramètre ReadProtection.
Bool = <CRemoteData>.WriteProtection	Bool	Renvoie la valeur du paramètre WriteProtection
Bool = <CRemoteData>.LogSession	Bool	VRAI si la consignation de session est activée.
String = <CRemoteData>.RecordFile	Chaîne	Renvoie le nom et le chemin d'accès intégralement qualifié du fichier d'enregistrement de la session.
Bool = <CRemoteData>.RecordSession	Bool	Renvoie la valeur du paramètre d'enregistrement de session.

Propriétés du type de connexion

Le tableau suivant décrit les propriétés et les paramètres du type de connexion.

Dans [Tableau 2-5](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-5 Type de connexion - Propriétés et paramètres

Propriété	Paramètre	Description
<CRemoteData>.ConnectionType(chaine)	Chaîne	<p>Définit le type de connexion de l'ordinateur distant. La valeur transmise doit être un type de connexion valide, défini par les fonctions FirstConnectionType() et NextConnectionType().</p> <p>Les types suivants sont des exemples de connexions valides :</p> <ul style="list-style-type: none"> ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ RNIS via CAPI 2.0 ■ Nom du modem (tel qu'il apparaît sur l'ordinateur)

Le tableau suivant décrit les propriétés et les valeurs renvoyées pour le type de connexion.

Dans [Tableau 2-6](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-6 Type de connexion - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CRemoteData>.ConnectionTypes	Chaîne	Renvoie le type de connexion de l'objet distant.

Propriété	Valeur renvoyée	Description
Integer = <CRemoteData>.ConnectionType	Entier	Renvoie le nombre de types de connexion disponibles. Les types suivants sont des exemples de connexions valides : <ul style="list-style-type: none"> ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ RNIS via CAPI 2.0 ■ Nom du modem (tel qu'il apparaît sur l'ordinateur)
String = <CRemoteData>.FirstConnectionType	Chaîne	Renvoie le premier type de connexion disponible.
String = <CRemoteData>.NextConnectionType	Chaîne	Renvoie le type de connexion disponible suivant. L'appel s'effectue de façon séquentielle pour le nombre de types de connexion défini dans <CRemoteData>.ConnectionType, afin d'énumérer tous les types de connexion.
Bool = <CRemoteData>.FindConnectionType(type de connexion)	Bool	VRAI si le type de connexion indiqué a été trouvé dans la liste des types de connexion disponibles.

Propriétés de numérotation

Le tableau suivant décrit les propriétés et les paramètres pour définir les propriétés de numérotation pour les connexions par modem.

Dans [Tableau 2-7](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-7 Propriétés de numérotation - Propriétés et paramètres

Propriété	Paramètre	Description
<CRemoteData>.AreaCode(chaine)	Chaîne	Définit les propriétés de numérotation de code de zone pour les connexions par modem
<CRemoteData>.CountryCode(chaine)	Chaîne	Définit les propriétés de numérotation d'indicatif international pour les connexions par modem

Le tableau suivant définit les propriétés et les valeurs renvoyées pour les propriétés de numérotation par modem.

Dans [Tableau 2-8](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-8 Propriétés de numérotation - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CRemoteData>.AreaCode	Chaîne	Renvoie les propriétés de numérotation de code de zone
String = <CRemoteData>.CountryCode	Chaîne	Renvoie les propriétés de numérotation d'indicatif international
Integer = <CRemoteData>.CountryCodes	Entier	Renvoie le nombre d'indicatifs internationaux disponibles.
String = <CRemoteData>.FirstCountryCode	Chaîne	Renvoie le premier indicatif de pays apparaissant dans la liste du système d'exploitation.
String = <CRemoteData>.NextCountryCode	Chaîne	Renvoie l'indicatif international disponible suivant. L'appel s'effectue de façon séquentielle pour le nombre d'indicatifs internationaux défini dans <CRemoteData>.CountryCodes, afin d'énumérer tous les indicatifs internationaux.

Propriétés de périphérique COM

Le tableau suivant définit les propriétés et les paramètres qui vous permettent de personnaliser les paramètres de port du modem et des autres connexions COM.

Dans [Tableau 2-9](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-9 Périphérique COM - Propriétés et paramètre

Propriété	Paramètre	Description
<CRemoteData>.ComParity(chaine)	Chaîne	Définit la parité des communications. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ <Aucun> ■ Impaire ■ Paire ■ Marque ■ Espace
<CRemoteData>.ComFlowControl(chaine)	Chaîne	Définit le contrôle de flux des types de connexion COM. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ <Aucune> ■ XONXOFF ■ RTS/CTS ■ Les deux
<CRemoteData>.ComStartedBy(chaine)	Chaîne	Définit le paramètre de début des types de connexion COM. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI) ■ Réception 2 <RC> ■ Réponse du modem
<CRemoteData>.ComEndedBy(chaine)	Chaîne	Définit le paramètre de fin des types de connexion COM. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI)

Propriété	Paramètre	Description
<CRemoteData>.ComSpeed(Long)	Long	Contient le paramètre de vitesse COM maximale Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Le tableau suivant décrit les propriétés et les valeurs renvoyées par le périphérique COM.

Dans [Tableau 2-10](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-10 Périphériques COM - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CRemoteData>.ComParity	Chaîne	Renvoie l'une des valeurs suivantes comme parité de communications : <ul style="list-style-type: none"> ■ <Aucune> ■ Impaire ■ Paire ■ Marque ■ Espace
String = <CRemoteData>.ComFlowControl	Chaîne	Renvoie le paramètre de contrôle de flux de l'objet distant. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ <Aucun> ■ XONXOFF ■ RTS/CTS ■ Les deux

Propriété	Valeur renvoyée	Description
String = <CRemoteData>.ComStartedBy	Chaîne	Renvoie le contrôle de début des types de connexion COM. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI) ■ Réception 2 <RC> ■ Réponse du modem
String = <CRemoteData>.ComEndedBy	Chaîne	Renvoie le contrôle de fin des types de connexion COM Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI)
Long = <CRemoteData>.ComSpeed	Long	Renvoie le paramètre de vitesse de communication de l'objet distant Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Propriétés de périphérique NetBIOS

Le tableau suivant définit les propriétés et les paramètres d'un périphérique réseau NetBIOS.

Dans [Tableau 2-11](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-11 NetBIOS - Propriétés et paramètres

Propriété	Paramètre	Description
<CRemoteData>.LanaNumber(entier)	Entier	Définit le numéro d'adaptateur réseau (LANA) à utiliser pour les connexions NetBIOS.

Le tableau suivant définit les propriétés et les valeurs renvoyées par NetBIOS.

Dans [Tableau 2-12](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-12 Propriété NetBIOS et valeur renvoyée

Propriété	Valeur renvoyée	Description
Integer = <CRemoteData>.LanaNumber	Entier	Renvoie le paramètre en cours du numéro LANA pour les connexions NetBIOS.

Propriétés de périphérique RNIS via CAPI 2.0

Le tableau suivant définit les propriétés et les paramètres pour les connexions RNIS européennes.

Dans [Tableau 2-13](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-13 RNIS via CAPI 2.0 - Propriétés et paramètres

Propriété	Paramètre	Description
<CRemoteData>.CapiChannelBonding(Bool)	Bool	Active ou désactive la liaison de canaux pour les périphériques RNIS CAPI.
<CRemoteData>.CapiExtensions(chaine)	Chaîne	Définit toute extension CAPI supplémentaire nécessaire pour les communications.

Le tableau suivant définit les propriétés et les valeurs renvoyées pour les connexions RNIS européennes.

Dans [Tableau 2-14](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-14 RNIS via CAPI 2.0 - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
Bool= <CRemoteData>.CapiChannelBonding	Bool	Renvoie le paramètre de liaison de canaux RNIS CAPI en cours dans l'objet distant.
String = <CRemoteData>.CapiExtensions	Chaîne	Renvoie la liste actuelle des extensions CAPI à partir de l'objet distant.

objet CRemoteDataEx

L'objet CRemoteDataEx contient la même fonctionnalité que l'objet CRemoteData avec une fonctionnalité supplémentaire.

Le tableau suivant décrit les propriétés et les paramètres.

Dans [Tableau 2-15](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-15 CRemoteDataEx - Paramètres

Propriété	Paramètre	Description
<CRemoteData>.PrivateKey (chaîne)	Chaîne	Définit le nom du conteneur de clé privée à utiliser
<CRemoteData>.CertificateName(chaîne)	Chaîne	Définit le nom commun de la clé privée à utiliser.
<CRemoteData>.AutoLoginPassword(octet)	Octet	Définit le niveau de chiffrement. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ -1: Aucun ■ 0: pcAnywhere ■ 1: Symétrique ■ 2: Clé publique
<CRemoteData>.DenyLowerEncrypt(Bool)	Bool	Définit si l'ordinateur distant autorise une connexion à un ordinateur hôte utilisant un niveau de chiffrement inférieur.
<CRemoteData>.AutoDomain(chaîne)	Chaîne	Définit le nom de domaine pour les connexions automatiques. Cette option s'utilise avec les types d'authentification NT et Windows.

Le tableau suivant décrit les propriétés et les valeurs renvoyées pour l'objet CRemoteDataEx.

Dans [Tableau 2-16](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-16 CRemoteDataEx - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CRemoteData>.PrivateKey	Chaîne	Renvoie le nom du conteneur de clé privée actuellement actif
String = <CRemoteData>.CertificateName	Chaîne	Renvoie le nom commun du conteneur de clé privée actif.
Byte = <CRemoteData>.EncryptionLevel	Octet	Renvoie l'un des paramètres de chiffrement suivants : <ul style="list-style-type: none"> ■ -1: Aucun ■ 0: pcAnywhere ■ 1: Symétrique ■ 2: Clé publique
Bool = <CRemoteData>.DenyLowerEncrypt	Bool	Renvoie la valeur du paramètre de rejet de niveau de chiffrement inférieur.
String = <CRemoteData>.AutoDomain	Chaîne	Renvoie le paramètre de nom de domaine pour les connexions automatiques. Cette option s'utilise avec les types d'authentification NT et Windows.

Exemple de code Visual Basic pour la fonction distant

L'exemple de code Visual Basic suivant récupère un objet de données Distant et modifie ses propriétés.

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

    ''créer un objet CRemoteDataManager
    Set RemoteDataManager = CreateObject(WINAWSVR.REMOTEDATAMANAGER)
```

```
'Afficher et modifier le répertoire en cours
s = RemoteDataManager.CurrentDirectory()
MsgBox (s)
RemoteDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
s = RemoteDataManager.CurrentDirectory()
MsgBox (s)

''récupérer un objet de données distant
Set RemoteData = RemoteDataManager.RetrieveObjectEx("pod.CHF",
2, 0)

''afficher certaines propriétés
s = RemoteData.AreaCode()
MsgBox (s)
s = RemoteData.PhoneNumber()
MsgBox (s)

''définir certaines propriétés
RemoteData.AreaCode = "212"
RemoteData.PhoneNumber = "555-5555"

''écrire l'objet sur le disque
RemoteData.WriteObject (0)
End Sub
```

Utilisez les méthodes FindFirst et FindNext pour afficher le fichier distant dans un répertoire, comme suit :

```
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
```

Créez un objet distant. Définissez le type de connexion sur TCP/IP et le nom de l'ordinateur sur "Host1". Lancez l'objet comme suit :

```
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
```

Méthodes CHostDataManager

Les méthodes CHostDataManager fournissent les paramètres et les valeurs renvoyées pour accéder aux objets CHostData et les contrôler.

Méthode CHostDataManager	Paramètre ou valeur renvoyés	Description
CurrentDirectory()	Valeur renvoyée = chaîne	Nom complet de chemin d'accès au dossier de données pcAnywhere en cours dans lequel les objets hôte pcAnywhere sont enregistrés.

Méthode CHostDataManager	Paramètre ou valeur renvoyés	Description
FindFirst(schéma)	Paramètre = schéma en tant que chaîne	Schéma de nom de fichier pour filtrer les fichiers d'objet (un astérisque [*] trouve tous les fichiers dans le répertoire en cours).
FindFirst(nom)	Paramètre = Nom comme chaîne	Tampon de retour pour le chemin d'accès complet du fichier d'objet hôte correspondant au schéma spécifié.
FindFirst	Valeur renvoyée = booléen	VRAI si un fichier d'objet hôte correspondant au schéma spécifié est trouvé. Le chemin d'accès complet du fichier est stocké dans Nom.
FindNext(nom)	Paramètre = Nom comme chaîne	Tampon de retour pour le chemin d'accès complet du fichier d'objet distant correspondant au schéma spécifié dans l'appel initial de FindFirst().
FindNext	Valeur renvoyée = booléen	VRAI si un autre fichier d'objet hôte correspondant au schéma spécifié dans l'appel de FindFirst() est trouvé. Le chemin d'accès complet du fichier est stocké dans Nom.
RetrieveObject(nom)	Paramètre = Nom comme chaîne	Nom intégralement qualifié du fichier d'objet hôte à charger.
RetrieveObject(mode d'accès)	Paramètre = Mode d'accès comme valeur entière	Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes : <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
RetrieveObject(mot de passe)	Paramètre = Mot de passe comme chaîne	Mot de passe de l'objet. Peut être NULL.
RetrieveObject	Valeur renvoyée = Objet	Objet CHostData du fichier spécifié

Méthode CHostDataManager	Paramètre ou valeur renvoyés	Description
RetrieveObjectEx(nom)	Paramètre = Nom comme chaîne	Nom intégralement qualifié du fichier d'objet hôte à charger.
RetrieveObjectEx(mode d'accès)	Paramètre = Mode d'accès comme valeur entière	Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes : <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
RetrieveObjectEx(mot de passe)	Paramètre = Mot de passe comme chaîne	Mot de passe de l'objet. Peut être NULL.
RetrieveObjectEx	Valeur renvoyée = Objet	Objet CHostDataEx du fichier spécifié
CreateObject(nom)	Paramètre = Nom comme chaîne	Crée un objet CHostData et renvoie un pointeur LPDISPATCH qui le désigne. Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet.
CreateObject	Valeur renvoyée = Objet	CHostData
CreateObjectEx(nom)	Paramètre = Nom comme chaîne	Crée un objet CHostDataEx et renvoie un pointeur LPDISPATCH qui le désigne. Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet.
CreateObjectEx	Valeur renvoyée = Objet	CHostDataEx
DeleteObject(nom)	Paramètre = Nom comme chaîne	Nom intégralement qualifié du fichier d'objet hôte de l'objet à supprimer.
DeleteObject(mot de passe)	Paramètre = Mot de passe comme chaîne	Mot de passe de l'objet

Méthode CHostDataManager	Paramètre ou valeur renvoyés	Description
DeleteObject	Valeur renvoyée = booléen	VRAI si l'objet est supprimé.
Launch(nom)	Paramètre = Nom comme chaîne	Lance un fichier d'objet hôte, ce qui ouvre la fenêtre de terminal de l'hôte pcAnywhere. Le paramètre est le fichier objet hôte entièrement qualifié de l'objet à lancer.
Lancer	Valeur renvoyée = booléen	VRAI si l'objet est lancé avec succès.

Propriétés CHostData

Le tableau suivant décrit les propriétés et les paramètres disponibles pour l'objet CHostData.

Dans [Tableau 2-17](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-17 CHostData - Propriétés et paramètres

Propriété	Paramètre	Description
<CHostData>.PhoneNumber(chaine)	Chaîne	Définit le numéro de téléphone de l'ordinateur hôte.
<CHostData>.UseDialingProperties(Bool)	Bool	Définit les propriétés de numérotation du système.
<CHostData>.RedialCount(entier)	Entier	Définit le nombre de tentatives de rappel avant l'annulation de l'appel.
<CHostData>.RedialDelay(entier)	Entier	Définit le nombre de secondes d'attente entre les tentatives de rappel.
<CHostData>.Password(chaine)	Chaîne	Définit le mot de passe sur l'objet hôte, à utiliser avec les méthodes ExecuteProtection, ReadProtection et WriteProtection. Pour des raisons de sécurité, le serveur d'automatisation pcAnywhere ne permet pas de lire la valeur du mot de passe. Aucune valeur de mot de passe n'est renvoyée.

Propriété	Paramètre	Description
<CHostData>.ExecuteProtection (Bool)	Bool	Définit la nécessité d'un mot de passe pour exécuter l'objet. Définir par mot de passe.
<CHostData>.ReadProtection(Bool)	Bool	Définit l'obligation d'un mot de passe pour afficher les propriétés de l'objet hôte. Définir par mot de passe.
<CHostData>.WriteProtection(Bool)	Bool	Définit l'obligation d'un mot de passe pour enregistrer les modification de l'objet hôte. Définir par mot de passe.
<CHostData>.LogSession (Bool)	Bool	Active et désactive la consignation des sessions
<CHostData>.RecordFile(chaine)	Chaîne	Définit le nom et le chemin d'accès intégralement qualifié du fichier d'enregistrement de la session active.
<CHostData>.RecordSession(Bool)	Bool	Active et désactive la consignation automatique de session.
<CHostData>.ReadObject(chaine)	Chaîne	Définit le mot de passe de l'objet. Vous pouvez utiliser cette propriété pour actualiser la copie locale des données de l'objet hôte.
<CHostData>.WriteObject(chaine)	Chaîne	Définit le mot de passe de l'objet. Utilisez cette propriété pour créer l'objet hôte ou pour écrire les changements apportés à celui-ci.

Le tableau suivant décrit les propriétés et les valeurs renvoyées par CHostData.

Dans [Tableau 2-18](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-18 CHostData - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CHostData>.PhoneNumber	Chaîne	Renvoie le numéro de téléphone de l'ordinateur hôte
Bool = <CHostData>.UseDialingProperties	Bool	Renvoie l'utilisation des propriétés de numérotation du système définies dans l'objet hôte.
Integer = <CHostData>.RedialCount	Entier	Renvoie le nombre de tentatives d'appel défini dans l'objet hôte.

Propriété	Valeur renvoyée	Description
Integer = <CHostData>.RedialDelay	Entier	Renvoie le nombre de secondes d'attente entre les tentatives de rappel.
Bool = <CHostData>.ExecuteProtection	Bool	Renvoie la valeur du paramètre ExecuteProtection.
Bool = <CHostData>.ReadProtection	Bool	Renvoie la valeur du paramètre ReadProtection.
Bool = <CHostData>.WriteProtection	Bool	Renvoie la valeur du paramètre WriteProtection.
Bool = <CHostData>.LogSession	Bool	VRAI si la consignation de session est activée.
String = <CHostData>.RecordFile	Chaîne	Renvoie le nom et le chemin d'accès intégralement qualifié du fichier d'enregistrement de la session.
Bool = <CHostData>.RecordSession	Bool	Renvoie la valeur du paramètre d'enregistrement de session.

Propriétés du type de connexion

Le tableau suivant décrit les propriétés et les paramètres du type de connexion. Dans [Tableau 2-19](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-19 Type de connexion - Propriétés et paramètres

Propriété	Paramètre	Description
<CHostData>.ConnectionType(chaine)	Chaîne	Définit le type de connexion de l'hôte. La valeur transmise doit être un type de connexion valide, défini par les fonctions FirstConnectionType() et NextConnectionType(). Les types suivants sont des exemples de connexions valides : <ul style="list-style-type: none"> ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ RNIS via CAPI 2.0 ■ Nom du modem (tel qu'il apparaît sur l'ordinateur)
<CHostData>.AssignConnection(chaine)	Chaîne	Définit le type de connexion présent dans la chaîne sur l'état actif.
<CHostData>.UnassignConnection(chaine)	Chaîne	Définit le type de connexion présent dans la chaîne sur l'état inactif.

Le tableau suivant décrit les propriétés et les valeurs renvoyées pour le type de connexion.

Dans [Tableau 2-20](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-20 Type de connexion - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CHostData>.ConnectionTypes	Chaîne	Renvoie le type de connexion de l'objet hôte.

Propriété	Valeur renvoyée	Description
Integer = <CHostData>.ConnectionType	Entier	Renvoie le nombre de types de connexion disponibles. Les types suivants sont des exemples de connexions valides : <ul style="list-style-type: none"> ■ COM1 ■ COM2 ■ COM3 ■ COM4 ■ TCP/IP ■ SPX ■ NetBIOS ■ RNIS via CAPI 2.0 ■ Nom du modem (tel qu'il apparaît sur l'ordinateur)
String = <CHostData>.FirstConnectionType	Chaîne	Renvoie le premier type de connexion disponible.
String = <CHostData>.NextConnectionType	Chaîne	Renvoie le type de connexion disponible suivant. L'appel s'effectue de façon séquentielle pour le nombre de types de connexion défini dans <CHostData>.ConnectionType, afin d'énumérer tous les types de connexion.
Bool = <CHostData>.FindConnectionType(type de connexion)	Bool	VRAI si le type de connexion indiqué a été trouvé dans la liste des types de connexion disponibles.
Integer = <CHostData>.MaxAssignedConnections	Entier	Renvoie le nombre maximum de types de connexion qui peuvent être actifs sur cet hôte.
String = <CHostData>.FirstAssignedConnection	Chaîne	Renvoie le premier type de connexion actif affecté.
String = <CHostData>.NextAssignedConnection	Chaîne	Renvoie le type de connexion actif affecté suivant.
Bool = <CHostData>.FindAssignedConnection(type de connexion)	Bool	VRAI si le type de connexion transmis correspond à l'un des types de connexion actifs.

Méthode AssignConnection(type de connexion)

Cette méthode place le type de connexion requis dans la liste des types de connexion affectés à l'objet Hôte et en fait le type de connexion en cours lors du traitement ultérieur des appels de méthodes spécifiques au périphérique.

Si le type de connexion requis se trouve déjà dans la liste des connexions affectées, celle-ci ne change pas. Seul le type de connexion actuel est remplacé par le type demandé. Il est normal d'appeler la méthode AssignConnection sur le même objet plusieurs fois lors de l'obtention et de la définition de valeurs propres à la connexion.

AssignConnection renvoie VRAI si le type de connexion transmis existe sur l'ordinateur et est affecté avec succès ou déjà affecté. Il renvoie FAUX si le type de connexion requis n'existe pas sur l'ordinateur ou si le nombre de connexions affectées a déjà atteint le maximum.

Un objet hôte pcAnywhere peut prendre en charge jusqu'à deux types de connexion affectés. La méthode AssignConnection renvoie FAUX si elle détecte une tentative de dépassement de cette limite.

Le tableau suivant définit le paramètre AssignConnection :

Paramètre	Description
Type de connexion comme chaîne	Nom d'un type de périphérique de connexion à affecter.

Le tableau suivant définit la valeur renvoyée pour AssignConnection :

Valeur renvoyée	Description
Booléenne	VRAI si ce type de périphérique est disponible et si le nombre maximum de connexions affectées n'a pas encore été atteint.

Méthode UnassignConnection (type de connexion)

Cette méthode renvoie VRAI si le type de connexion transmis a pu être retiré de la liste des types de connexion affectés.

Le tableau suivant définit le paramètre UnassignConnection :

Paramètre	Description
Type de connexion comme chaîne	Nom d'un type de périphérique de connexion à désaffecter.

Le tableau suivant définit la valeur renvoyée pour UnassignConnection :

Valeur renvoyée Description

Booléenne VRAI si ce type de périphérique est désaffecté avec succès.

Propriétés de numérotation

Le tableau suivant décrit les propriétés et les valeurs renvoyées pour définir les propriétés de numérotation pour les connexions par modem.

Dans [Tableau 2-21](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-21 Propriétés de numérotation - Propriétés et paramètres

Propriété	Paramètre	Description
<CHostData>.AreaCode(chaine)	Chaîne	Définit le code de zone pour les connexions par modem
<CHostData>.CountryCode(chaine)	Chaîne	Définit l'indicatif international pour les connexions par modem

Le tableau suivant définit les propriétés et les valeurs renvoyées pour les propriétés de numérotation par modem.

Dans [Tableau 2-22](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-22 Propriétés de numérotation par modem - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CHostData>.AreaCode	Chaîne	Renvoie le code de zone
String = <CHostData>.CountryCode	Chaîne	Renvoie l'indicatif international.
Integer = <CHostData>.CountryCodes	Entier	Renvoie le nombre d'indicatifs internationaux disponibles.
String = <CHostData>.FirstCountryCode	Chaîne	Renvoie le premier indicatif de pays apparaissant dans la liste du système d'exploitation.

Propriété	Valeur renvoyée	Description
String = <CHostData>.NextCountryCode	Chaîne	Renvoie l'indicatif international disponible suivant. L'appel s'effectue de façon séquentielle pour le nombre d'indicatifs internationaux défini dans <CHostData>.CountryCodes, afin d'énumérer tous les indicatifs internationaux.

Propriétés de périphérique COM

Le tableau suivant définit les propriétés et les paramètres qui vous permettent de personnaliser les paramètres de port du modem et des autres connexions COM.

Dans [Tableau 2-23](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-23 Périphérique COM - Propriétés et paramètre

Propriété	Paramètre	Description
<CHostData>.ComParity (chaîne)	Chaîne	Définit la parité des communications. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ <Aucune> ■ Impaire ■ Paire ■ Marque ■ Espace
<CHostData>.ComFlowControl (chaîne)	Chaîne	Définit le contrôle de flux des types de connexion COM. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ <Aucune> ■ XONXOFF ■ RTS/CTS ■ Les deux

Propriété	Paramètre	Description
<CHostData>.ComStartedBy(chaine)	Chaîne	Définit le paramètre de début des types de connexion COM. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI) ■ Réception 2 <RC> ■ Réponse du modem
<CHostData>.ComEndedBy(chaine)	Chaîne	Définit le paramètre de fin des types de connexion COM. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI)
<CHostData>.ComSpeed(Long)	Long	Définit la vitesse COM maximale Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Le tableau suivant décrit les propriétés et les valeurs renvoyées par le périphérique COM.

Dans [Tableau 2-24](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-24 Périphériques COM - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CHostData>.ComParity	Chaîne	<p>Renvoie l'une des valeurs suivantes comme parité de communications :</p> <ul style="list-style-type: none"> ■ <Aucune> ■ Impaire ■ Paire ■ Marque ■ Espace
String = <CHostData>.ComFlowControl	Chaîne	<p>Renvoie le paramètre de contrôle de flux de l'objet hôte.</p> <p>Les valeurs suivantes sont valides :</p> <ul style="list-style-type: none"> ■ <Aucun> ■ XONXOFF ■ RTS/CTS ■ Les deux
String = <CHostData>.ComStartedBy	Chaîne	<p>Renvoie le contrôle de début des types de connexion COM.</p> <p>Les valeurs suivantes sont valides :</p> <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI) ■ Réception 2 <RC> ■ Réponse du modem
String = <CHostData>.ComEndedBy	Chaîne	<p>Renvoie le contrôle de fin des types de connexion COM</p> <p>Les valeurs suivantes sont valides :</p> <ul style="list-style-type: none"> ■ Toujours connecté ■ Détection de porteuse (DCD) ■ Prêt à émettre (CTS) ■ Ensemble de données prêt (DSR) ■ Signal de sonnerie (RI)

Propriété	Valeur renvoyée	Description
Long = <CHostData>.ComSpeed	Long	Renvoie le paramètre de vitesse de communication de l'objet hôte. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 110 ■ 300 ■ 600 ■ 1200 ■ 2400 ■ 4800 ■ 9600 ■ 38400 ■ 57600 ■ 115200

Propriétés de périphérique NetBIOS

Le tableau suivant définit les propriétés et les paramètres d'un périphérique réseau NetBIOS.

Dans [Tableau 2-25](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-25 NetBIOS - Propriétés et paramètres

Propriété	Paramètre	Description
<CHostData>.LanaNumber(entier)	Entier	Définit le numéro d'adaptateur réseau (LANA) à utiliser pour les connexions NetBIOS.

Le tableau suivant définit les propriétés et les valeurs renvoyées pour un périphérique réseau NetBIOS.

Dans [Tableau 2-26](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-26 Valeurs NetBIOS renvoyées

Propriété	Valeur renvoyée	Description
Integer = <CHostData>.LanaNumber	Entier	Renvoie le paramètre en cours du numéro LANA pour les connexions NetBIOS.

Propriétés de périphérique RNIS via CAPI 2.0

Le tableau suivant définit les propriétés et les paramètres pour les connexions RNIS européennes.

Dans [Tableau 2-27](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-27 RNIS via CAPI 2.0 - Propriétés et paramètres

Propriété	Paramètre	Description
<CHostData>.CapiChannelBonding(Bool)	Bool	Active ou désactive la liaison de canaux pour les périphériques RNIS CAPI.
<CHostData>.CapiExtensions(chaine)	Chaîne	Définit toute extension CAPI supplémentaire nécessaire pour les communications.

Le [Tableau 2-28](#) définit les propriétés et les valeurs renvoyées pour les connexions RNIS européennes.

Tableau 2-28 RNIS via CAPI 2.0 - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
Bool= <CHostData>.CapiChannelBonding	Bool	Renvoie le paramètre de liaison de canaux RNIS CAPI en cours dans l'objet hôte.
String = <CHostData>.CapiExtensions	Chaîne	Renvoie la liste actuelle des extensions CAPI à partir de l'objet hôte.

objet CHostDataEx

L'objet CHostDataEx contient la même fonctionnalité que l'objet CHostData avec une fonctionnalité supplémentaire.

Le tableau suivant décrit les propriétés et les paramètres.

Dans [Tableau 2-29](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-29 CHostDataEx - Propriétés et paramètres

Propriété	Paramètre	Description
<CHostData>.CryptPrivateKey (chaîne)	Chaîne	Définit le nom du conteneur de clé privée à utiliser

Propriété	Paramètre	Description
<CHostData>.CryptCommonName (chaîne)	Chaîne	Définit le nom commun de la clé privée à utiliser.
<CHostData>.CryptReqLevel(Octet)	Octet	Définit le niveau de chiffrement. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ -1: Aucun ■ 0: pcAnywhere ■ 1: Symétrique ■ 2: Clé publique
<CHostData>.CryptRefuseLower(Bool)	Bool	Définit si l'ordinateur hôte accepte une connexion d'un ordinateur distant utilisant un niveau de chiffrement inférieur.
<CHostData>.CallersPath(chaîne)	Chaîne	Définit le chemin d'accès intégralement qualifié des fichiers d'appelant.
<CHostData>.ConfirmConnect(Bool)	Bool	Définit si un message demandera à l'utilisateur hôte de confirmer les connexions.
<CHostData>.ConfirmTimeout(Octet)	Octet	Définit le nombre de secondes de délai pour confirmer la connexion.
<CHostData>.ConfirmDeny(Bool)	Bool	Définit si la connexion doit se terminer lorsque le délai de confirmation de la connexion expire.
<CHostData>.PwCaseSensitive(Bool)	Bool	Impose l'utilisation de mots de passe sensibles à la casse.
<CHostData>.PwAttempts(Octet)	Octet	Définit le nombre d'échecs d'ouverture de session consécutifs autorisés avant l'abandon de la connexion.
<CHostData>.PwTimeout(Octet)	Octet	Définit le nombre de minutes dont dispose l'utilisateur pour ouvrir une session avant l'abandon de la connexion.
<CHostData>.ActiveKbds(Octet)	Octet	Définit quelle souris et quel clavier seront actifs pendant la connexion. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 0: Hôte et distant ■ 1: Hôte ■ 2: Distant

Propriété	Paramètre	Description
<CHostData>.InactiveTimeout(Octet)	Octet	Définit le nombre de minutes pendant lequel le clavier et la souris peuvent rester inactifs avant l'abandon de la connexion.
<CHostData>.LockSystemWhileWait(Bool)	Bool	Définit le verrouillage de l'ordinateur hôte lors du démarrage.
<CHostData>.MinimizeOnLaunch(Bool)	Bool	Définit l'option de démarrage de l'hôte en réduction.
<CHostData>.RunAsService(Bool)	Bool	Active le fonctionnement de l'hôte comme service.
<CHostData>.ConnLostWait(Octet)	Octet	Définit le nombre de minutes d'attente avant d'autoriser un autre appelant à se connecter.
<CHostData>.ConnLostHostOpts(Bool)	Bool	Définit s'il faut attendre une autre connexion ou annuler l'hôte si la session se termine de façon anormale. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ FAUX : Attente de la connexion suivante ■ VRAI : Annulation de l'hôte
<CHostData>.EnableConnLostSecurity(Bool)	Bool	Active ou désactive les options de sécurité de fin de session pour les sessions qui se terminent de façon anormale.

Propriété	Paramètre	Description
<CHostData>.AuthenticationType(octet)	Octet	<p>Définit le type d'authentification.</p> <p>Les valeurs suivantes sont valides :</p> <ul style="list-style-type: none"> ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS Active Directory Services ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID <p>L'authentification pcAnywhere est utilisée par défaut si aucune valeur d'authentification n'est définie ou si la valeur définie n'est pas valide (par exemple, si le type d'authentification n'est pas disponible).</p>
<CHostData>.ConnLostSecurity(octet)	Octet	<p>Définit les options de sécurité pour traiter une fin de session anormale.</p> <p>Les valeurs suivantes sont valides :</p> <ul style="list-style-type: none"> ■ 1: Déconnecter l'utilisateur ■ 2: Redémarrer l'ordinateur hôte ■ 3: Verrouiller l'ordinateur
<CHostData>.CallbkDelay(octet)	Octet	<p>Définit le nombre de secondes d'attente avant que le modem hôte ne rappelle l'ordinateur distant.</p>
<CHostData>.EndSessHostOpts(Bool)	Bool	<p>Définit si l'hôte attend une autre connexion ou est annulé après la fin normale de la session.</p> <p>Les valeurs suivantes sont valides :</p> <ul style="list-style-type: none"> ■ FAUX : Attente de la connexion suivante ■ VRAI : Annulation de l'hôte
<CHostData>.EnableEndSessSecurity(Bool)	Bool	<p>Active ou désactive les options de sécurité pour une fin de session normale.</p>

Propriété	Paramètre	Description
<CHostData>.EndSessSecurity(octet)	Octet	Définit les options de sécurité pour une fin de session normale. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 1: Déconnecter l'utilisateur ■ 2: Redémarrer l'ordinateur hôte ■ 3: Verrouiller l'ordinateur
<CHostData>.BlankHost(Bool)	Bool	Définit l'effacement de l'écran de l'ordinateur hôte. Certaines cartes d'affichage ne prennent pas en charge cette fonction.
<CHostData>.AllowRemoteMouse(Bool)	Bool	Désactive l'utilisation de la souris sur l'ordinateur distant pendant une session.
<CHostData>.RebootOnDisconnect(Bool)	Bool	Si VRAI, force l'ordinateur à redémarrer après toute fin de session.
<CHostData>.PasswordAfterDisc(Bool)	Bool	Déconnecte l'utilisateur lorsque la session se termine.
<CHostData>.LogFailures(Bool)	Bool	Consigne les tentatives de mot de passe qui ont échoué.
<CHostData>.AllowDriveSecurity(Bool)	Bool	Active les options de sécurité de lecteur. Ce paramètre n'est valide que sur les systèmes de fichiers NTFS.
<CHostData>.UseDirectoryServices(Bool)	Bool	Active l'utilisation des services d'annuaire pour l'authentification.
<CHostData>.DirectoryServiceEntry(chaine)	Chaîne	Définit les paramètres des services d'annuaire.

Le tableau suivant décrit les propriétés et les valeurs renvoyées pour l'objet CHostDataEx.

Dans [Tableau 2-30](#), remplacez les informations entre les chevrons par les valeurs réelles.

Tableau 2-30 CHostDataEx - Propriétés et valeurs renvoyées

Propriété	Valeur renvoyée	Description
String = <CHostDataEx>.CryptPrivateKey	Chaîne	Renvoie le nom du conteneur de clé privée actuellement actif.

Propriété	Valeur renvoyée	Description
String = <CHostDataEx>.CryptCommonName	Chaîne	Renvoie le nom commun du conteneur de clé privée actif.
Byte = <CHostDataEx>.CryptReqLevel	Octet	Renvoie l'un des paramètres de chiffrement suivants : <ul style="list-style-type: none"> ■ -1: Aucun ■ 0: pcAnywhere ■ 1: Symétrique ■ 2: Clé publique
Bool = <CHostDataEx>.CryptRefuseLower	Bool	Renvoie la valeur du paramètre de rejet de niveau de chiffrement inférieur.
String = <CHostDataEx>.CallersPath	Chaîne	Renvoie le chemin d'accès intégralement qualifié des fichiers d'appelant.
Bool = <CHostDataEx>.ConfirmConnect	Bool	Renvoie la valeur du paramètre de confirmation de connexion.
Byte = <CHostDataEx>.ConfirmTimeout	Octet	Renvoie la valeur du paramètre de rejet de niveau de chiffrement inférieur.
Bool = <CHostDataEx>.ConfirmDeny	Bool	Renvoie la valeur du paramètre de déconnexion sur délai dépassé.
Bool = <CHostDataEx>.PwCaseSensitive	Bool	Renvoie la valeur du paramètre de mot de passe sensible à la casse.
Byte = <CHostDataEx>.PwAttempts	Octet	Renvoie la valeur du paramètre de nombre limite de tentatives d'ouverture de session par appel.
Byte = <CHostDataEx>.PwTimeout	Octet	Renvoie la valeur du paramètre de délai limite d'ouverture de session (en minutes).
Byte = <CHostDataEx>.ActiveKbds	Octet	Renvoie les paramètres de clavier et de souris actifs. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 0: Hôte et distant ■ 1: Hôte ■ 2: Distant

Propriété	Valeur renvoyée	Description
Byte = <CHostDataEx>.InactiveTimeout	Octet	Renvoie le nombre de minutes d'attente avant déconnexion si la limite d'inactivité est atteinte.
Bool = <CHostDataEx>.LockSystemWhileWait	Bool	Renvoie le paramètre de verrouillage de l'ordinateur hôte au démarrage.
Bool = <CHostDataEx>.MinimizeOnLaunch	Bool	Renvoie le paramètre de démarrage de l'hôte en réduction.
Bool = <CHostDataEx>.RunAsService	Bool	Renvoie le paramètre de fonctionnement de l'hôte comme service.
Byte = <CHostDataEx>.ConnLostWait	Octet	Renvoie le nombre de minutes d'attente avant d'autoriser un autre appelant à se connecter.
Bool = <CHostDataEx>.ConnLostHostOpts	Bool	Renvoie le paramètre déterminant si l'hôte attend une autre connexion ou s'arrête si la session se termine de façon anormale. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ FAUX : Attente de la connexion suivante ■ VRAI : Annulation de l'hôte
Bool = <CHostDataEx>.ConnLostWait	Bool	Renvoie la valeur de l'option de sécurité définie pour traiter une fin de session anormale.

Propriété	Valeur renvoyée	Description
Byte = <CHostDataEx>.AuthenticationType	Octet	Renvoie la référence de nombre du type d'authentification. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 0: pcAnywhere ■ 1: pcAnywhere ■ 2: Windows ■ 3: NT ■ 4: pcAnywhere ■ 5: pcAnywhere ■ 6: ADS Active Directory Services ■ 7: Microsoft LDAP ■ 8: FTP ■ 9: HTTP ■ 10: HTTPS ■ 11: RSA SecurID L'authentification pcAnywhere est utilisée par défaut si aucune valeur d'authentification n'est définie ou si la valeur définie n'est pas valide (par exemple, si le type d'authentification n'est pas disponible).
Byte = <CHostDataEx>.ConnLostSecurity	Octet	Renvoie la représentation numérique du niveau de sécurité défini pour traiter une fin de session anormale. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 1: Déconnecter l'utilisateur ■ 2: Redémarrer l'ordinateur hôte ■ 3: Verrouiller l'ordinateur
Byte = <CHostDataEx>.CallbkDelay	Octet	Renvoie le nombre de secondes d'attente avant que le modem hôte ne rappelle l'ordinateur distant.
Bool = <CHostDataEx>.EndSessHostOpts	Bool	Renvoie le paramètre déterminant si l'hôte attend une autre connexion ou est annulé après une fin de session normale. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ FAUX : Attente de la connexion suivante ■ VRAI : Annulation de l'hôte

Propriété	Valeur renvoyée	Description
Bool = <CHostDataEx>.EnableEndSessSecurity	Bool	Renvoie le paramètre déterminant si les options de sécurité de fin de session sont activées pour une fin de session normale.
Byte = <CHostDataEx>.EndSessSecurity	Octet	Renvoie l'option de sécurité définie pour une fin de session normale. Les valeurs suivantes sont valides : <ul style="list-style-type: none"> ■ 1: Déconnecter l'utilisateur ■ 2: Redémarrer l'ordinateur hôte ■ 3: Verrouiller l'ordinateur
Bool = <CHostDataEx>.BlankHost	Bool	Renvoie l'option définie pour l'effacement de l'écran.
Bool = <CHostDataEx>.AllowRemoteMouse	Bool	Renvoie l'option déterminant si l'utilisateur distant dispose du contrôle du clavier et de la souris au cours d'une session.
Bool = <CHostDataEx>.RebootOnDisconnect	Bool	Renvoie l'option déterminant si l'ordinateur hôte redémarre après la fin de la session.
Bool = <CHostDataEx>.PasswordAfterDisc	Bool	Renvoie l'option déterminant si l'hôte est déconnecté après la fin de la session.
Bool = <CHostDataEx>.LogFailures	Bool	Renvoie l'option déterminant si les échecs de mots de passe sont consignés.
Bool = <CHostDataEx>.AllowDriveSecurity	Bool	Renvoie l'option déterminant si la sécurité du lecteur est activée.
Bool = <CHostDataEx>.UseDirectoryServices	Bool	Renvoie l'option déterminant si les services d'annuaire sont activés.
String = <CHostDataEx>.DirectoryServiceEntry	Chaîne	Renvoie les paramètres de services d'annuaire.

Exemple de code Visual Basic pour la fonctionnalité hôte.

L'exemple de code Visual Basic suivant récupère un objet de données hôte et modifie ses propriétés.

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

```
'créer un objet CHostDataManager
Set HostDataManager = CreateObject(WINAWSVR.BEHOSTDATAMANAGER)

'Afficher et modifier le répertoire en cours
s = HostDataManager.CurrentDirectory()
MsgBox (s)
HostDataManager.ChangeDirectory ("C:\dev\bin.w32\data")
s = HostDataManager.CurrentDirectory()
MsgBox (s)

'Récupérer un objet de données distant
Set HostData = HostDataManager.RetrieveObject("pod.BHF", 2, 0)

'Afficher certaines propriétés
s = HostData.AreaCode()
MsgBox (s)
s = HostData.PhoneNumber()
MsgBox (s)
'Définir certaines propriétés
RemoteData.HostData = "212"
RemoteData.HostData = "555-5555"

'Écrire l'objet sur le disque
HostData.WriteObject (0)
End Sub
```

Utilisez les méthodes FindFirst et FindNext pour afficher le fichier d'hôte dans un répertoire, comme suit :

```
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

```

fonctions awrem32

Les fonctions Awrem32 fournissent des paramètres et renvoient des valeurs pour la gestion des connexions entre l'ordinateur hôte et l'ordinateur distant.

fonction awrem32	Paramètre ou valeur renvoyés	Description
awConnect(nom de fichier)	Paramètre = Nom comme chaîne	Nom intégralement qualifié du fichier .chf contenant les informations sur l'ordinateur hôte.
awConnect	Valeur renvoyée = booléen	VRAI si la commande est exécutée.
awDisconnect()	Valeur renvoyée = booléen	Après avoir appelé cette fonction, le programme appelant doit supprimer l'objet (C++ - delete IAwrem32X*, VB - set ObjectName = Nothing;)
FileXferFromHost(HostFile)	Paramètre = HostFile comme chaîne	Contient le chemin d'accès intégralement qualifié du fichier à copier à partir de l'ordinateur hôte.
FileXferFromHost(RemoteFile)	Paramètre = RemoteFile comme chaîne	Contient le chemin d'accès de destination, intégralement qualifié. Les chaînes de fichier hôte et de fichier distant ne sont pas obligatoirement identiques.
FileXferFromHost	Valeur renvoyée = booléen	VRAI si la commande est exécutée.

fonction awrem32	Paramètre ou valeur renvoyés	Description
FileXferToHost(HostFile)	Paramètre = HostFile comme chaîne	<p>Copie un fichier de l'ordinateur distant vers l'ordinateur hôte. Les paramètres peuvent contenir des caractères génériques.</p> <p>Le paramètre contient le chemin d'accès de destination et le nom de fichier, entièrement qualifiés.</p>
FileXferToHost(RemoteFile)	Paramètre = RemoteFile comme chaîne	Contient le chemin d'accès intégralement qualifié du fichier à copier à partir de l'ordinateur distant. Les chaînes de fichier hôte et de fichier distant ne sont pas obligatoirement identiques.
CreateFolderOnHost(nom de répertoire)	Paramètre = FolderName comme chaîne	Contient le lecteur et le chemin d'accès du dossier sur l'ordinateur hôte.
CreateFolderOnHost	Valeur renvoyée = booléen	VRAI si la commande est exécutée.
ExecuteHostFile(nom de fichier)	Paramètre = Nom de fichier comme chaîne	<p>Exécute un fichier existant sur l'ordinateur hôte. Cette fonction ne peut exécuter que les fichiers batch, de commandes et exécutables. Elle n'exécute pas les fichiers associés aux exécutables. Par exemple, cette fonction n'ouvre pas Microsoft Word si vous exécutez un fichier .doc.</p> <p>Le paramètre contient le chemin d'accès intégralement qualifié du fichier sur l'ordinateur hôte.</p>
ExecuteHostFile	Valeur renvoyée = booléen	VRAI si la commande est exécutée.
GetError()	Valeur renvoyée = Chaîne	Renvoie la dernière erreur générée dans Awrem32.

fonction awrem32	Paramètre ou valeur renvoyés	Description
ConnectionStatus()	Valeur retournée = Court	Renvoie l'état courant de la connexion à l'ordinateur hôte. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none">■ -1 = Connexion perdue■ 0 = Aucune connexion■ 1 = Session connectée

Définitions d'objets Visual C++

Ce chapitre traite des sujets suivants :

- [A propos des objets Visual C++](#)
- [Méthodes CRemoteDataManager](#)
- [objet CRemoteData](#)
- [objet CRemoteDataEx](#)
- [Méthodes CHostDataManager](#)
- [objet CHostData](#)
- [objet CHostDataEx](#)
- [fonctions awrem32](#)

A propos des objets Visual C++

Le serveur d'automatisation pcAnywhere offre les composants suivants pour la prise en charge de l'automatisation OLE :

Winawsvr	Contient les informations nécessaires pour se connecter au serveur d'automatisation pcAnywhere et accéder à ses interfaces.
Awrem32	Contient les informations nécessaires au contrôle des connexions pcAnywhere.

Les objets décrits dans le [Tableau 3-1](#) composent Winawsvr.

Tableau 3-1 Objets Winawsvr

Objet	Description	Référence
CRemoteDataManager	Fournit les méthodes pour créer, ouvrir, modifier, enregistrer et supprimer les objets CRemoteData.	Se reporter à " Méthodes CRemoteDataManager " à la page 62.
CRemoteData	Définit les paramètres permettant d'accéder à la fonction distant de pcAnywhere et de la contrôler.	Se reporter à " objet CRemoteData " à la page 66. Se reporter à " objet CRemoteDataEx " à la page 72.
CHostDataManager	Fournit les méthodes pour créer, ouvrir, modifier, enregistrer et supprimer les objets ChostData.	Se reporter à " Méthodes CHostDataManager " à la page 73.
CHostData	Définit les paramètres permettant d'accéder à la fonction hôte de pcAnywhere et de la contrôler.	Se reporter à " objet CRemoteData " à la page 66. Se reporter à " objet CRemoteDataEx " à la page 72.

Awrem32 possède un objet, constitué de huit interfaces, pour prendre en charge les sessions de téléintervention et de transfert de fichiers.

Se reporter à "[fonctions awrem32](#)" à la page 88.

Quelques fonctions ne sont plus prises en charge. Cependant, les définitions d'objets sont utilisables avec les versions précédentes.

Pour les fonctions nécessitant des mots de passe, les valeurs des mots de passe peuvent être définies mais non récupérées, pour des raisons de sécurité.

Méthodes CRemoteDataManager

Les méthodes CRemoteDataManager fournissent les paramètres et renvoient les valeurs permettant d'accéder aux objets CRemoteData et de les contrôler.

Méthode	Paramètre ou valeur renvoyés	Description
BSTR CurrentDirectory();	Paramètre = BSTR	Nom complet de chemin d'accès au dossier de données pcAnywhere en cours dans lequel les objets distants sont enregistrés.

Méthode	Paramètre ou valeur renvoyés	Description
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Paramètre = LPCTSTR lpszNewDirectory Valeur renvoyée = booléen	Change le dossier actuel dans lequel les objets distants pcAnywhere sont stockés. Le paramètre est le nom d'un dossier existant. La valeur retournée est VRAIE en cas de succès.
BOOL FindFirst(LPCTSTR lpszPattern, BSTR FAR*);	Paramètre = LPCTSTR lpszPattern	Schéma de nom de fichier pour filtrer les fichiers d'objet (un astérisque [*] trouve tous les fichiers dans le répertoire en cours).
BOOL FindFirst(pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName	Tampon de retour pour le chemin d'accès complet du fichier d'objet distant correspondant au schéma spécifié.
BOOL FindFirst	Valeur renvoyée = booléen	VRAI si un fichier d'objet distant correspondant au schéma spécifié est trouvé. Le chemin d'accès complet du fichier est stocké dans pbstrFullQualName.
BOOL FindNext(BSTR FAR* pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName Valeur renvoyée = booléen	Après l'appel réussi de FindFirst() pour obtenir le nom d'un fichier d'objet distant dans le dossier en cours, FindNext() peut être appelé pour trouver le fichier suivant correspondant au schéma. Renvoie la valeur VRAI si un autre fichier d'objet distant correspondant au schéma spécifié dans l'appel de FindFirst() est trouvé. Le chemin d'accès complet du fichier est stocké dans pbstrFullQualName.
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Récupère un objet CRemoteData par nom de fichier. Le paramètre est le nom intégralement qualifié du fichier d'objet distant à charger.

Méthode	Paramètre ou valeur renvoyés	Description
LPDISPATCH RetrieveObject(short wAccessMode);	Parameter = short wAccessMode	Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes : <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Mot de passe de l'objet. Peut être NULL.
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Récupère un objet CRemoteDataEx par nom de fichier. Le paramètre est le nom intégralement qualifié du fichier d'objet distant à charger.
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parameter = short wAccessMode	Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes : <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Mot de passe de l'objet. Peut être NULL.
LPDISPATCH RetrieveObjectEx	Valeur renvoyée = LPDISPATCH	Pointeur sur un objet envoi OLE. L'objet est un objet CRemoteDataEx. Voir Exemple de code Visual C++ pour la fonctionnalité du distant.

Méthode	Paramètre ou valeur renvoyés	Description
LPDISPATCH CreateObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Valeur renvoyée = LPDISPATCH	<p>Crée un objet CRemoteData et renvoie un pointeur LPDISPATCH qui le désigne.</p> <p>Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet.</p> <p>La valeur renvoyée est un pointeur vers un objet envoi OLE. L'objet est un objet CRemoteData.</p> <p>Voir Exemple de code Visual C++ pour la fonctionnalité du distant.</p>
LPDISPATCH CreateObjectEx(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Valeur renvoyée = LPDISPATCH	<p>Crée un objet CRemoteDataEx et renvoie un pointeur LPDISPATCH qui le désigne.</p> <p>Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet.</p> <p>La valeur renvoyée est un pointeur vers un objet envoi OLE. L'objet est un objet CRemoteDataEx.</p> <p>Voir Exemple de code Visual C++ pour la fonctionnalité du distant.</p>
BOOL DeleteObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	<p>Supprime un fichier d'objet distant.</p> <p>Le paramètre est le nom intégralement qualifié du fichier d'objet distant de l'objet à supprimer.</p>
BOOL DeleteObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	<p>Mot de passe de l'objet. Peut être NULL.</p>
BOOL DeleteObject	BOOL	<p>VRAI si l'objet est supprimé.</p>
BOOL Launch(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Valeur renvoyée = booléen	<p>Lance un fichier d'objet distant, ce qui ouvre la fenêtre de terminal distant pcAnywhere.</p> <p>Le paramètre est le fichier intégralement qualifié d'objet distant de l'objet à lancer.</p> <p>La valeur renvoyée est VRAI si l'objet est lancé avec succès.</p>

objet CRemoteData

Utilisez cet objet pour modifier les données d'objet distant.

Méthodes Get et Set (obtenir et définir)

Les méthodes suivantes servent à obtenir (Get) et définir (Set) les propriétés de l'objet CRemoteData.

Le nom de l'ordinateur est le nom de système de l'hôte pcAnywhere à appeler lorsque l'objet distant est lancé.

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

Le numéro de téléphone est le numéro à composer pour établir une connexion par modem avec un ordinateur hôte pcAnywhere.

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

Indique si des propriétés de numérotation TAPI doivent être utilisées (informations d'emplacement) (VRAI) ou si la chaîne du numéro de téléphone doit être utilisée exactement telle qu'elle apparaît (FAUX).

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

Si les propriétés de numérotation sont utilisées, il s'agit du code de zone du numéro à appeler.

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

Si les propriétés de numérotation sont utilisées, il s'agit de l'indicatif international du numéro à appeler.

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

Nombre de tentatives d'appel de ce numéro si l'appel échoue.

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

Délai en secondes entre les tentatives d'appel.

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

Nom d'utilisateur à envoyer à l'hôte lorsqu'une connexion est établie. Si ce champ reste vide, l'utilisateur doit fournir un nom à la connexion.

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

Mot de passe de session envoyé à l'hôte lorsqu'une connexion est établie. Si ce champ reste vide, l'utilisateur doit fournir un mot de passe lors de la connexion.

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

Mot de passe de cet objet.

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

L'objet ne peut être lancé que si le mot de passe est utilisé (VRAI).

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

L'objet ne peut être affiché que si le mot de passe correct est fourni (VRAI).

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

L'objet ne peut être écrit que si le mot de passe correct est fourni (VRAI).

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

Contrôle si les sessions utilisant cet objet sont consignées.

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

Contrôle si les sessions utilisant cet objet sont enregistrées depuis le début.

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

Nom du fichier d'enregistrement pour les sessions utilisant cet objet.

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

Définit les types de connexion.

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

Les types de connexion suivants sont disponibles :

- COM1
- COM2
- COM3
- COM4
- RNIS via CAPI 2.0
- NetBIOS
- SPX
- DEFAULT TAPI
- TCP/IP

Le nom d'un périphérique TAPI peut également servir de type de connexion. DEFAULT TAPI utilise le premier périphérique TAPI trouvé dans le système. Pour utiliser un périphérique TAPI spécifique, utilisez FirstConnectionType() / NextConnectionType() pour rechercher les périphériques disponibles.

Méthodes de détail d'objet distant

Lorsqu'un type de connexion est affecté à un objet distant, les détails du périphérique sont définis sur les valeurs par défaut valides.

Les types de connexion suivants possèdent des options de configuration avancées qui peuvent être définies dans l'application :

- Périphériques COM
- Périphériques NetBIOS
- Périphériques RNIS via CAPI 2.0 (RNIS européen seulement)

Détails de périphérique COM

Définit la parité des communications.

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

Les valeurs de parité de communication sont les suivantes :

- Aucune
- Impaire
- Paire
- Marque
- Espace

Définit le niveau de contrôle de flux.

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

Les valeurs de contrôle de flux sont les suivantes :

- <Aucun>
- XONXOFF
- RTS/CTS
- LES DEUX

Définit le paramètre de début de connexion.

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

Les valeurs de début de connexion sont les suivantes :

- Toujours connecté
- Détection de porteuse (DCD)
- Prêt à émettre (CTS)
- Ensemble de données prêt (DSR)
- Signal de sonnerie (RI)
- Réception 2 <CR>
- Réponse du modem

Définit les valeurs de fin de connexion.

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

Les valeurs de fin de connexion sont les suivantes :

- Toujours connecté
- Détection de porteuse (DCD)

- Prêt à émettre (CTS)
- Ensemble de données prêt (DSR)
- Signal de sonnerie (RI)

Définit la vitesse de connexion.

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

Les valeurs de vitesse de connexion sont les suivantes :

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

Détails de périphérique NetBios

Définit le numéro de LANA (adaptateur réseau) à utiliser pour cette connexion.

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

Détails de périphérique RNIS via CAPI2.0

Active la liaison de canaux (utilise deux canaux RNIS pour une connexion) si VRAI.

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

Définit toute extension CAPI supplémentaire nécessaire pour les communications.

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

Méthodes d'objet distant

Les tableaux suivants décrivent les méthodes normales de l'objet distant (elles ne servent pas à obtenir et à définir les propriétés de l'objet).

Méthode	Paramètre ou valeur renvoyés	Description
short ConnectionTypes();	Valeur retournée = Court	Nombre de types de connexion disponibles sur cet ordinateur
BSTR FirstConnectionType(); et BSTR NextConnectionType();	Valeur renvoyée = BSTR	FirstConnectionType() et NextConnectionType() servent à itérer parmi les types de connexion disponibles. Ces fonctions renvoient un BSTR qui est le nom d'un type de connexion disponible. Les types de connexion renvoyés peuvent être utilisés avec la fonction SetConnectionType(). La valeur renvoyée est le nom d'un type de périphérique de connexion pris en charge.
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Paramètre = LPCTSTR lpszConnectionType Valeur renvoyée = booléen	Nom d'un type de périphérique de connexion. La valeur renvoyée est VRAI si le type de connexion transmis existe sur l'ordinateur.
short CountryCodes();	Valeur retournée = Court	Renvoie le nombre d'indicatifs internationaux disponibles.
BSTR FirstCountryCode(); et BSTR NextCountryCode();	Valeur renvoyée = BSTR	FirstCountryCode() et NextCountryCode() servent à circuler parmi les indicatifs internationaux disponibles. Ces fonctions renvoient un BSTR qui est le nom d'un indicatif international disponible. Les valeurs renvoyées peuvent être utilisées avec la fonction SetCountryCode(). La valeur renvoyée est la première chaîne d'indicatif international ou chaîne suivante.

Méthode	Paramètre ou valeur renvoyés	Description
BOOL ReadObject(LPCTSTR lpszPassword);	Paramètre = LPCTSTR lpszPassword Valeur renvoyée = booléen	Lit les données d'objet du fichier d'objet distant. Le paramètre est le mot de passe objet La valeur renvoyée est VRAI si l'objet est lancé avec succès.
BOOL WriteObject(LPCTSTR lpszPassword);	Paramètre = LPCTSTR lpszPassword Valeur renvoyée = booléen	Ecrit les données d'objet dans le fichier d'objet distant. Le paramètre est le mot de passe objet La valeur renvoyée est VRAI si l'objet est lancé avec succès.

objet CRemoteDataEx

L'objet CRemoteDataEx contient la même fonction que l'objet CRemoteData, avec les méthodes supplémentaires Get et Set (obtenir et définir) suivantes :

```
BSTR GetPrivateKey(); //Renvoie les informations de PrivateKey
void SetPrivateKey(LPCTSTR lpszNewValue);
BSTR GetCertificationName(); //Renvoie le nom de certificat
void SetCertificationName(LPCTSTR lpszNewValue);
short GetEncryptionLevel(); //Renvoie la valeur du niveau de chiffrement
void SetEncryptionLevel(short nNewValue);
BOOL GetDenyLowerEncrypt(); //Renvoie la valeur DenyLowerEncrypt
void SetDenyLowerEncrypt(BOOL bNewValue);
BSTR GetAutoDomain(); //Renvoie la valeur AutoDomain
void SetAutoDomain(LPCTSTR lpszNewValue);
```

Exemple de code Visual C++ pour la fonctionnalité du distant

L'exemple suivant de fonction C++ crée un objet distant, définit son type de connexion sur TCP/IP, définit le nom de l'ordinateur sur l'adresse TCP/IP transmise dans la fonction, puis lance l'objet distant.

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

```
CRemoteDataManager remoteDM;

CRemoteData remoteData;

// Créer d'abord le CRemoteDataManager
remoteDM.CreateDispatch( _T( "WINAWSVR.RemoteDataManager" ) );
// Créer ensuite CRemoteData et le rattacher
remoteData.AttachDispatch( remoteDM.CreateObject("Test", 0) );

// Définir à présent les propriétés requises
remoteData.SetConnectionType("TCP/IP");

remoteData.SetComputerName(lpszAddress);

// Enregistrer les données de l'objet
if (remoteData.WriteObject(0))
{
    // Et le lancer
    if (remoteData.Launch())
        bReturn = TRUE;
}
// Libérer l'objet distant.
remoteData.ReleaseDispatch();

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

return bReturn;
}
```

Méthodes CHostDataManager

Le table suivant décrit les méthodes CHostDataManager, y compris les paramètres et les valeurs renvoyées pour accéder aux objets CHostData et les contrôler :

Méthode	Paramètre ou valeur renvoyés	Description
BSTR CurrentDirectory();	Valeur renvoyée = BSTR	Renvoie le chemin d'accès complet du répertoire en cours dans lequel les objets hôte de pcAnywhere sont stockés.
BOOL ChangeDirectory(LPCTSTR lpszNewDirectory);	Paramètre = LPCTSTR lpszNewDirectory Valeur renvoyée = booléen	Change le dossier en cours dans lequel les objets hôte de pcAnywhere sont stockés. Le paramètre est le nom d'un dossier existant. La valeur renvoyée est VRAI en cas de succès.
BOOL FindFirst(LPCTSTR lpszPattern);	Paramètre = LPCTSTR lpszPattern Valeur renvoyée = booléen	Trouve le premier fichier d'objet hôte pcAnywhere (*.bhf) dans le dossier en cours, sur la base du schéma du nom de fichier spécifié. Le paramètre est le schéma de nom de fichier pour filtrer les fichiers d'objet (un astérisque [*] trouve tous les fichiers hôte dans le répertoire en cours). La valeur renvoyée est VRAI si un fichier d'objet hôte correspondant au schéma spécifié est trouvé. Le chemin d'accès complet du fichier est stocké dans pbstrFullQualName.
BOOL FindFirst(BSTR FAR* pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName	Tampon de retour pour le chemin d'accès complet du fichier d'objet distant correspondant au schéma spécifié. La valeur renvoyée est VRAI si un fichier d'objet hôte correspondant au schéma spécifié est trouvé. Le chemin d'accès complet du fichier est stocké dans pbstrFullQualName.

Méthode	Paramètre ou valeur renvoyés	Description
BOOL FindNext(BSTR FAR* pbstrFullQualName);	Parameter = BSTR FAR * pbstrFullQualName Valeur renvoyée = booléen	<p>Après l'appel réussi de FindFirst() pour obtenir le nom d'un fichier d'objet hôte dans le dossier en cours, FindNext() peut être appelé pour trouver le fichier suivant correspondant au schéma.</p> <p>Le paramètre est le tampon de retour pour le chemin d'accès complet du fichier d'objet distant correspondant au schéma spécifié dans l'appel initial de FindFirst().</p> <p>Renvoie la valeur VRAI si un autre fichier d'objet distant correspondant au schéma spécifié dans l'appel de FindFirst() est trouvé. Le chemin d'accès complet du fichier est stocké dans pbstrFullQualName.</p>
LPDISPATCH RetrieveObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	<p>Récupère un objet CHostData par nom de fichier.</p> <p>Le paramètre est le nom intégralement qualifié du fichier d'objet hôte à charger.</p>
LPDISPATCH RetrieveObject(short wAccessMode);	Parameter = short wAccessMode	<p>Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes :</p> <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
LPDISPATCH RetrieveObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	<p>Mot de passe de l'objet. Peut être NULL.</p>
LPDISPATCH RetrieveObject	Valeur renvoyée = LPDISPATCH	<p>Pointeur sur un objet envoi OLE. L'objet est un objet CHostData.</p> <p>Voir Exemple de code Visual C++ pour la fonctionnalité Hôte.</p>

Méthode	Paramètre ou valeur renvoyés	Description
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Récupère un objet CHostDataEx par nom de fichier. Le paramètre est le nom intégralement qualifié du fichier d'objet hôte à charger.
LPDISPATCH RetrieveObjectEx(short wAccessMode);	Parameter = short wAccessMode	Spécifie comment cet objet doit être utilisé. Ce paramètre est lié à la protection par mot de passe. Les options sont les suivantes : <ul style="list-style-type: none"> ■ 0 = Non spécifié ■ 1 = Afficher uniquement ■ 2 = Afficher et modifier ■ 3 = Exécuter
LPDISPATCH RetrieveObjectEx(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Mot de passe de l'objet. Peut être NULL.
LPDISPATCH RetrieveObjectEx	Valeur renvoyée = LPDISPATCH	Pointeur sur un objet envoi OLE. L'objet est un objet CHostDataEx. Voir Exemple de code Visual C++ pour la fonctionnalité Hôte .
LPDISPATCH CreateObject(LPCTSTR lpszName);	Parameter = LPCTSTR lpszFQName Valeur renvoyée = LPDISPATCH	Crée un objet CHostData et renvoie un pointeur LPDISPATCH qui le désigne. Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet. La valeur renvoyée est un pointeur vers un objet envoi OLE. L'objet est un objet CHostData. Voir Exemple de code Visual C++ pour la fonctionnalité Hôte .

Méthode	Paramètre ou valeur renvoyés	Description
LPDISPATCH CreateObjectEx(LPCTSTR lpszName);	Parameter = LPCTSTR lpszFQName Valeur renvoyée = LPDISPATCH	Crée un objet CHostDataEx et renvoie un pointeur LPDISPATCH qui le désigne. Le paramètre est le nom de fichier entièrement qualifié d'objet distant pour le nouvel objet. La valeur renvoyée est un pointeur vers un objet envoi OLE. L'objet est un objet CHostDataEx. Voir Exemple de code Visual C++ pour la fonctionnalité Hôte .
BOOL DeleteObject(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName	Supprime un fichier d'objet hôte. Le paramètre est le fichier objet hôte entièrement qualifié de l'objet à supprimer.
BOOL DeleteObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword	Mot de passe de l'objet. Peut être NULL.
BOOL DeleteObject	Valeur renvoyée = booléen	VRAI si l'objet est supprimé.
BOOL Launch(LPCTSTR lpszFQName);	Parameter = LPCTSTR lpszFQName Valeur renvoyée = booléen	Lance un fichier d'objet hôte, ce qui ouvre la fenêtre de terminal de l'hôte pcAnywhere. Le paramètre est le fichier objet hôte entièrement qualifié de l'objet à lancer. La valeur renvoyée est VRAI si l'objet est lancé avec succès.

objet CHostData

Utilisez cet objet pour modifier les données de l'objet hôte.

Méthodes Get et Set (obtenir et définir)

Les méthodes suivantes servent à obtenir (Get) et définir (Set) les propriétés de l'objet CHostData.

Le numéro de téléphone est le numéro à composer pour établir une connexion par modem avec un ordinateur distant `pcAnywhere`.

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

Indique si des propriétés de numérotation TAPI doivent être utilisées (informations d'emplacement) (VRAI) ou si la chaîne du numéro de téléphone doit être utilisée exactement telle qu'elle apparaît (FAUX).

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

Si les propriétés de numérotation sont utilisées, il s'agit du code de zone du numéro à appeler.

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

Si les propriétés de numérotation sont utilisées, il s'agit de l'indicatif international du numéro à appeler.

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

Nombre de tentatives d'appel de ce numéro si l'appel échoue.

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

Délai en secondes entre les tentatives d'appel.

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

Contrôle si les sessions utilisant cet objet sont consignées.

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

Contrôle si les sessions utilisant cet objet sont enregistrées depuis le début.

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

Nom du fichier d'enregistrement pour les sessions utilisant cet objet.

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

Méthodes de détail d'objet hôte

Lorsqu'un type de connexion est affecté à un objet hôte, les détails du périphérique sont définis sur des valeurs par défaut valides.

Les types de connexion suivants possèdent des options de configuration avancées qui peuvent être définies dans l'application :

- Périphériques COM
- Périphériques NetBIOS
- Périphériques NASI/NCSI
- Périphériques RNIS via CAPI 2.0 (RNIS européen seulement)

Détails de périphérique COM

Le code suivant place le type de connexion requis dans la liste des types de connexion affectés à l'objet hôte et en fait le type de connexion en cours lors du traitement ultérieur des appels de méthodes propres au périphérique.

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

Si le type de connexion requis se trouve déjà dans la liste des connexions affectées, celle-ci ne change pas. Seul le type de connexion actuel est remplacé par le type demandé. Il est normal d'appeler la méthode `AssignConnection` sur le même objet plusieurs fois lors de l'obtention et de la définition de valeurs propres à la connexion.

`AssignConnection` renvoie VRAI si le type de connexion transmis existe sur l'ordinateur et est affecté avec succès ou déjà affecté. Il renvoie FAUX si le type de connexion requis n'existe pas sur l'ordinateur ou si le nombre de connexions affectées a déjà atteint le maximum.

Un objet hôte `pcAnywhere` peut prendre en charge jusqu'à deux types de connexion affectés. La méthode `AssignConnection` renvoie FAUX si elle détecte une tentative de dépassement de cette limite.

Les types de connexion suivants sont disponibles :

- COM1
- COM2
- COM3
- COM4
- SPX
- NetBIOS

- TCP/IP
- RNIS via CAPI 2.0
- DEFAULT TAPI

Le nom d'un périphérique TAPI peut également servir de type de connexion. DEFAULT TAPI utilise le premier périphérique TAPI trouvé dans le système. Pour utiliser un périphérique TAPI spécifique, utilisez `FirstConnectionType()` et `NextConnectionType()` pour rechercher les périphériques disponibles.

Le code suivant désaffecte un type de connexion. Après avoir désaffecté un type de connexion, la connexion affectée restante, s'il y en a une, devient le type de connexion actuel pour les appels ultérieurs de méthode propres au périphérique.

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

Le code suivant définit la parité des communications :

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

Les valeurs de parité de communication sont les suivantes :

- Aucune
- Impaire
- Paire
- Marque
- Espace

Le code suivant définit les niveaux de contrôle de flux :

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

Les valeurs de contrôle de flux sont les suivantes :

- <Aucun>
- XONXOFF
- RTS/CTS
- LES DEUX

Le code suivant définit les valeurs de début de connexion :

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

Les valeurs de début de connexion sont les suivantes :

- Toujours connecté
- Détection de porteuse (DCD)
- Prêt à émettre (CTS)
- Ensemble de données prêt (DSR)
- Signal de sonnerie (RI)
- Réception 2 <CR>
- Réponse du modem

Le code suivant définit les valeurs de fin de connexion :

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

Les valeurs de fin de connexion sont les suivantes :

- Toujours connecté
- Détection de porteuse (DCD)
- Prêt à émettre (CTS)
- Ensemble de données prêt (DSR)
- Signal de sonnerie (RI)

Le code suivant définit la vitesse de connexion :

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

Les valeurs de vitesse de connexion sont les suivantes :

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

- 57600
- 115200

Détails de périphérique NetBios

Définit le numéro de LANA (adaptateur réseau) à utiliser pour cette connexion.

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

Détails de périphérique NASI/NCSI

Définit le nom de l'utilisateur pour le serveur NASI.

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

Définit le mot de passe de l'utilisateur pour le serveur NASI.

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

Définit le nom de session NASI.

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

Spécifie le serveur NASI à utiliser.

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

Spécifie le service NASI à utiliser.

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

Spécifie le port NASI à utiliser.

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

Détails de périphérique RNIS via CAPI2.0

Active la liaison de canaux (utilise deux canaux RNIS pour une connexion) si VRAI.

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

Définit toute extension CAPI supplémentaire nécessaire pour les communications.

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

Méthodes d'objet hôte

Les tableaux suivants décrivent les méthodes normales de l'objet (elles ne servent pas à obtenir et à définir les propriétés de l'objet).

Méthode	Paramètre ou valeur renvoyés	Description
short ConnectionTypes();	Valeur retournée = Court	Renvoie le nombre de types de connexion disponibles sur cet ordinateur
BSTR FirstConnectionType(); et BSTR NextConnectionType();	Valeur renvoyée = BSTR	FirstConnectionType() et NextConnectionType() servent à itérer parmi les types de connexion disponibles. Ces fonctions renvoient un BSTR qui est le nom d'un type de connexion disponible. Les types de connexion renvoyés peuvent être utilisés avec la fonction SetConnectionType(). La valeur renvoyée est le nom d'un type de périphérique de connexion pris en charge.
BOOL FindConnectionType(LPCTSTR lpszConnectionType);	Paramètre = LPCTSTR lpszConnectionTyp Valeur renvoyée = booléen	Le paramètre est le nom d'un type de périphérique de connexion La valeur renvoyée est VRAIE si ce type de périphérique est disponible
short MaxAssignedConnections()	Valeur retournée = Court	Renvoie le nombre maximum de types de connexion qui peuvent être affectés simultanément (actuellement deux).
short AssignedConnections()	Valeur retournée = Court	Renvoie le nombre de types de connexion affectés.

Méthode	Paramètre ou valeur renvoyés	Description
BSTR FirstAssignedConnection(); et BSTR NextAssignedConnection();	Valeur renvoyée = BSTR	FirstAssignedConnection() et NextAssignedConnection() servent à itérer parmi les connexions affectées. Ces fonctions renvoient un BSTR, qui est le nom d'un type de connexion affecté. Les types de connexion renvoyés peuvent être utilisés avec la fonction AssignConnection(). La valeur renvoyée est le nom d'un type de périphérique de connexion pris en charge.
BOOL FindAssignedConnection (LPCTSTR lpszConnectionType);	Paramètre = LPCTSTR lpszConnectionTyp Valeur renvoyée = booléen	Le paramètre est le nom d'un type de périphérique de connexion La valeur renvoyée est VRAI si ce type de périphérique est actuellement affecté
short CountryCodes();	Valeur retournée = Court	Renvoie le nombre d'indicatifs internationaux disponibles.
BSTR FirstCountryCode(); et BSTR NextCountryCode();	Valeur renvoyée = BSTR	FirstCountryCode() et NextCountryCode() servent à itérer parmi la liste des indicatifs internationaux disponibles. Les fonctions renvoient un BSTR qui est le nom d'un indicatif international disponible. Les valeurs renvoyées peuvent être utilisées avec la fonction SetCountryCode(). La valeur renvoyée est la première chaîne d'indicatif international ou chaîne suivante.
BOOL ReadObject(LPCTSTR lpszPassword);	Parameter = LPCTSTR lpszPassword Valeur renvoyée = booléen	Lit les données d'objet à partir du fichier d'objet hôte. Le paramètre est le mot de passe objet La valeur renvoyée est VRAI si l'objet est lancé avec succès.

Méthode	Paramètre ou valeur renvoyés	Description
BOOL WriteObject(LPCTSTR lpszPassword);	Paramètre = LPCTSTR lpszPassword Valeur renvoyée = booléen	Ecrit les données d'objet dans le fichier d'objet hôte. Le paramètre est le mot de passe objet La valeur renvoyée est VRAI si l'objet est lancé avec succès.

objet CHostDataEx

L'objet CHostDataEx contient la même fonction que l'objet CHostData, avec les méthodes supplémentaires Get et Set (obtenir et définir) suivantes :

```

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

Exemple de code Visual C++ pour la fonctionnalité Hôte

L'exemple suivant de fonction Visual C++ crée un objet hôte, définit son type de connexion sur TCP/IP, définit le nom de l'ordinateur sur l'adresse TCP/IP transmise à la fonction, puis lance l'objet hôte.

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

    CHostDataManager hostDM;
    CHostData hostData;

    // Créer d'abord le CHostDataManager
    hostDM.CreateDispatch( _T( "WINAWSVR.BeHostDataManager" ) );

    // Créer ensuite CRemoteData et le rattacher
    hostData.AttachDispatch(hostDM.CreateObject("Test", 0) );

    // Définir à présent les propriétés requises
    hostData.SetConnectionType("TCP/IP");

    // Enregistrer les données de l'objet
    if (hostData.WriteObject(0))
    {
        // Et le lancer
        if (hostData.Launch())
            bReturn = TRUE;
    }

    // Libérer l'objet hôte.
    hostData.ReleaseDispatch();

    return (bReturn);
}
```

fonctions awrem32

Le tableau suivant décrit les fonctions Awrem32, y compris les paramètres et les valeurs renvoyées pour la gestion des connexions entre un hôte et un ordinateur distant :

Méthode	Paramètre ou valeur renvoyés	Description
boolean awConnect(BSTR FileName);	Paramètre = Nom comme chaîne Valeur renvoyée = booléen	Crée la connexion à l'ordinateur hôte. Le paramètre est le nom intégralement qualifié du fichier .chf contenant les informations sur l'ordinateur hôte. La valeur renvoyée exécute la commande
boolean awDisconnect();	Valeur renvoyée = booléen	Déconnecte l'ordinateur hôte. Après avoir appelé cette fonction, le programme appelant doit supprimer l'objet (C++ - delete IAwrem32X*, VB - set ObjectName = Nothing;).
boolean FileXferFromHost(BSTR HostFile);	Paramètre = HostFile comme chaîne	Copie un fichier de l'ordinateur hôte vers l'ordinateur distant. Les paramètres peuvent contenir des caractères génériques. Le paramètre contient le chemin d'accès intégralement qualifié du fichier à copier à partir de l'ordinateur hôte. Les chaînes de fichier hôte et de fichier distant ne sont pas obligatoirement identiques.
boolean FileXferFromHost(BSTR RemoteFile);	Paramètre = RemoteFile comme chaîne	Copie un fichier de l'ordinateur hôte vers l'ordinateur distant. Les paramètres peuvent contenir des caractères génériques. Le paramètre contient le chemin d'accès de destination et le nom de fichier, entièrement qualifiés. Les chaînes de fichier hôte et de fichier distant ne sont pas obligatoirement identiques.
boolean FileXferFromHost	Valeur renvoyée = booléen	VRAI si la commande est exécutée.

Méthode	Paramètre ou valeur renvoyés	Description
boolean FileXferToHost(BSTR HostFile);	Paramètre = HostFile comme chaîne	Copie un fichier de l'ordinateur distant vers l'ordinateur hôte. Les paramètres peuvent contenir des caractères génériques. Le paramètre contient le chemin d'accès de destination et le nom de fichier, entièrement qualifiés. Les chaînes de fichier hôte et de fichier distant ne sont pas obligatoirement identiques.
boolean FileXferToHost(BSTR RemoteFile);	Paramètre = RemoteFile comme chaîne	Copie un fichier de l'ordinateur distant vers l'ordinateur hôte. Les paramètres peuvent contenir des caractères génériques. Le paramètre contient le chemin d'accès intégralement qualifié du fichier à copier à partir de l'ordinateur distant. Les chaînes de fichier hôte et de fichier distant ne sont pas obligatoirement identiques.
boolean FileXferToHost	Valeur renvoyée = booléen	VRAI si la commande est exécutée.
boolean CreateFolderOnHost(BSTR FolderName);	Paramètre = Nom de dossier comme chaîne Valeur renvoyée = booléen	Crée un nouveau dossier sur l'ordinateur hôte. Cette fonction crée un dossier temporaire sur l'ordinateur distant, puis copie ce dossier sur l'ordinateur hôte. Ce paramètre contient le lecteur et le chemin d'accès du dossier sur l'ordinateur hôte. La valeur renvoyée est VRAI si la commande est exécutée.

Méthode	Paramètre ou valeur renvoyés	Description
boolean ExecuteHostFile(BSTR FileName);	Paramètre = Nom de fichier comme chaîne Valeur renvoyée = booléen	Exécute un fichier existant sur l'ordinateur hôte. Cette fonction ne peut exécuter que les fichiers batch, de commandes et exécutables. Elle n'exécute pas les fichiers associés aux exécutables. Par exemple, elle n'ouvre pas Microsoft Word si vous exécutez un fichier .doc. Contient le chemin d'accès intégralement qualifié du fichier sur l'ordinateur hôte. La valeur renvoyée est VRAI si la commande est exécutée.
BSTR GetError();	Valeur renvoyée = Chaîne	Renvoie la dernière erreur générée dans Awrem32, comme chaîne.
short ConnectionStatus();	Valeur retournée = Court	Renvoie l'état courant de la connexion à l'ordinateur hôte. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> ■ -1 = Connexion perdue ■ 0 = Aucune connexion ■ 1 = Session connectée

Index

A

automatisation OLE. *Se reporter à* serveur d'automatisation pcAnywhere
Awrem32
fonctions 58, 88

B

bibliothèques API 14
bibliothèques de type 14

C

C++.. *Se reporter à* Visual C++
CHostData
utilisation de Visual Basic 37
utilisation de Visual C++ 77
CHostDataEx
utilisation de Visual Basic 48
utilisation de Visual C++ 85
CHostDataManager
utilisation de Visual Basic 34
utilisation de Visual C++ 73
connexions
arrêt 58, 88
démarrage 37, 58, 65, 88
erreurs de retour 90
retour de l'état 60, 90
connexions CAPI
propriétés de l'hôte 48, 83
propriétés des distants 30, 70
connexions NetBIOS
propriétés de l'hôte 47, 82
propriétés des distants 29, 70
connexions par modem
sur l'hôte
indicateurs internationaux 84
périphériques TAPI 80
propriétés COM 44, 79
propriétés de numérotation 43
sur l'ordinateur distant
indicateurs internationaux 71

périphériques TAPI 68
propriétés COM 26, 68
propriétés de numérotation 25
connexions réseau
sur l'hôte
propriétés NetBIOS 47, 82
sur l'ordinateur distant
propriétés NetBIOS 29, 70
contrôle de flux 27, 44, 68, 80
contrôleurs d'automatisation
à propos de Visual Basic 12
à propos de Visual C++ 14
CRemoteData
utilisation de Visual Basic 21
utilisation de Visual C++ 66
CRemoteDataEx
utilisation de Visual Basic 31
utilisation de Visual C++ 72
CRemoteDataManager
utilisation de Visual Basic 18
utilisation de Visual C++ 62

D

définition des classes
affichage 15
importing (importation) 14
dossiers
création sur l'hôte 59, 89

E

erreurs
renvoi 59, 90
état
renvoi 60, 90
exemples de code
Visual Basic 32, 56
Visual C++ 72, 87

- F**
- fichiers
 - exécution sur l'hôte 59, 90
- G**
- GUID 11
- I**
- identificateurs. *Se reporter à* GUID
 - indicatifs internationaux
 - sur l'hôte 84
 - sur l'ordinateur distant 71
- M**
- moteur distant
 - inscription automatique 11
 - inscription manuelle 12
 - mots de passe
 - sur les objets distants 72
 - sur les objets hôte 84
- O**
- objets distants
 - création 65
 - détails du périphérique 68
 - fichiers 19–20, 35, 63–64
 - méthodes 71
 - mots de passe 72
 - propriétés de numérotation 25
 - répertoires 18, 62
 - suppression 65
 - objets hôte
 - création 76
 - démarrage 37, 77
 - détails du périphérique 79
 - méthodes 83
 - mots de passe 84
 - propriétés de numérotation 43
 - recherche 35–36, 74, 76
 - répertoires 34, 74
 - suppression 36, 77
- P**
- paramètres de chiffrement
 - sur l'hôte 48, 85
 - sur l'ordinateur distant 31–32, 72
 - parité 27, 44, 68, 80
- Périphériques COM
 - détails de périphérique distant 26, 68
 - détails de périphérique hôte 44, 79
- périphériques NASI 82
- périphériques NCSI 82
- périphériques TAPI
 - sur l'hôte 80
 - sur l'ordinateur distant 68
- périphériques.. *Se reporter à* types de connexion
- R**
- RNIS
 - connexions CAPI
 - propriétés de l'hôte 48, 83
 - propriétés des distants 30, 70
- S**
- serveur d'automatisation pcAnywhere
 - à propos de 10
 - accès avec Visual Basic 12
 - accès avec Visual C++ 14
 - bibliothèques de type 14
 - exemples d'utilisation 10
 - inscription des GUID 11
 - serveur d'automatisation.. *Se reporter à* serveur d'automatisation pcAnywhere
- T**
- transfert de fichiers
 - à partir de l'hôte 58, 88
 - vers l'hôte 59, 89
 - types de connexion
 - attribution 42
 - désaffectation 42
 - propriétés de l'hôte 39, 79, 83
 - propriétés des distants 23, 68, 71
- V**
- Visual Basic
 - accès au serveur d'automatisation pcAnywhere 12
 - exemples de code 32, 56
 - fonctions awrem32 58
 - objet CHostData 37
 - objet CHostDataEx 48
 - objet CHostDataManager 34
 - objet CRemoteData 21
 - objet CRemoteDataEx 31

objet CRemoteDataManager 18

Visual C++

accès au serveur d'automatisation

pcAnywhere 14

ajout de Winawsvr.h 16

exemples de code 72, 87

fonctions awrem32 88

importation de classes 14

objet CHostData 77

objet CHostDataEx 85

objet CHostDataManager 73

objet CRemoteData 66

objet CRemoteDataEx 72

objet CRemoteDataManager 62

W

Winawsvr

exécutable 12

fichier d'en-tête 16

objets

CHostData 37, 77

CHostDataEx 48, 85

CHostDataManager 34, 73

CRemoteData 21, 66

CRemoteDataEx 31, 72

CRemoteDataManager 18, 62